



Руководство пользователя  
Центральный процессор (ЦП) серии Nexto  
NX3004

MU214616 Ред. В

27 мая 2022 года

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

В соответствии с действующим в Бразилии законодательством, Кодексом защиты прав потребителей, мы предоставляем клиентам, использующим нашу продукцию, следующую информацию, касающуюся личной безопасности и помещений.

Оборудование для промышленной автоматизации, производимое компанией Altus, является прочным и надежным благодаря строгому контролю качества, которому оно подвергается. Тем не менее, любое электронное оборудование промышленного управления (программируемые контроллеры, числовые команды и т.д.) может повредить машины или управляемые ими процессы при наличии неисправных элементов и/или при ошибке программирования или установки. Это может даже поставить под угрозу человеческие жизни. Пользователь должен учитывать возможные последствия дефектов и в целях безопасности предусмотреть дополнительные внешние установки. Данная проблема является более серьезной при первоначальном вводе в эксплуатацию и проведении испытаний.

Оборудование, производимое компанией Altus, не подвергает окружающую среду непосредственной опасности, поскольку в процессе использования не выделяет загрязняющих веществ. Тем не менее, что касается утилизации оборудования, важно отметить, что встроенная электроника может содержать материалы, которые при неправильной утилизации могут нанести вред природе. По этой причине рекомендуется при утилизации такого типа продукции направлять ее на перерабатывающие предприятия, которые гарантируют правильную утилизацию отходов.

Перед установкой или использованием изделия необходимо прочитать и вникнуть в суть документации, прилагаемой к нему, например, руководства и технические характеристики. Примеры и рисунки, представленные в настоящем документе, приведены исключительно в иллюстративных целях. В связи с возможными обновлениями и улучшениями, которые могут быть внесены в изделия, компания Altus не несет ответственности за использование данных примеров и рисунков в реальных приложениях. Они должны использоваться только для обучения пользователей и улучшения опыта работы с изделиями и их функциями.

Компания Altus обеспечивает гарантией свое оборудование, как описано в Общих условиях поставки, прилагаемых к коммерческим предложениям.

Компания Altus гарантирует, что ее оборудование работает в соответствии с четкими инструкциями, содержащимися в руководствах и/или технических характеристиках, не гарантируя успех какого-либо конкретного типа применения оборудования.

Компания Altus не признает никаких других гарантий, прямых или подразумеваемых, преимущественно, когда конечные клиенты имеют дело со сторонними поставщиками. Запросы на получение дополнительной информации о поставке, характеристиках оборудования и/или любых других услугах компании Altus должны быть сделаны в письменной форме. Компания Altus не несет ответственности за предоставление информации о своем оборудовании без официального запроса. В данных изделиях может использоваться технология EtherCAT® ([www.ethercat.org](http://www.ethercat.org)).

#### АВТОРСКИЕ ПРАВА

Nexto, MasterTool, Grano и WebPLC являются зарегистрированными торговыми марками компании Altus Sistemas de Automação S.A.

Windows, Windows NT и Windows Vista являются зарегистрированными торговыми марками корпорации Microsoft.

#### УВЕДОМЛЕНИЕ ОБ ИСПОЛЬЗОВАНИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

Для получения исходного кода в рамках лицензий GPL (генеральная общедоступная лицензия), LGPL (общедоступная лицензия ограниченного применения), MPL (открытая лицензия Mozilla) и других лицензий на программное обеспечение с открытым исходным кодом, содержащихся в данном продукте, обращайтесь по адресу [opensource@altus.com.br](mailto:opensource@altus.com.br). Помимо исходного кода, по запросу могут быть предоставлены все упомянутые лицензионные условия, отказы от гарантий и уведомления об авторских правах.

# Содержание

1. Введение.....	1
1.1. Серия Nexto .....	1
1.2. Инновационные функции .....	2
1.3. Документы, связанные с данным руководством .....	3
1.4. Визуальный контроль.....	4
1.5. Техническая поддержка.....	5
1.6. Предупреждающие сообщения, используемые в данном руководстве .....	5
2. Техническое описание .....	6
2.1. Панели и соединения.....	6
2.2. Общие характеристики .....	7
2.2.1. Стандартные общие характеристики.....	7
2.2.2. Память.....	9
2.2.3. Протоколы .....	11
2.2.4. Последовательные интерфейсы .....	12
2.2.4.1. COM 1.....	12
2.2.5. Интерфейсы локальной сети Ethernet .....	13
2.2.5.1. NET 1.....	13
2.2.6. Источник питания .....	14
2.2.7. Характеристики окружающей среды.....	14
2.3. Совместимость с другими изделиями.....	15
2.4. Эксплуатационные характеристики .....	15
2.4.1. Время применения .....	15
2.4.2. Время, необходимое для выполнения команд.....	15
2.4.3. Время инициализации .....	16
2.4.4. Временной интервал.....	16
2.5. Фактические размеры .....	16
2.6. Сведения о покупке .....	17
2.6.1. Элементы, входящие в комплект .....	17
2.6.2. Код изделия .....	17
2.7. Родственные изделия.....	18
3. Монтаж .....	19
3.1. Механический монтаж.....	19
3.2. Электрический монтаж .....	19

---

## СОДЕРЖАНИЕ

3.3. Подключение к локальной сети Ethernet.....	20
3.3.1. IP-адрес.....	20
3.3.2. Добровольный запрос ARP .....	20
3.3.3. Монтаж сетевого кабеля.....	20
3.4. Последовательное сетевое подключение RS-485/422.....	21
3.4.1. Связь по RS-485 без терминирования.....	22
3.4.2. Связь по RS-485 с внутренним терминированием .....	23
3.4.3. Связь по RS-485 с внешним терминированием .....	24
3.4.4. Связь по RS-422 без терминирования.....	25
3.4.5. Связь по RS-422 с внутренним терминированием .....	25
3.4.6. Связь по RS-422 с внешним терминированием .....	26
3.4.7. Пример сети RS-422 .....	27
3.5. Установка архитектуры.....	28
3.5.1. Установка модулей в главную стойку объединительной платы .....	28
3.6. Установка устройства программирования (программатора) .....	28
4. Начальное программирование .....	29
4.1. Организация памяти и доступ к ней .....	29
4.2. Профили проектов.....	32
4.2.1. Одиночный (Single).....	32
4.2.2. Базовый (Basic).....	33
4.2.3. Обычный (Normal) .....	33
4.2.4. Экспертный (Expert).....	34
4.2.5. Настраиваемый («Custom»).....	34
4.2.6. Машинный профиль (Machine Profile).....	35
4.2.7. Общая таблица.....	35
4.2.8. Максимальное число задач .....	36
4.3. Конфигурация центрального процессора (ЦП).....	37
4.4. Библиотеки.....	39
4.5. Вставка экземпляра протокола .....	39
4.5.1. Протокол MODBUS RTU .....	39
4.5.2. Протокол MODBUS Ethernet.....	40
4.6. Поиск устройства .....	42
4.7. Вход в систему .....	44
4.8. Режим выполнения (Run).....	46
4.9. Режим останова (Stop) .....	47
4.10. Запись и форсирование переменных .....	48
4.11. Выход из системы .....	49

---

## СОДЕРЖАНИЕ

4.12. Загрузка проекта .....	49
4.13. Рабочие состояния центрального процессора (ЦП) .....	51
4.13.1. Выполнение (Run) .....	51
4.13.2. Останов (Stop) .....	51
4.13.3. Точка контрольного прерывания (Breakpoint) .....	51
4.13.4. Исключение (Exception) .....	51
4.13.5. Теплый сброс (Reset Warm) .....	51
4.13.6. Холодный сброс (Reset Cold) .....	51
4.13.7. Сброс до исходного состояния (Reset Origin) .....	51
4.14. Программы (Структурные единицы программы (POU)) и списки глобальных переменных (GVL) .....	52
4.14.1. Программа MainPrg .....	52
4.14.2. Программа StartPrg .....	52
4.14.3. Программа UserPrg .....	53
4.14.4. Список глобальных переменных (GVL) System_Diagnostics .....	53
4.14.5. Список глобальных переменных (GVL) «Отключения» (Disables) .....	54
4.14.6. Список глобальных переменных (GVL) IOQualities .....	55
4.14.7. Список глобальных переменных (GVL) Module_Diagnostics .....	56
4.14.8. Список глобальных переменных (GVL) ReqDiagnostics .....	56
4.14.9. Функция Prepare_Start .....	58
4.14.10. Функция Prepare_Stop .....	58
4.14.11. Функция Start_Done .....	58
4.14.12. Функция Stop_Done .....	58
5. Конфигурация .....	59
5.1. Конфигурация центрального процессора (ЦП) .....	59
5.1.1. Общие параметры .....	59
5.1.1.1. «Горячая» замена .....	61
5.1.1.1.1. «Горячая» замена выключена, только для объявленных модулей .....	62
5.1.1.1.2. «Горячая» замена выключена .....	62
5.1.1.1.3. «Горячая» замена выключена, без согласованности при запуске .....	63
5.1.1.1.4. Горячая замена включена, с согласованностью при запуске только для объявленных модулей .....	63
5.1.1.1.5. «Горячая» замена включена с согласованностью при запуске .....	63
5.1.1.1.6. «Горячая» замена включена без согласованности при запуске .....	63
5.1.1.1.7. Как выполнить «горячую» замену .....	64
5.1.1.2. Области сохраняющей и постоянной памяти .....	66
5.1.1.3. Конфигурации TCP .....	67
5.1.1.4. Параметры проекта .....	69
5.1.2. Конфигурация внешних событий .....	69

---

## СОДЕРЖАНИЕ

5.1.3. Синхронизация времени.....	71
5.1.3.1. Простой протокол сетевого времени (SNTP).....	72
5.1.3.2. Скорректированное летнее время (DST) .....	73
5.2. Конфигурация последовательных интерфейсов .....	74
5.2.1. COM 1 .....	74
5.2.1.1. Расширенные конфигурации.....	75
5.3. Конфигурация интерфейсов локальной сети Ethernet.....	75
5.3.1. Локальные интерфейсы сети Ethernet.....	75
5.3.1.1. NET 1.....	75
5.3.2. Зарезервированные порты TCP.....	76
5.4. Конфигурация протоколов.....	76
5.4.1. Логика работы протокола x состояние центрального процессора (ЦП).....	79
5.4.2. Ведущее устройство MODBUS RTU (MODBUS RTU Master) .....	80
5.4.2.1. Конфигурирование протокола ведущего устройства MODBUS с помощью символьного преобразования .....	80
5.4.2.1.1. Общие параметры протокола ведущего устройства MODBUS - Конфигурация символьного преобразования.....	80
5.4.2.1.2. Конфигурация устройств - конфигурация символьного преобразования .....	84
5.4.2.1.3. Конфигурация преобразований – Настройки символьного преобразования.....	85
5.4.2.1.4. Конфигурация запросов – Настройки символьного преобразования .....	87
5.4.2.2. Конфигурация протокола ведущего устройства MODBUS (MODBUS Master) для прямого представления (%Q).....	92
5.4.2.2.1. Общие параметры протокола ведущего устройства MODBUS (MODBUS Master) - настройка посредством прямого представления (%Q) .....	92
5.4.2.2.2. Конфигурация устройств - Конфигурация для прямого представления (%Q).....	94
5.4.2.2.3. Конфигурация преобразований - Конфигурация для прямого представления (%Q).....	95
5.4.3. Ведомое устройство MODBUS RTU (MODBUS RTU Slave) .....	98
5.4.3.1. Конфигурирование протокола ведомого устройства MODBUS (MODBUS Slave) с помощью символьного преобразования.....	98
5.4.3.1.1. Общие параметры протокола ведомого устройства MODBUS (MODBUS Slave) – Конфигурирование посредством символьного преобразования .....	98
5.4.3.1.2. Конфигурация отношений - настройка символьного преобразования .....	103
5.4.3.2. Конфигурирование протокола ведомого устройства MODBUS (MODBUS Slave) через прямое представление (%Q) .....	105
5.4.3.2.1. Общие параметры протокола ведомого устройства (MODBUS) – Конфигурация через прямое представление (%Q).....	105
5.4.3.2.2. Конфигурация преобразований - Конфигурация через прямое представление (%Q) .....	106
5.4.4. Протокол MODBUS Ethernet.....	108
5.4.5. Клиент MODBUS Ethernet (MODBUS Ethernet Client) .....	110

---

## СОДЕРЖАНИЕ

5.4.5.1. Конфигурирование клиента MODBUS Ethernet через символьное преобразование .....	111
5.4.5.1.1. Общие параметры протокола клиента MODBUS - конфигурирование через символьное преобразование .....	111
5.4.5.1.2. Конфигурация устройств - Конфигурация через символьное преобразование .....	113
5.4.5.1.3. Конфигурация преобразований - Конфигурация через символьное преобразование .....	115
5.4.5.1.4. Конфигурация запросов - Конфигурация через символьное преобразование .....	116
5.4.5.2. Конфигурация клиента MODBUS Ethernet (MODBUS Ethernet Client) через прямое представление (%Q) .....	121
5.4.5.2.1. Общие параметры клиента протокола MODBUS - конфигурация для прямого представления (%Q) .....	121
5.4.5.2.2. Конфигурирование устройства - Конфигурация через прямое представление (%Q) .....	122
5.4.5.2.3. Конфигурация преобразований - Конфигурация через прямое представление (%Q) .....	124
5.4.5.3. Запуск отношения клиента MODBUS (MODBUS Client) в ациклической форме.....	126
5.4.6. Сервер MODBUS Ethernet (MODBUS Ethernet Server) .....	126
5.4.6.1. Конфигурация протокола Ethernet сервера MODBUS (MODBUS Server) для символьного преобразования .....	127
5.4.6.1.1. Общие параметры протокола сервера MODBUS (MODBUS Server) - Конфигурация через символьное преобразование.....	127
5.4.6.1.2. Диагностика сервера MODBUS (MODBUS Server) - Конфигурация через символьное преобразование .....	129
5.4.6.1.3. Конфигурация преобразований – Конфигурация через символьное преобразование.....	131
5.4.6.2. Конфигурирование протокола Ethernet сервера MODBUS (MODBUS Server) через прямое представление (%Q).....	132
5.4.6.2.1. Общие параметры протокола сервера MODBUS (MODBUS Server) - конфигурирование через прямое представление (%Q).....	132
5.4.6.2.2. Конфигурация преобразований - Конфигурация через прямое представление (%Q) .....	134
5.4.7. Сервер OPC DA (OPC DA Server) .....	137
5.4.7.1. Создание проекта для связи с OPC DA.....	139
5.4.7.2. Конфигурирование ПЛК на сервере OPC DA (OPC DA Server) .....	142
5.4.7.2.1. Импорт конфигурации проекта.....	144
5.4.7.3. Переменные качества и состояния связи OPC DA .....	145
5.4.7.4. Пределы связи с сервером OPC DA.....	146
5.4.7.5. Доступ к данным через клиент OPC DA (OPC DA Client) .....	147
5.4.8. Сервер OPC UA (OPC UA Server) .....	148
5.4.8.1. Создание проекта для связи с OPC UA.....	151
5.4.8.2. Типы поддерживаемых переменных .....	152
5.4.8.3. Ограничение количества подключенных клиентов на сервере OPC UA (OPC UA Server).....	152
5.4.8.4. Ограничение переменных связи на сервере OPC UA (OPC UA Server) .....	152
5.4.8.5. Настройки шифрования .....	152

---

## СОДЕРЖАНИЕ

5.4.8.6. Основные параметры связи, настраиваемые в клиенте OPC UA (OPC UA Client).....	153
5.4.8.6.1. URL-адрес конечной точки.....	154
5.4.8.6.2. Интервал публикации (мс) и Интервал выборки (мс) .....	154
5.4.8.6.3. Отсчет времени существования и Отсчет времени активного соединения .....	154
5.4.8.6.4. Размер очереди и Сброс самого старого значения .....	155
5.4.8.6.5. Тип фильтра и Тип зоны нечувствительности .....	155
5.4.8.6.6. Параметры «PublishingEnabled», «MaxNotificationsPerPublish» и «Priority» («Приоритет»).....	156
5.4.8.7. Доступ к данным через клиент OPC UA (OPC UA Client).....	156
5.4.9. Протокол EtherNet/IP .....	158
5.4.9.1. Протокол EtherNet/IP .....	158
5.4.9.2.1. Общее .....	160
5.4.9.2.2. Соединения.....	161
5.4.9.2.3. Узлы .....	162
5.4.9.2.4. Преобразование входов/выходов протокола EtherNet/IP .....	163
5.4.9.3. Конфигурация адаптера EtherNet/IP.....	163
5.4.9.3.1. Типы модулей.....	164
5.4.9.3.2. Преобразование входов/выходов модуля EtherNet/IP .....	164
5.5. Производительность связи .....	165
5.5.1. Сервер MODBUS.....	165
5.5.1.1. Локальные интерфейсы центрального процессора (ЦП) .....	165
5.5.2. Сервер OPC DA (OPC DA Server) .....	166
5.5.3. Сервер OPC UA .....	167
5.6. Эксплуатационные характеристики системы.....	167
5.6.1. Время сканирования входов/выходов .....	168
5.7. Часы реального времени (RTC).....	168
5.7.1. Функциональные блоки для считывания и записи часов реального времени (RTC) .....	169
5.7.1.1. Функциональные блоки для считывания часов реального времени (RTC).....	169
5.7.1.1.1. Функциональный блок <i>GetDateAndTime</i> .....	169
5.7.1.1.2. Функциональный блок <i>GetTimeZone</i> .....	170
5.7.1.1.3. Функция <i>GetDayOfWeek</i> .....	171
5.7.1.2. Функциональные блоки и функции записи и конфигурации часов реального времени (RTC).....	171
5.7.1.2.1. Функция <i>SetDateAndTime</i> .....	171
5.7.1.2.2. Функциональный блок <i>SetTimeZone</i> .....	173
5.7.2. Структуры данных часов реального времени (RTC) .....	174
5.7.2.1. Структура «EXTENDED_DATE_AND_TIME» .....	174
5.7.2.2. Структура «DAYS_OF_WEEK».....	174
5.7.2.3. Структура RTC_STATUS.....	174



---

## СОДЕРЖАНИЕ

5.7.2.4. Структура TIMEZONESETTINGS .....	175
5.8. Память пользовательских файлов.....	175
5.9. Информационное и конфигурационное меню центрального процессора (ЦП) .....	177
5.10. Функциональные блоки и функции .....	179
5.10.1. Специальные функциональные блоки для последовательных интерфейсов .....	179
5.10.1.1. Функциональный блок «SERIAL_CFG».....	184
5.10.1.2. Функциональный блок «SERIAL_GET_CFG» .....	186
5.10.1.3. Функциональный блок «SERIAL_GET_CTRL».....	187
5.10.1.4. Блок «SERIAL_GET_RX_QUEUE_STATUS» .....	189
5.10.1.5. Функциональный блок «SERIAL_PURGE_RX_QUEUE».....	190
5.10.1.6. Функциональный блок «SERIAL_RX».....	192
5.10.1.7. Функциональный блок «SERIAL_RX_EXTENDED» .....	194
5.10.1.8. Блок «SERIAL_SET_CTRL» .....	196
5.10.1.9. Функциональный блок «SERIAL_TX».....	198
5.10.2. Обновление входов и выходов .....	201
5.10.2.1. Функциональный блок «REFRESH_INPUT» .....	201
5.10.2.2. Функциональный блок «REFRESH_OUTPUT» .....	202
5.10.3. Функциональный блок PID (Пропорционально-Интегрально-Дифференциальный Регулятор) .....	203
5.10.4. Сохранение таймера .....	204
5.10.4.1. Функциональный блок «TOF_RET» .....	204
5.10.4.2. Функциональный блок «TON_RET».....	205
5.10.4.3. Функциональный блок «TP_RET».....	206
5.11. Простой протокол сетевого управления (SNMP) .....	208
5.11.1. Введение .....	208
5.11.2. Простой протокол сетевого управления (SNMP) на центральных процессорах (ЦП) Nexto .....	208
5.11.3. Частная информационная база управления (MIB) .....	209
5.11.4. Конфигурация простого протокола сетевого управления (SNMP).....	209
5.11.5. Пользователь и сообщества простого протокола сетевого управления (SNMP).....	211
5.12. Управление пользователями и права доступа .....	212
6. Техническое обслуживание .....	213
6.1. Диагностика модуля .....	213
6.1.1. Диагностика одним касанием (One Touch Diag).....	213
6.1.2. Диагностика с помощью светодиодного индикатора.....	216
6.1.2.1. DG (Диагностика) .....	216
6.1.2.2. WD (Сторожевой таймер) .....	217
6.1.2.3. Светодиодные индикаторы разъема RJ45 .....	217
6.1.3. Диагностика с помощью веб-страницы.....	217

---

**СОДЕРЖАНИЕ**

6.1.4. Диагностический проводник .....	220
6.1.5. Диагностика с помощью переменных .....	220
6.1.5.1. Сводная диагностика .....	221
6.1.5.2. Подробная диагностика .....	225
6.1.6. Диагностика с помощью функциональных блоков .....	234
6.1.6.1. Функция «GetTaskInfo» .....	234
6.2. Графический дисплей .....	235
6.3. Системный журнал .....	238
6.4. Отсутствие загрузки приложения при запуске .....	238
6.5. Распространенные проблемы .....	239
6.6. Поиск и устранение неисправностей .....	239
6.7. Планово-предупредительное техническое обслуживание .....	240

## 1. Введение

Программируемые (логические) контроллеры серии Nexto представляют собой оптимальное решение для промышленной автоматизации и управления системами. Благодаря высоким встроенным технологиям, изделия данного семейства способны управлять сложными промышленными системами, машинами, высокопроизводительными технологическими линиями и наиболее современными процессами Индустрии 4.0 распределенным и резервированным способом. Будучи современной и высокоскоростной, серия Nexto использует самые передовые технологии для обеспечения надежности и возможности подключения, помогая повысить производительность различных предприятий.

Компактные, устойчивые к сбоям и с высокой эксплуатационной готовностью, изделия данной серии обладают превосходной производительностью обработки данных и возможностями расширения стойки. Их архитектура обеспечивает простую интеграцию с сетями контроля, управления и промышленными сетями, в дополнение к резервированию ЦП (центрального процессора) и источника питания. Оборудование семейства также предлагает расширенную диагностику и «горячую» замену, сводя к минимуму либо исключая простои вследствие технического обслуживания и обеспечивая непрерывный производственный процесс.



Рисуно 1: NX3004

### 1.1. Серия Nexto

Серия Nexto представляет собой мощную и полнокомплектную серию программируемых (логических) контроллеров (ПЛК) с эксклюзивными и инновационными характеристиками. Благодаря своей гибкости, функциональному дизайну, расширенным диагностическим ресурсам и модульной архитектуре ПЛК Nexto могут использоваться для управления системами малого, среднего и крупного масштаба.

Архитектура серии Nexto отличается большим разнообразием модулей ввода и вывода. Данные модули в сочетании с мощным 32-разрядным процессором и высокоскоростной шиной на базе локальной сети Ethernet подходят для различных областей применения, таких как высокоскоростное управление малыми машинами, сложные распределенные процессы, резервированные приложения и системы с большим количеством входов/выходов, например, автоматизация зданий. Помимо этого, в серии Nexto присутствуют модули для управления движением, коммуникационные модули, охватывающие наиболее популярные полевые сети, а также другие возможности.

В шине серии Nexto используется передовая технология, основанная на высокоскоростном интерфейсе локальной сети Ethernet, позволяющая обмениваться входной и выходной информацией и данными между несколькими контроллерами внутри одной системы. Система может быть с легкостью разделена и распределена по всему полю, позволяя использовать расширение шины с той же производительностью, что и локальный модуль, что делает возможным использование любого модуля в локальном фрейме либо в фреймах расширения без каких-либо ограничений. Для обеспечения связи между фреймами расширения используется простой стандартный кабель локальной сети Ethernet.



Рисунок 2: Серия Nexto – Обзор

## 1.2. Инновационные функции

Серия Nexto предлагает пользователю множество инновационных решений, касающихся использования, контроля и технического обслуживания системы. Данные характеристики были разработаны с учетом новой концепции промышленной автоматизации.



**Работа без аккумуляторной батареи:** Для поддержания памяти и работы часов реального времени в серии Nexto не требуется использование батареи. Данная особенность чрезвычайно важна, поскольку она снижает потребность в техническом обслуживании системы и позволяет использовать ее в удаленных местах, где выполнение технического обслуживания может быть затруднительным. Помимо этого, данная функция является безвредной для окружающей среды.



**Система простого подключения (Easy Plug System):** Серия Nexto имеет эксклюзивный метод подключения и отключения клеммных колодок входа/выхода. Клеммные колодки легко снимаются одним движением и без применения специальных инструментов. Для того чтобы подключить клеммную колодку обратно к модулю, передняя крышка облегчает процедуру установки, устанавливая клеммную колодку на модуль.



**Многоблочное устройство хранения данных:** В ЦП (центральных процессорах) серии Nexto пользователю доступно несколько видов памяти, что позволяет подобрать оптимальный вариант для любых потребностей пользователя. Данные виды памяти подразделяются на энергозависимые и энергонезависимые. Для энергонезависимой памяти ЦП (центральные процессоры) серии Nexto предлагают адресуемый вход (%I), адресуемый выход (%Q), адресуемую память (%M), память данных и резервную память данных. Для приложений, требующих функциональности энергонезависимой памяти, центральные процессоры серии Nexto предлагают сохраняемую адресуемую память (%Q), сохраняемую память данных, постоянную адресуемую память (%Q), постоянную память данных, память программ, память исходного кода, файловую систему ЦП (центрального процессора) (doc, PDF, данные) и интерфейс карты памяти.



**One Touch Diag (диагностика одним касанием):** One Touch Diag (диагностика одним касанием) представляет собой эксклюзивную функцию, которую серия Nexto привносит в ПЛК. Благодаря этой новой концепции пользователь может проверить диагностическую информацию любого модуля, присутствующего в системе, непосредственно на графическом дисплее ЦП (центрального процессора) одним нажатием на диагностический переключатель соответствующего модуля. OTD представляет собой мощный диагностический инструмент, который можно использовать в автономном режиме (без супервизора или программиста), что сокращает время технического обслуживания и ввода в эксплуатацию.

**OFD – Полная документация, размещенная на плате:** ЦП (центральные процессоры) серии Nexto способны хранить всю проектную документацию в собственной памяти. Данная функция может быть очень удобна для целей

## 1. ВВЕДЕНИЕ

резервного копирования и технического обслуживания, поскольку вся информация хранится в одном надежном месте.

ETD – Электронный тег на дисплее: Еще одной эксклюзивной функцией, которую серия Nexto привносит в ПЛК, является электронный тег на дисплее. Данная новая функциональность позволяет проверить названия тегов любого входного/выходного контакта или модуля, используемого в системе, непосредственно на графическом дисплее ЦП (центрального процессора). Наряду с этой информацией пользователь также может проверить и описание. Эта функция чрезвычайно полезна во время выполнения технического обслуживания и устранения неисправностей.

DHW – Аппаратное обеспечение с удвоенной шириной: Модули серии Nexto были разработаны для экономии места в пользовательских шкафах или машинах. По этой причине модули серии Nexto поставляются в двух вариантах ширины: удвоенной ширины (требуется два слота для установки в стойку объединительной платы) и одинарной ширины (требуется только один слот для установки в стойку объединительной платы). Данная концепция позволяет использовать компактные модули входа/выхода с высокой плотностью точек входа/выхода наряду со сложными модулями, такими как ЦП (центральные процессоры), ведущие устройства (мастера) промышленной шины и модули источника питания.

Высокоскоростной центральный процессор: Все ЦП (центральные процессоры) серии Nexto были разработаны для обеспечения выдающейся производительности для пользователя, позволяя охватить большой диапазон требований к приложениям.



Премия iF Product Design Award 2012: Серия Nexto стала победителем iF Product Design Award 2012 в группе «Промышленность + профессии, требующие высокой квалификации». Данная награда признана во всем мире как знак качества и совершенства, и считается «Оскаром» в сфере проектирования в Европе.

### 1.3. Документы, связанные с данным руководством

Для получения дополнительной информации касательно серии Nexto, помимо данного документа, можно обратиться к другим документам (руководствам и техническим характеристикам). Эти документы доступны в последней версии на сайте [http://www.altus.com.br/site\\_en/](http://www.altus.com.br/site_en/).

Каждое изделие имеет документ, оформленный в форме Технических характеристик (CE), в котором описаны особенности изделия. Помимо этого, на изделие также могут иметься руководства по эксплуатации (коды руководств указаны в CE).

Например, для модуля NX2020 в CE указана информация об особенностях эксплуатации и приобретения. С другой стороны, модуль NX5001, помимо CE, имеет руководство пользователя (MU).

В качестве дополнительного источника информации рекомендуется использовать следующие документы:

Код	Описание	Язык
CE114000	Серия Nexto – Технические характеристики	Английский
CT114000	Série Nexto – Características Técnicas	Португальский
CS114000	Serie Nexto – Características Técnicas	Испанский
CE114103	Технические характеристики NX3004	Английский
CT114103	Características Técnicas NX3004	Португальский
CS114103	Especificaciones y Configuraciones NX3004	Испанский
CE114700	Технические характеристики стоек объединительных плат серии Nexto	Английский
CT114700	Características Técnicas dos Bastidores da Série Nexto	Португальский
CS114700	Características Técnicas de los Bastidores de la Serie Nexto	Испанский
CE114810	Технические характеристики принадлежностей серии Nexto для стоек объединительных плат	Английский
CT114810	Características Técnicas Acessórios para Bastidor Série Nexto	Португальский

## 1. ВВЕДЕНИЕ

CS114810	Características Técnicas del Cierres Laterales para el Bastidor	Испанский
CE114902	Технические характеристики интерфейса PROFIBUS-DP Master серии Nexto	Английский
Код	Описание	Язык
CT114902	Características Técnicas do Mestre PROFIBUS-DP da Série Nexto	Португальский
CS114902	Características Técnicas del Módulo Profibus-DP Maestro	Испанский
CE114908	Технические характеристики головок интерфейсов PROFIBUS-DP NX5110 и NX5210	Английский
CT114908	Características Técnicas Interfaces Cabeça PROFIBUSDP NX5110 e NX5210	Португальский
CS114908	Especificaciones y Configuraciones PROFIBUS-DP Interfaz Cabezas NX5110 y NX5210	Испанский
MU214600	Руководство пользователя серии Nexto	Английский
MU214000	Manual de Utilização Série Nexto	Португальский
MU214616	Руководство пользователя ЦП (центрального процессора) NX3004	Английский
MU214104	Manual de Utilização UCP NX3004	Португальский
MU299609	Руководство пользователя MasterTool IEC XE	Английский
MU299048	Manual de Utilização MasterTool IEC XE	Португальский
MP399609	Руководство по программированию MasterTool IEC XE	Английский
MP399048	Manual de Programação MasterTool IEC XE	Португальский
MU214601	Руководство пользователя ведущего устройства (мастера) PROFIBUS DP NX5001	Английский
MU214001	Manual de Utilização Mestre PROFIBUS-DP NX5001	Португальский
MU214608	Руководство по эксплуатации головок PROFIBUS-DP Nexto	Английский
MU214108	Manual de Utilização da Cabeça PROFIBUS-DP Nexto	Португальский
MU214603	Руководство по протоколу HART серии Nexto	Английский
MU214610	Руководство пользователя по расширенным функциям управления	Английский
NAP151	Utilização do Tunneller OPC	Португальский
NAP152	Extensão da potência de saída para até 20 W	Португальский
NAP165	Comunicação OPC UA com Controladores ALTUS	Португальский
NAP165_ing	Связь OPC UA с контроллерами компании ALTUS	Английский

Таблица 1: Сопутствующие документы

### 1.4. Визуальный контроль

Прежде чем приступить к производству монтажа, рекомендуется внимательно осмотреть оборудование, чтобы убедиться в отсутствии повреждений, полученных во время транспортировки. Убедитесь в том, что все требуемые детали находятся в идеальном состоянии. В случае обнаружения повреждений сообщите об этом транспортной компании или ближайшему дистрибьютору компании Altus.

## 1. ВВЕДЕНИЕ

### ОСТОРОЖНО:

Перед тем как извлечь модули из корпуса, необходимо разрядить потенциально возможную статическую энергию, накопившуюся в корпусе. Для этого перед выполнением погрузочно-разгрузочных операций с модулями необходимо прикоснуться (голыми руками) к любой металлической заземленной поверхности. Выполнение данной процедуры гарантирует, что предельные значения статической энергии модулей не будут превышены.

Важно зарегистрировать серийный номер каждого полученного оборудования, а также ревизии программного обеспечения, если таковые существуют. Данная информация необходима на случай обращения в службу технической поддержки компании Altus.

### 1.5. Техническая поддержка

Для получения информации о технической поддержке компании Altus в Сан-Леопольдо, Риу-Гранди-ду-Сул (RS), звоните по телефону +55 51 3589-9500. Для получения дополнительной информации о технической поддержке компании Altus, существующей в других местах, см. <https://www.altus.com.br/en/> или отправьте письмо по адресу [altus@altus.com.br](mailto:altus@altus.com.br).

Если оборудование уже установлено, то на момент обращения в службу технической поддержки необходимо располагать следующей информацией:

- Модель используемого оборудования и конфигурация установленной системы
- Серийный номер изделия
- Ревизия оборудования и версия исполнительного программного обеспечения (ПО), записанные на бирке, закрепленной на боковой стороне изделия
- Информация о режиме работы центрального процессора (ЦП), полученная с помощью MasterTool IEC XE
- Содержание прикладного программного обеспечения, полученное через MasterTool IEC XE
- Версия используемого устройства программирования (программатора)

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

В данном руководстве предупреждающие сообщения будут представлены в следующих форматах и со следующими значениями:

#### ОПАСНО:

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

#### ОСТОРОЖНО:

Сообщает о деталях конфигурации, применения или установки, которые должны быть приняты во внимание, чтобы избежать любых ситуаций, которые могут привести к сбою системы и последующему негативному воздействию.

#### ВНИМАНИЕ:

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

## 2. Техническое описание

В данной главе представлены все технические характеристики центральных процессоров (ЦП) NX3004 серии Nexto.

### 2.1. Панели и соединения

На следующем рисунке показана передняя панель центрального процессора (ЦП).



Рисунок 3: NX3004

Как видно из рисунка, в верхней части передней панели расположен графический дисплей, на котором отображается состояние и диагностика всей системы, включая специфическую диагностику каждого модуля. На графическом дисплее также присутствует удобное меню, которое предоставляет пользователю быстрый режим для считывания либо определения параметров, таких как: внутренняя температура (только считывание) и местное время (только считывание).

Сразу под графическим дисплеем расположены 2 светодиода, служащие для индикации диагностики аварийных сигналов и работы сторожевого таймера. Описание светодиодов представлено в таблице, приведенной ниже. Более подробная информация о состоянии и значении светодиодов приведена в разделе «[Диагностика с помощью светодиодов](#)».

Светодиод	Описание
DG	Диагностический светодиод
WD	Светодиод сторожевого таймера

Таблица 2: Описание светодиодов

В центральных процессорах (ЦП) серии Nexto пользователю доступны два переключателя. Описание данных переключателей приведено в таблице ниже. Более подробную информацию о переключателе диагностики см. в разделе «[Диагностика одним касанием \(One Touch Diag\)](#)».

Клавиши	Описание
Диагностический переключатель	Переключатель расположен на верхней части модуля. Используется для визуализации диагностики на графическом дисплее или для навигации по информационному меню и конфигурации центрального процессора (ЦП).

Таблица 3: Описание клавиш

На передней панели расположены интерфейсы подключения центральных процессоров (ЦП) серии Nexto. В таблице, приведенной ниже, представлено краткое описание данных интерфейсов.



Интерфейсы	Описание
NET 1	Коммуникационный разъем RJ45 стандарта 10/100Base-TX. Обеспечивает связь по типу «точка-точка» (коммуникация между двумя узлами) либо сетевую связь. Более подробная информация об использовании приведена в разделе « <a href="#">Конфигурация интерфейсов локальной сети Ethernet</a> ».
COM 1	Разъем типа мама (с внутренней резьбой) DB9 для стандарта связи RS-232. Обеспечивает связь по типу «точка-точка» (коммуникация между двумя узлами) либо сетевую связь. Более подробная информация об использовании приведена в разделе « <a href="#">Конфигурация последовательных интерфейсов</a> ».
Источник питания	Подключение с помощью клемм на выводе V1 к 24 В постоянного тока и на выводе N1 к 0 В постоянного тока. Он обеспечивает питание центрального процессора (ЦП), счетчиков, быстродействующих выходов и стойки, выдавая на них мощность до 10 Вт. Выводы V2 и N2, соответственно 24 В постоянного тока и 0 В постоянного тока, обеспечивают питание обычных выходов и входов.

Таблица 4: Интерфейсы подключения

## 2.2. Общие характеристики

### 2.2.1. Стандартные общие характеристики

	NX3004
Размещение в стойке объединительной платы	2 последовательных слота
Встроенный источник питания	Да
Локальный интерфейс Ethernet TCP/IP	1
Последовательный интерфейс	1
CAN-интерфейс	Нет
Хост-узел USB-порта	Нет
Интерфейс карты памяти	Нет
Часы реального времени (RTC)	Да Разрешение 1 мс, максимальное отклонение 2 секунды в день.
Сторожевой таймер (Устройство обеспечения безопасности сети)	Да
Индикация состояния и диагностики	Графический дисплей Светодиоды Веб-страницы Внутренняя память ЦП (центрального процессора)

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Языки программирования	Язык структурированного текста (ST) Язык лестничных диаграмм (LD) Язык последовательностных функциональных схем (SFC) Язык функциональных блоковых диаграмм (FBD) Язык непрерывных диаграмм функций (CFC)
Задачи	Циклические (периодические) Срабатывание по событию (программное прерывание) Срабатывание по внешнему событию (аппаратное прерывание) Непрерывные (свободный ход) Срабатывание по состоянию (программное прерывание)
Изменения в диалоговом режиме («онлайн»)	Да
Максимальное количество задач	16
Максимальное количество шин расширения	1
Поддержка избыточности расширения шины	Да
	NX3004
Максимальное количество модулей входа/выхода на шине	32
Максимальное количество дополнительных модулей интерфейса Ethernet TCP/IP	0
Поддержка резервирования интерфейсов Ethernet TCP/IP	Нет
Максимальное количество сетей PROFIBUS-DP (с использованием модулей ведущего устройства (мастера) PROFIBUS-DP)	1
Поддержка резервирования сети PROFIBUS-DP	Нет
Поддержка резервирования (полукластеры)	Нет
Поддержка «горячей» замены	Да
Событийно-ориентированная отчетность по данным (SOE)	Нет
Протокол	-
Максимальный размер очереди событий	-
Разработка веб-страниц (доступно через протокол HTTP)	Нет
Диагностика одним касанием (One Touch Diag) (OTD)	Да
Электронный тег на дисплее (ETD)	Да
Стандарты МЭК 61131-3	Да
Утверждение типа DNV-GL – DNVGL-CG-0339 (TAA000013D)	Да
МЭК 61131-2	Да

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ





 CE – 2014/35/EU (LVD (низковольтное оборудование)) и 2014/30/EU (EMC (электромагнитная совместимость))  RoHS Правила ограничения содержания вредных веществ (RoHS) – 2011/65/EU  LISTED Испытано и одобрено компанией Underwriters' Laboratories, Inc. – UL61010-1 (файл E473496)  EAC EAC – CU TR 004/2011 (LVD (низковольтное оборудование)) и CU TR 020/2011 (EMC (электромагнитная совместимость))	Да
	Да
	Да
	Да

Таблица 5: Стандартные характеристики

### Примечания:

Часы реального времени (RTC): Время хранения данных (время, в течение которого часы реального времени будут продолжать обновлять дату и время после выключения питания ЦП (центрального процессора)) составляет 15 дней для эксплуатации при температуре 25 °С. При максимальной температуре изделия, время хранения данных уменьшается до 10 дней.

Максимальное количество модулей входа/выхода на шине: Максимальное количество модулей входа/выхода относится к сумме всех модулей на локальной шине и расширений.

### 2.2.2. Память

	NX3004
Память адресуемых входных переменных (%I)	32 Кбайт
Память адресуемых выходных переменных (%Q)	32 Кбайт
Память переменных прямого представления (%M)	16 Кбайт
Память символических переменных	2 Мбайт
Максимальный объем памяти, конфигурируемой как долговременная (сохраняющая) или постоянная	7,5 Кбайт
Память данных с полным резервированием	-
Память входных переменных прямого представления (%I)	-
Память выходных переменных прямого представления (%Q)	-
Память переменных прямого представления (%M)	-
Память символических переменных	-
Память программ	3 Мбайт
Память исходного кода (резервное копирование)	32 Мбайт
Память пользовательских файлов	16 Мбайт

Таблица 6: Память

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Примечания:

Память адресуемых входных переменных (%I): Область, в которой хранятся адресуемые входные переменные. Адресуемые переменные означают, что к ним можно обращаться напрямую, используя нужный адрес. Например: %IB0, %IW100. Адресуемые входные переменные могут использоваться для преобразования точек цифровых либо аналоговых входов. В качестве примера можно привести 8 цифровых входов, представленных в одном байте, и одну точку аналогового входа, представленную в двух байтах.

Общая память адресуемых выходных переменных (%Q): Область, в которой хранятся адресуемые выходные переменные. Адресуемые переменные означают, что к ним можно обращаться напрямую, используя нужный адрес. Например: %QB0, %QW100. Адресуемые выходные переменные могут использоваться для преобразования точек цифрового или аналогового выхода. В качестве примера можно привести 8 цифровых выходов, представленных на один байт, и одну точку аналогового выхода, представленную на два байта. Адресуемые выходные переменные могут быть сконфигурированы как сохраняемые, постоянные или избыточные (резервированные), однако их общий размер не изменяется в зависимости от конфигурации.

Память адресуемых переменных (%M): Область, в которой хранятся адресуемые двузначные (маркерные) переменные. Адресуемые переменные означают, что к ним можно обращаться напрямую, используя нужный адрес. Например: %MB0, %MW100.

Память символических переменных: Область, в которой выделяются символические переменные. Символические переменные представляют собой переменные стандарта МЭК (IEC), создаваемые в структурных единицах программы (POU) и списках глобальных переменных (GVL) в процессе разработки приложения, которые не адресуется непосредственно в памяти. Символические переменные могут быть определены как сохраняемые или постоянные: в этом случае будут использоваться области памяти сохраняемых символических переменных либо памяти постоянных символических переменных соответственно. Система ПЛК выделяет переменные в данной области, в связи с чем пространство, доступное для размещения переменных, созданных пользователем, меньше, чем указано в таблице. Заполнение системных переменных зависит от характеристик проекта (количество модулей, драйверов и т.д.), поэтому рекомендуется следить за доступным пространством в сообщениях компиляции инструмента MasterTool IEC XE.

Постоянная и сохраняемая память символических переменных: Область, в которой размещаются сохраняемые символические переменные. Сохраняемые данные сохраняют свои значения даже после циклов выключения и включения питания центрального процессора (ЦП). Постоянные данные сохраняют свои значения даже после загрузки нового приложения в центральный процессор (ЦП).

**ВНИМАНИЕ:**

Объявление и использование символических постоянных переменных должно осуществляться исключительно через объект *Persistent Vars (Постоянные переменные)*, который может быть включен в проект через древовидное представление в *Application -> Add Object -> Persistent Variables (Приложение -> Добавить объект -> Постоянные переменные)*. Не следует использовать выражение *VAR PERSISTENT (Постоянная переменная)* при объявлении переменных поля структурных единиц программы (POU).

Полный список того, когда символические постоянные переменные сохраняют свои значения и когда значение теряется, приведен в таблице ниже. Помимо размера постоянной области, указанного в таблице выше, для хранения информации о постоянных переменных зарезервированы следующие 44 байта (не доступны для использования).

В таблице, приведенной ниже, представлена логика работы сохраняемых и постоянных переменных для различных ситуаций, в которых «-» означает потерю значения, а «X» - сохранение значения.

Команда	VAR (Переменная)	VAR RETAIN (Сохраняемая переменная)	VAR PERSISTENT (Постоянная переменная)
Теплый сброс / Цикл включения/выключения питания	-	X	X
Холодный сброс	-	-	X

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Сброс до исходного состояния	-	-	-
Извлеките центральный процессор (ЦП) или источник питания из стойки под напряжением	-	-	-
Скачать (загрузить)	-	-	X
Изменение в диалоговом режиме («онлайн»)	X	X	X
Перезагрузка ПЛК	-	X	X
Очистить все	-	-	X
Сброс процесса (МЭК 60870-5-104 (IEC 60870-5-104))	-	X	X

Таблица 7: Логика работы переменных после выполнения команд

В случае команды «Очистить все» (Clean All), если приложение было изменено таким образом, что постоянные переменные были удалены, вставлены в начало списка либо их тип был изменен, значение данных переменных будет потеряно (при запросе на загрузку инструментом MasterTool). Поэтому рекомендуется, чтобы изменения в списке глобальных переменных (GVL) постоянных переменных включали в себя только добавление новых переменных в список.

Память программ: Память программ представляет собой максимальный объем, который может быть использован для хранения пользовательского приложения. Данная область используется совместно с памятью исходного кода, в связи с чем общая площадь равна сумме «памяти программ» и «памяти исходного кода».

Память исходного кода (резервное копирование): Данная область памяти используется в качестве резервной копии проекта. Если пользователь захочет импортировать проект, MasterTool IEC XE получит необходимую информацию в данной области. Во избежание потери важной информации необходимо следить за тем, чтобы проект, сохраненный в качестве резервной копии, был актуальным. Данная область является общей с памятью исходного кода, в связи с чем общая площадь равна сумме «памяти программ» и «памяти исходного кода».

Память пользовательских файлов: Данная область памяти предоставляет пользователю еще один способ хранения файлов, таких как doc, pdf, изображений и других. Эта функция позволяет записывать данные, как на карту памяти. Дополнительную информацию см. в разделе «Память пользовательских файлов».

### 2.2.3. Протоколы

	NX3004	Интерфейс
Открытый протокол	Да	COM1
MODBUS RTU Master	Да	COM1
MODBUS RTU Slave	Да	COM1
MODBUS TCP Client	Да	NET1
MODBUS TCP Server	Да	NET1
MODBUS RTU через TCP Client	Да	NET1
MODBUS RTU через TCP Server	Да	NET1

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

CANopen Master	Нет	-
CANopen Slave	Нет	-
CAN low level	Нет	-
SAE J-1939	Нет	-
Сервер OPC DA	Да	NET1
Сервер OPC UA	Да	NET1
EtherCAT Master	Нет	-
SNMP Agent	Да	NET1
Сервер DNP3 (данные, ориентированные на события)	Нет	-
Сервер МЭК 60870-5-104 (IEC 60870-5-104)	Нет	-
EtherNet/IP Scanner	Да	NET1
	NX3004	Интерфейс
EtherNet/IP Adapter	Да	NET1
Клиент MQTT	Да	NET1
Клиент SNTP (для синхронизации часов)	Да	NET1
Контроллер PROFINET	Да	NET1
Устройство PROFINET	Нет	-

Таблица 8: Протоколы

Примечание:

Контроллер PROFINET: Включен для использования в простой (не кольцевой) сети с числом устройств до 8. В случае более крупных приложений, обратитесь в службу технической поддержки.

### 2.2.4. Последовательные интерфейсы

#### 2.2.4.1. COM 1

	COM 1
Разъем	Экранированный разъем типа мама (с внутренней резьбой) DB9
Физический интерфейс	RS-422 или RS-485 (в зависимости от выбранного кабеля)
Направление связи	RS-422: полный дуплекс RS-485: полный дуплекс
Максимальное количество приемопередатчиков RS-422	11 (1 передатчик и 10 приемников)
Максимальное количество приемопередатчиков RS-485	32

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Терминирование	Да (опционально через выбор кабеля)
Скорость передачи данных в бодах	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит / с
Изоляция Логика для последовательного порта Последовательный порт к защитному заземлению ⚡	1000 В переменного тока / 1 минута 1000 В переменного тока / 1 минута

Таблица 9: Характеристики последовательного интерфейса COM 1

### Примечания:

Физический интерфейс: В зависимости от конфигурации используемого кабеля можно выбрать вид физического интерфейса: RS-422 или RS-485. Список кабелей можно найти в разделе «[Родственные изделия](#)».

Максимальное количество приемопередатчиков RS-422: Это максимальное количество приемопередатчиков RS-422, которые могут использоваться на одной шине.

Максимальное количество приемопередатчиков RS-485: Это максимальное количество приемопередатчиков RS-485, которые могут использоваться на одной шине.

## 2.2.5. Интерфейсы локальной сети Ethernet

### 2.2.5.1. NET 1

	NET 1
Разъем	Экранированный разъем типа мама (с внутренней резьбой) RJ45
Функция автоматического определения типа кабеля	Да
Максимальная длина кабеля	100 м
Тип кабеля	UTP или ScTP, категория 5
Скорость передачи данных в бодах	10/100 Мбит/с
Физический уровень	10/100 BASE-TX (полный дуплекс)
Канальный уровень	LLC (управление логическими связями)
Сетевой уровень	IP (протокол сетевого уровня из набора протоколов Internet))
Транспортный уровень	TCP (протокол управления линией передачи данных) UDP (протокол передачи пользовательских датаграмм)
Диагностика	Светодиоды - зеленый (скорость), желтый (связь/активность)
Изоляция Интерфейс локальной сети Ethernet к последовательному порту	1500 В переменного тока / 1 минута

Таблица 10: Характеристики интерфейса локальной сети Ethernet NET 1

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

### 2.2.6. Источник питания



	Источник питания
Номинальное входное напряжение	24 В постоянного тока
Максимальная выходная мощность	15 Вт
Максимальный выходной ток	3 А
Входное напряжение	от 19,2 до 30 В постоянного тока
Максимальный входной ток (пусковой)	30 А
Максимальный входной ток	1,4 А
Время прерывания максимального входного напряжения	10 мс при 24 В постоянного тока
Изоляция Вход к логике	1000 В переменного тока / 1 минута
 Вход к защитному заземлению	1500 В переменного тока / 1 минута
 Вход к функциональному заземлению	1000 В переменного тока / 1 минута
Поперечное сечение (провода)	0,5 мм <sup>2</sup>
Защита от инверсии полярности	Да
Внутренний плавкий предохранитель автоматического восстановления	Да
Защита от короткого замыкания на выходе	Да
Защита от перегрузки по току	Да

Таблица 11: Характеристики источника питания

Примечание:

Максимальная выходная мощность: Используя модули входа/выхода NextoJet, можно расширить и получить возможность использования 20 Вт выходной мощности. Ограничения для использования данного предельного значения см. в Примечании по применению NAP152.

### 2.2.7. Характеристики окружающей среды

	NX3004
Потребление тока на шине источника питания	-
Мощность рассеяния (тока)	4 Вт
Рабочий диапазон температуры	От 0 до 60 °C
Температура хранения	От -25 до 75 °C
Относительная влажность	От 5% до 96%, без образования конденсата
Конформное (однородное) покрытие	Да
Степень защиты (IP)	IP 20
Габаритные размеры модуля (Ш x В x Г)	3,00 x 114,63 x 115,30 мм
Габаритные размеры упаковки (Ш x В x Г)	44,00 x 122,00 x 147,00 мм



## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Масса	350 г
Масса с упаковкой	400 г

Таблица 12: Характеристики окружающей среды

Примечания:

Конформное (однородное) покрытие электронных схем: Покрытие электронных схем защищает внутренние части изделия от влаги, пыли и других элементов, неблагоприятных для электронных схем.

### 2.3. Совместимость с другими изделиями

Чтобы разработать приложение для ЦП (центральных процессоров) серии Nexto, необходимо проверить версию MasterTool IEC XE. В следующей таблице указана минимальная требуемая версия (где были представлены контроллеры) и соответствующая версия микропрограммы (прошивки) на тот момент времени:

Центральные процессоры (ЦП) серии Nexto	MasterTool IEC XE	Версия микропрограммы (прошивки)
NX3004	от 2.01 до 2.09	от 1.5.0.18 до 1.7.17.0
NX3004	3.00 и выше	1.8.11.0 и выше

Таблица 13: Совместимость с другими изделиями

В дополнение к этому, в ходе разработки MasterTool IEC XE могут быть включены некоторые функции (например, специальные функциональные блоки и т.д...), которые могут ввести требование минимальной версии микропрограммы (прошивки). Во время загрузки приложения, MasterTool IEC XE проверяет версию микропрограммы (прошивки), установленную на контроллере, и, если она не соответствует минимальному требованию, то выводит сообщение с запросом на обновление. Последнюю версию микропрограммы (прошивки) можно загрузить с сайта компании Altus, и она полностью совместима с предшествующими приложениями.

### 2.4. Эксплуатационные характеристики

Эксплуатационные характеристики центральных процессоров (ЦП) серии Nexto зависят от:

- Время применения пользователем
- Интервал между применениями
- Время работы системы
- Количество модулей (данные процесса, вход/выход и др.)

#### 2.4.1. Время применения

Время выполнения приложения центрального процессора (ЦП) Nexto зависит от следующих переменных:

- Время считывания входных данных (локальных и удаленных)
- Время выполнения заданий
- Время записи выходных данных (локальных и удаленных)

Важно подчеркнуть, что на время выполнения задачи "MainTask («Главная задача») будет оказывать непосредственное влияние системная задача «Конфигурация» – задача с высоким приоритетом, периодически выполняемая системой. Задача «Конфигурация» может прерывать выполнение задачи "MainTask («Главная задача»), а при использовании коммуникационных модулей, например, модуля локальной сети Ethernet NX5000, добавление времени к задаче "MainTask («Главная задача») может составлять до 25% от среднего времени выполнения.

#### 2.4.2. Время, необходимое для выполнения команд

В таблице, приведенной ниже, представлено необходимое время выполнения различных команд в центральных процессорах (ЦП) серии Nexto.

Команда	Язык	Переменные	Время обработки команды (мкс)
1000 Contacts (1000 контактов)	LD	BOOL (БУЛЕВА)	6 < 5000
		INT 43	
		REAL 81	
		INT 43	
		REAL 81	
		INT 15	
		REAL 23	
		INT 15	
		REAL 23	
		INT 15	
		REAL 23	
		INT 15	
		REAL 23	
		1000 PID (1000 ПИД- регуляторов)	

Таблица 14: Время выполнения команд

### 2.4.3. Время инициализации

Время инициализации центральных процессоров (ЦП) серии Nexto составляет 50 секунд, а начальный экран с логотипом NEXTO (Splash) появляется через 20 секунд после включения питания.

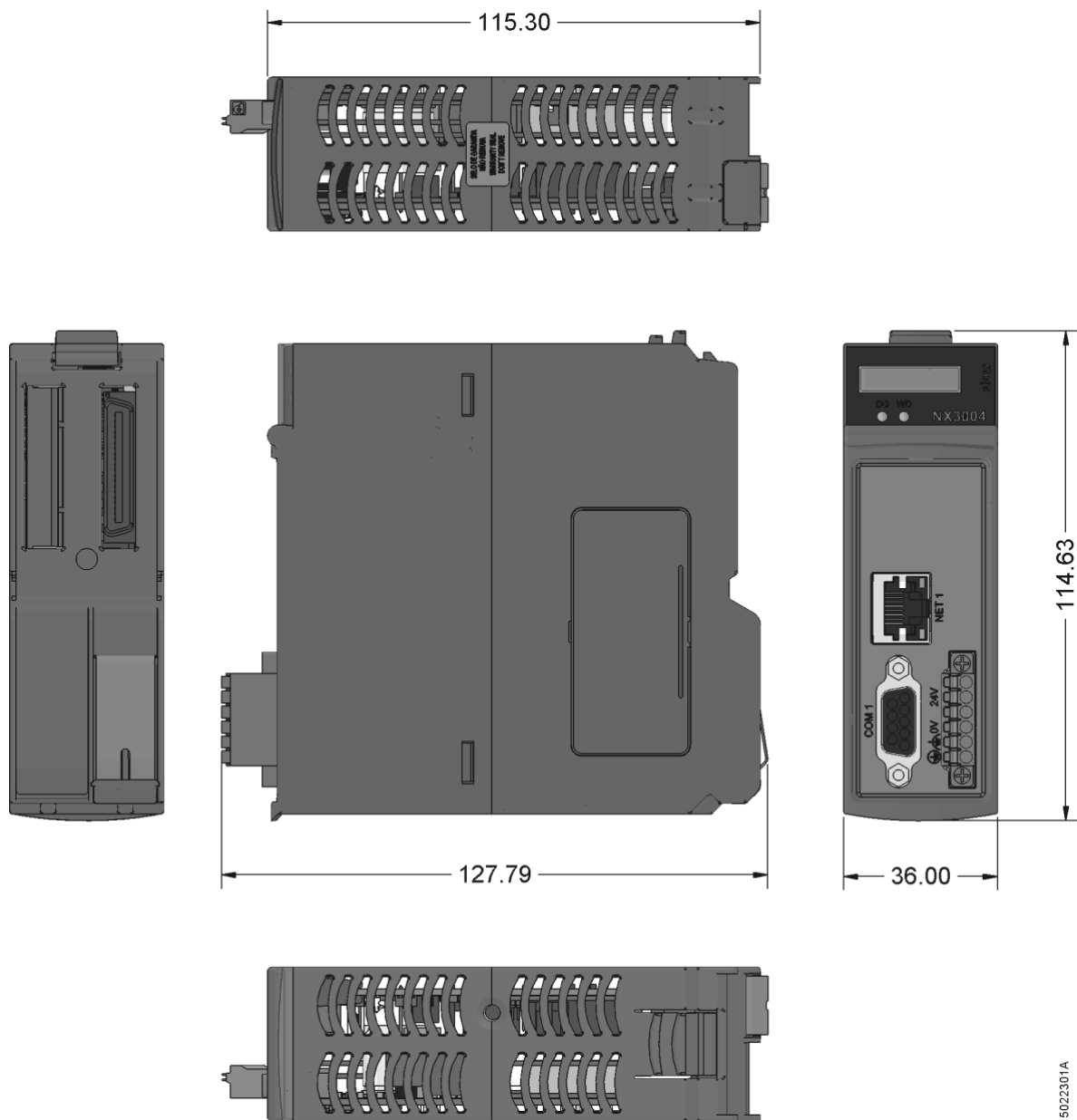
### 2.4.4. Временной интервал

Временной интервал выполнения основной задачи центрального процессора (ЦП) может быть установлен в диапазоне от 5 до 750 мс.

## 2.5. Фактические размеры

Габаритные размеры указаны в мм.

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ



15022301A

Рисунок 4: Фактические размеры центрального процессора (ЦП) NX3004 и NX3005

## 2.6. Сведения о покупке

### 2.6.1. Элементы, входящие в комплект

Упаковка изделия содержит следующие элементы:

- Модуль NX3004
- 6-контактный разъем с зажимом для проводов
- Руководство по монтажу

### 2.6.2. Код изделия

Для приобретения изделия следует использовать следующий код:

Код	Описание

NX3004	ЦП (центральный процессор), 1 порт локальной сети Ethernet, 1 последовательный канал, поддержка расширения дистанционной стойки и встроенный источник питания
--------	---

Таблица 15: Код изделия

## 2.7. Родственные изделия

При необходимости следующие изделия должны приобретаться отдельно:

Код	Описание
MT8500	MasterTool IEC XE
AL-2600	Ветвь сети связи и ограничитель RS-485
AL-2306	Кабель RS-485 для сети MODBUS или CAN
AL-2319	Кабель RJ45-RJ45
AL-1763	CMDB9-кабель клеммной колодки
NX9202	2м кабель RJ45-RJ45
NX9205	5м кабель RJ45-RJ45
NX9210	10м кабель RJ45-RJ45
NX9404	6-контактный зажим
NX9020	2-слотовое основание для сборки платы
NX9000	8-слотовая стойка объединительной платы
NX9001	12- слотовая стойка объединительной платы
NX9002	16- слотовая стойка объединительной платы
NX9003	24- слотовая стойка объединительной платы
NX9010	8-слотовая стойка объединительной платы (без «горячей» замены)

Таблица 16: Родственные изделия

Примечания:

MT8500: MasterTool IEC XE доступен в четырех различных версиях: LITE, BASIC, PROFESSIONAL и ADVANCED. Для получения более подробной информации просим ознакомиться с Руководством пользователя MasterTool IEC XE - MU299609.

AL-2600: Данный модуль используется для разветвления и терминирования сетей RS-422/485. Для каждого узла сети требуется модуль AL-2600. AL-2600, находящийся на концах сети, должен быть сконфигурирован с терминированием, за исключением случаев, когда имеется устройство с активным внутренним терминированием; остальные должны быть сконфигурированы без терминирования.

AL-2306: Две экранированные витые пары без разъемов, используются для сетей на базе RS-485 или CAN.

AL-2319: Два разъема RJ45 для программирования ЦП (центральных процессоров) серии Nexto и соединения точка-точка локальной сети Ethernet с другим устройством со связью интерфейса локальной сети Ethernet.

AL-1763: Кабель с одним разъемом типа папа (с наружной резьбой) DB9 и клеммной колодкой для связи между ЦП (центральными процессорами) серии Nexto и изделиями со стандартной клеммной колодкой RS-485/RS-422.

NX9202/NX9205/NX9210: Кабели, используемые для связи Ethernet и для соединения модулей расширения шины.

NX9404: 6-контактный зажим.

NX9020: 2-слотовое основание для сборки платы.

## 3. Монтаж

В данной главе представлены необходимые процедуры для физического монтажа центральных процессоров (ЦП) серии Nexto, а также меры предосторожности, которые следует предпринять при других видах монтажа на панели, где установлен центральный процессор (ЦП).

### ОСТОРОЖНО:

Если оборудование используется не по назначению, описанному в данном руководстве, защита, обеспечиваемая оборудованием, может быть нарушена.

### 3.1. Механический монтаж

Данное изделие должно быть установлено в стойку объединительной платы в позиции 0. Оно требует двух последовательных позиций, то есть использует позиции 0 и 1 стойки.

Механическая сборка данного модуля описана в руководстве пользователя серии Nexto – MU214600.

### 3.2. Электрический монтаж

#### ОПАСНО:

При выполнении любого рода монтажа в электрическом щите необходимо убедиться в том, что основной источник питания выключен.

На приведенном ниже рисунке показана электрическая схема центральных процессоров (ЦП) NX3004 и NX3005, установленных в стойку с объединительной платой серии Nexto.

Изображенное расположение разъемов носит исключительно иллюстративный характер.

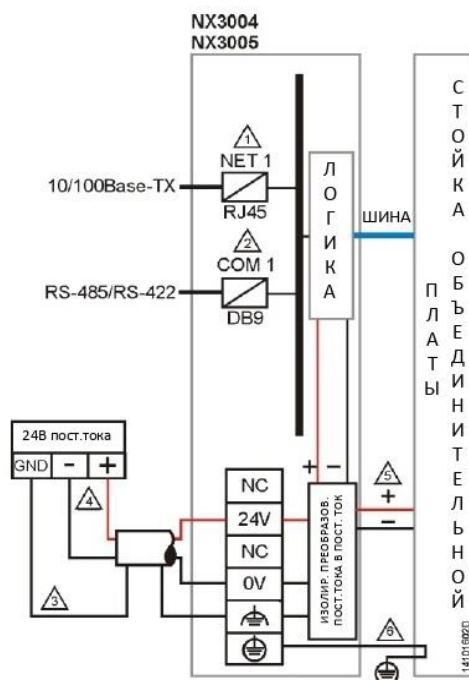


Рисунок 5: Электрическая схема центральных процессоров NX3004 и NX3005

Примечания к схеме:

1. Интерфейс сети Ethernet стандарта 10/100Base-TX для программирования, отладки и подключения к сети MODBUS TCP или других протоколов.
2. Последовательный интерфейс стандарта RS-485/RS-422 для сетевого подключения MODBUS RTU или других протоколов. Выбор физического интерфейса зависит от используемого кабеля.
3. Заземление от внешнего источника питания подключается к клемме . Используйте кабели сечением 0,5 мм<sup>2</sup>.

### 3. МОНТАЖ

4. Источник питания подключается к клеммам на 24 В и 0 В. Используйте кабели сечением 0,5 мм<sup>2</sup>.
5. Модуль питает другие модули серии Nexto через стоечное соединение.
6. Заземление модуля осуществляется посредством стойки объединительной платы серии Nexto.

## 3.3. Подключение к локальной сети Ethernet

Изолированный коммуникационный интерфейс NET 1 позволяет подключаться к локальной сети Ethernet, однако интерфейс NET 1 является наиболее подходящим для установления связи с MasterTool IEC XE.

Для подключения к локальной сети Ethernet используются кабели типа «витая пара» (10/100Base-TX), а определение скорости производится автоматически центральным процессором (ЦП) Nexto. Один конец кабеля должен быть подключен к интерфейсу, который будет использоваться, а другой - к концентратору (HUB), коммутатору, микрокомпьютеру или другой точке локальной сети Ethernet.

### 3.3.1. IP-адрес

Интерфейс локальной сети Ethernet NET 1 используется для связи по локальной сети Ethernet и для конфигурирования центрального процессора (ЦП), который поставляется со следующей конфигурацией параметров по умолчанию:

	NET 1
IP-адрес	192.168.15.1
Маска подсети	255.255.255.0
Адрес шлюза	192.168.15.253

Таблица 17: Конфигурация параметров по умолчанию для интерфейса локальной сети Ethernet NET 1

Параметры IP-адреса и маски подсети можно увидеть на графическом дисплее центрального процессора через меню параметров, как описано в разделе «[Информационное и конфигурационное меню центрального процессора \(ЦП\)](#)».

Первоначально интерфейс NET 1 должен быть подключен к сети ПК с той же маской подсети для связи с MasterTool IEC XE, где можно изменить параметры сети. Более подробная информация о конфигурации и изменении параметров приведена в разделе «[Конфигурация интерфейсов локальной сети Ethernet](#)».

### 3.3.2. Добровольный запрос ARP

Интерфейс локальной сети Ethernet NETx оперативным образом осуществляет рассылку ARP-пакетов широковещательного типа, сообщая свой IP- и MAC-адрес всем устройствам, подключенным к сети. Отправка данных пакетов осуществляется при загрузке нового приложения программой MasterTool IEC XE и при запуске центрального процессора (ЦП), когда приложение переходит в режим Run («Выполнение»).

Пять ARP-команд запускаются в течение начального интервала 200 мс с последующим удвоением интервала между каждой новой срабатывающей командой, что в сумме составляет 3 с. Пример: Первое срабатывание происходит в момент времени 0, второе - через 200 мс, третье - через 600 мс и так далее, пока не произойдет пятое срабатывание в момент времени 3 с.

### 3.3.3. Монтаж сетевого кабеля

Порты локальной сети Ethernet центральных процессоров (ЦП) серии Nexto, обозначенные на панели, как NET, имеют стандартную разводку выводов, аналогичную используемой в персональных компьютерах. Тип разъема, тип кабеля, физический уровень и другие детали, касающиеся центрального процессора и устройства локальной сети Ethernet, определены в разделе «[Интерфейсы локальной сети Ethernet](#)».

В таблице, приведенной ниже, представлен разъем типа мама (с внутренней резьбой) RJ-45 центрального процессора (ЦП) Nexto с идентификацией и описанием допустимой разводки выводов для физических уровней 10BASE-TE и 100BASE-TX.

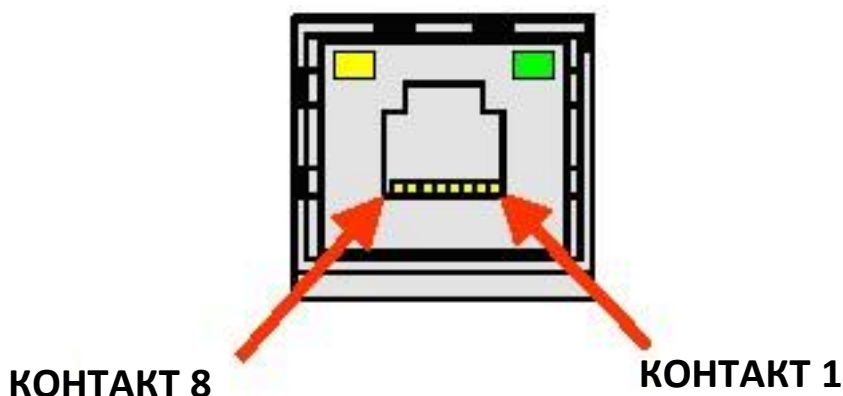


Рисунок 6: Разъем типа мама (с внутренней резьбой) RJ45

Контакт	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 18: Разводка выводов разъема типа мама (с внутренней резьбой) RJ45 - 10BASE-TE и 100BASE-TX

Интерфейс может быть подключен к коммуникационной сети посредством концентратора или коммутатора, либо непосредственно от коммуникационного оборудования. В последнем случае, благодаря функции автоматического определения типа кабеля центральных процессоров (ЦП) Nexto, нет необходимости в использовании кроссового (переходного) сетевого кабеля, который используется для соединения «точка-точка» двух ПК через порт локальной сети Ethernet.

Важно подчеркнуть, что под сетевым кабелем подразумевается пара разъемов типа папа (с наружной резьбой) RJ45, соединенных кабелем UTP или ScTP, категории 5, как путем прямого соединения, так и перекрестного. Он используется для связи двух устройств через порт локальной сети Ethernet.

Как правило, подобные кабели оснащены фиксатором соединения, который гарантирует идеальное соединение между разъемом типа мама (с внутренней резьбой) интерфейса и разъемом типа папа (с наружной резьбой) кабеля. В момент монтажа необходимо вставить разъем типа папа (с наружной резьбой) в разъем типа мама (с внутренней резьбой) модуля до щелчка, обеспечивающего срабатывание фиксатора. Для отсоединения кабеля от модуля необходимо с помощью рычага фиксатора отсоединить один от другого.

### 3.4. Последовательное сетевое подключение RS-485/422

Изолированный коммуникационный интерфейс COM 1 позволяет подключаться к сети RS-485/422. Ниже представлен разъем типа мама (с внутренней резьбой) DB9 для подключения к центральному процессору (ЦП) Nexto, с идентификацией и описанием сигналов.

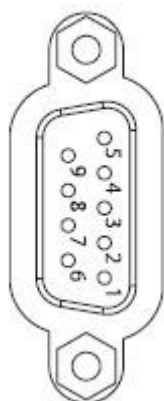


Рисунок 7: Разъем типа мама (с внутренней резьбой) DB9

Контакт	Сигнал	Описание
1	-	Не используется
2	Term+	Внутренне терминирование (оконечное устройство), положительное
3	TXD+	Передача данных, положительный
4	RXD+	Прием данных, положительный
5	GND	Отрицательный указатель для внешнего терминирования
6	+5V	Положительный указатель для внешнего терминирования
7	Term-	Внутренне терминирование (оконечное устройство), отрицательное
8	TXD-	Передача данных, отрицательный
9	RXD-	Прием данных, отрицательный

Таблица 19: COM 1 (NX3004/NX3005) и COM 2 (NX3010/NX3020/NX3030)

#### 3.4.1. Связь по RS-485 без терминирования

Для подключения в сети RS-485 без терминирования (использования оконечного устройства) идентифицированные клеммы кабеля AL-1763 должны быть подключены к соответствующим клеммам устройства, как показано в таблице, приведенной ниже.

Клеммы AL-1763	Сигналы клемм устройства
0	Экран
1	Не подключено
2	D+
3	D+
4	Не подключено
5	Не подключено
6	Не подключено
7	D-
8	D-

Таблица 20: Подключения RS-485 без терминирования



### 3. МОНТАЖ

На приведенном ниже схематическом изображении показано, каким образом следует подключать клеммы AL-1763 к клеммам устройства.

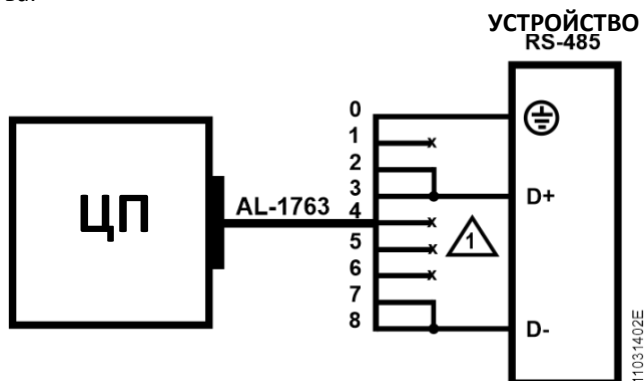


Рисунок 8: RS-485 Схема подключений RS-485 без терминирования

Примечание к схеме:

1. Неподключенные клеммы должны быть изолированы, чтобы они не соприкасались друг с другом.

#### 3.4.2. Связь по RS-485 с внутренним терминированием

Для подключения в сети RS-485 посредством внутреннего терминирования необходимо подключить идентифицированные клеммы кабеля AL-1763 к соответствующим клеммам устройства, как показано в таблице, приведенной ниже.

Клеммы AL-1763	Сигналы клемм центрального процессора (ЦП)
0	Экран
1	D+
2	D+
3	D+
4	Не подключено
5	Не подключено
6	D-
7	D-
8	D-

Таблица 21: Подключения RS-485 с внутренним терминированием

Примечание: Доступное внутреннее терминирование является типом безопасного состояния в открытом режиме.

На приведенном ниже схематическом изображении показано, каким образом следует подключать клеммы AL-1763 к клеммам устройства.

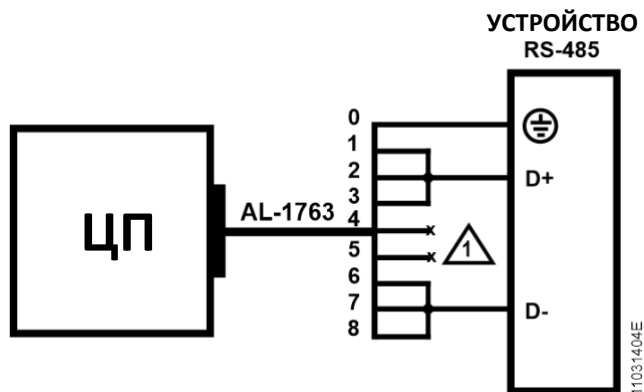


Рисунок 9: Схема подключений RS-485 с внутренним терминированием

Примечание к схеме:

1. Неподключенные клеммы должны быть изолированы, чтобы они не соприкасались друг с другом.

### 3.4.3. Связь по RS-485 с внешним терминированием

Для подключения в сети RS-485 с внешним терминированием идентифицированные клеммы кабеля AL-1763 должны быть подключены к соответствующим клеммам устройства в соответствии с таблицей, приведенной ниже.

Клеммы AL-1763	Сигналы клемм центрального процессора (ЦП)
0	Экран
1	Не подключено
2	D+
3	D+
4	0 V
5	+5 V
6	Не подключено
7	D-
8	D-

Таблица 22: RS-485 Подключения RS-485 с внешним терминированием

На приведенном ниже схематическом изображении показано, каким образом следует подключать клеммы AL-1763 к клеммам устройства.

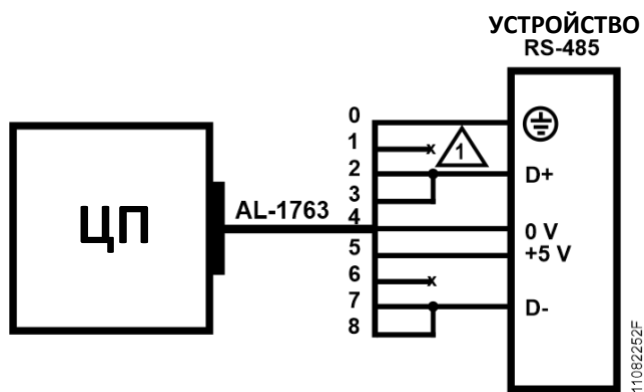


Рисунок 10: Схема подключений RS-485 с внешним терминированием

### 3. МОНТАЖ

Примечание к схеме:

1. Неподключенные клеммы должны быть изолированы, чтобы они не соприкасались друг с другом.

#### 3.4.4. Связь по RS-422 без терминирования

Для подключения в сети RS-422 без терминирования (использования оконечного устройства) идентифицированные клеммы кабеля AL-1763 должны быть подключены к соответствующим клеммам устройства, как показано в таблице, приведенной ниже.

Клеммы AL-1763	Сигналы клемм центрального процессора (ЦП)
0	Экран
1	Не подключено
2	TX+
3	RX+
Клеммы AL-1763	Сигналы клемм центрального процессора (ЦП)
4	Не подключено
5	Не подключено
6	Не подключено
7	TX-
8	RX-

Таблица 23: Подключения RS-422 без терминирования

На приведенном ниже схематическом изображении показано, каким образом следует подключать клеммы AL-1763 к клеммам устройства.

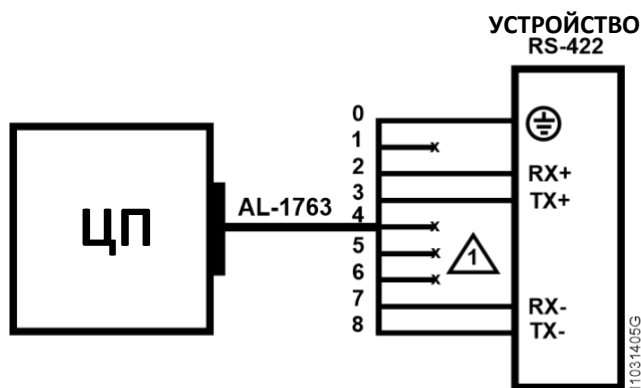


Рисунок 11: Схема подключений без терминирования

Примечание к схеме:

1. Неподключенные клеммы должны быть изолированы, чтобы они не соприкасались друг с другом.

#### 3.4.5. Связь по RS-422 с внутренним терминированием

Для подключения в сети RS-422 с использованием внутреннего терминирования необходимо подключить идентифицированные клеммы кабеля AL-1763 к соответствующим клеммам устройства, как показано в таблице, приведенной ниже.

Клеммы AL-1763	Сигналы клемм центрального процессора (ЦП)
0	Экран

### 3. МОНТАЖ

1	TERM+
2	TX+
3	RX+
4	Не подключено
5	Не подключено
6	TERM-
7	TX-
8	RX-

Таблица 24: Подключения RS-422 с внутренним терминированием

Примечание: Доступное внутреннее терминирование находится в безопасном состоянии в открытом режиме.

На приведенном ниже схематическом изображении показано, каким образом следует подключать клеммы AL-1763 к клеммам устройства.

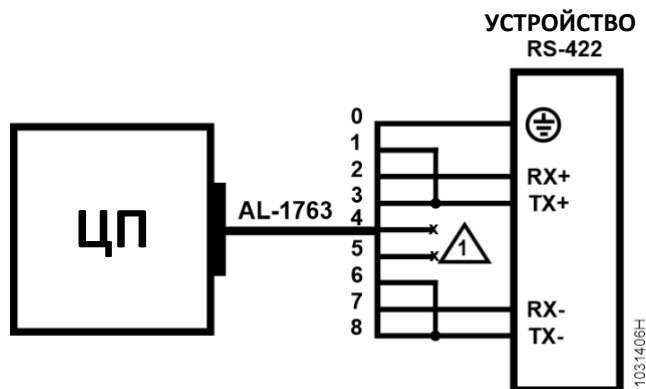


Рисунок 12: Схема подключений RS-422 с терминированием

Примечание к схеме:

1. Неподключенные клеммы должны быть изолированы, чтобы они не соприкасались друг с другом.

#### 3.4.6. Связь по RS-422 с внешним терминированием

Для подключения в сеть RS-422 с помощью внешнего терминирования интерфейса необходимо подключить идентифицированные клеммы кабеля AL-1763 в соответствующие клеммы устройства, как показано в таблице, приведенной ниже.

Клеммы AL-1763	Сигналы клемм центрального процессора (ЦП)
0	Экран
1	Не подключено
2	TX+
3	RX+
4	0 V
5	+5 V
6	Не подключено
7	TX-
8	RX-

Таблица 25: Подключения RS-422 с внешним терминированием

На приведенном ниже схематическом изображении показано, каким образом следует подключать клеммы AL-1763 к клеммам устройства.

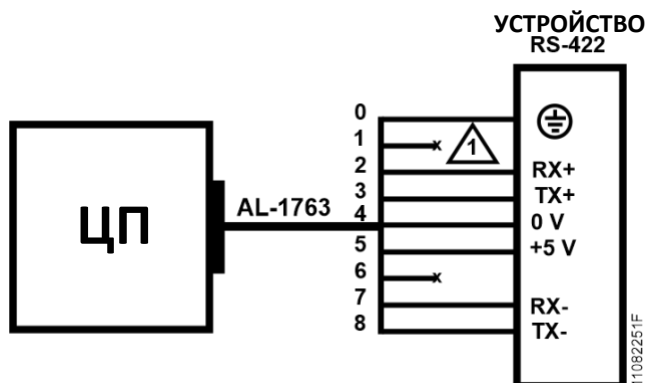


Рисунок 13: Схема подключений RS-422 с внешним терминированием

Примечание к схеме:

1. Неподключенные клеммы должны быть изолированы, чтобы они не соприкасались друг с другом.

### 3.4.7. Пример сети RS-422

На приведенном ниже рисунке показан пример использования сети RS-422 с использованием центрального процессора (ЦП) Nexto в качестве ведущего и ведомого устройств с интерфейсом RS-422, а также решений компании Altus для ограничителей и соединений.

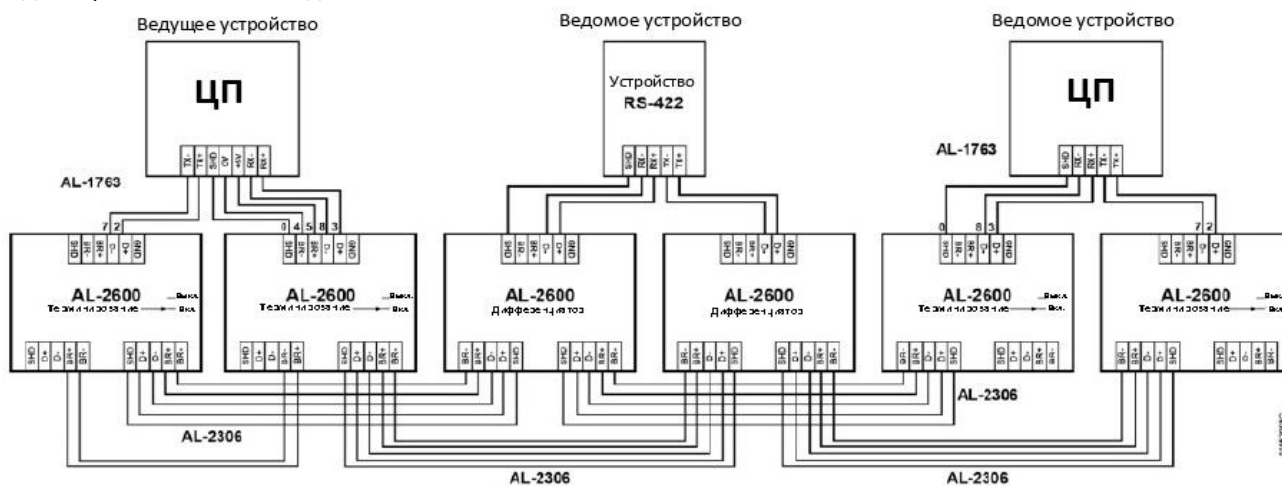


Рисунок 14: Пример сети RS-422

Примечание к схеме:

Модули AL-2600, находящиеся на концах сети, выполняют функцию ограничителей. В этом случае клавиши AL-2600 должны быть сконфигурированы в оконечном устройстве PROFIBUS.

## 3.5. Установка архитектуры

### 3.5.1. Установка модулей в главную стойку объединительной платы

В серии Nexto реализован эксклюзивный метод подключения и отключения модулей на шине, который не требует от оператора значительных усилий и гарантирует целостность соединения. Более подробная информация о креплении изделий серии Nexto приведена в руководстве пользователя серии Nexto – MU214600.

## 3.6. Установка устройства программирования (программатора)

Для выполнения установки программного обеспечения для разработки MasterTool IEC XE необходимо иметь дистрибутивный CD-ROM или загрузить установочный файл с сайта [http://www.altus.com.br/site\\_en/](http://www.altus.com.br/site_en/). Более подробная информация о пошаговом выполнении установки приведена в руководстве пользователя MasterTool IEC XE MT8500 – MU299609.

## 4. Начальное программирование

Основной целью данной главы является оказание помощи в программировании и конфигурировании центральных процессоров (ЦП) серии Nexto, позволяющей пользователю сделать первые шаги перед началом программирования устройства.

В центральных процессорах (ЦП) серии Nexto используется стандарт МЭК 61131-3 (IEC 61131-3) для языков программирования, которыми являются язык набора команд (IL), язык структурированного текста (ST), язык лестничных диаграмм (LD), язык последовательностных функциональных схем (SFC) и язык функциональных блоковых диаграмм (FBD), а также, помимо них, дополнительный язык – язык непрерывных диаграмм функций (CFC). Данные языки можно подразделить на текстовые и графические. Язык набора команд (IL) и язык структурированного текста (ST) являются текстовыми языками и похожи на Assembly и C, соответственно. Язык лестничных диаграмм (LD), язык последовательностных функциональных схем (SFC), язык функциональных блоковых диаграмм (FBD) и язык непрерывных диаграмм функций (CFC) являются графическими языками. Язык лестничных диаграмм (LD) использует релейно-блочное представление и похож на релейные схемы. Язык последовательностных функциональных схем (SFC) использует представление в форме диаграммы последовательности, что позволяет с легкостью увидеть последовательность событий. Язык функциональных блоковых диаграмм (FBD) и язык непрерывных диаграмм функций (CFC) используют группу функциональных блоков, позволяя 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, таким образом, пользователь может применять лучшие возможности, предлагаемые каждым языком, что приводит к более эффективной разработке приложений, облегчает документальное оформление и дальнейшее техническое обслуживание.

Дополнительную информацию по программированию см. в Руководстве пользователя MasterTool IEC XE – MU299609, Руководстве по программированию MasterTool IEC XE – MP399609 либо в стандарте МЭК 61131-3 (IEC 61131-3).

### 4.1. Организация памяти и доступ к ней

В серии Nexto используется инновационная технология организации памяти и доступа к ней, называемая big-endian (порядок байтов от старшего к младшему), при которой старший байт хранится первым и всегда является младшим адресом (например, %QB0 всегда будет более значимым, чем %QB1, как в таблице, приведенной ниже, где для строки CPUNEXTO буква C – байт 0, а буква O – байт 7).

Кроме того, доступ к памяти должен осуществляться аккуратно, поскольку переменные с большим количеством разрядов (WORD, DWORD, LONG) используют в качестве индекса старший байт, другими словами, %QD4 всегда будет иметь в качестве старшего байта %QB4. Поэтому не нужно будет производить вычисления, чтобы узнать, какие DWORD соответствуют определенным байтам. В таблице, приведенной ниже, показана организация порядка байтов от младшего к старшему (little endian) и от старшего к младшему (big endian).

MSB ← Порядок байтов от младшего к старшему (little endian) → 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 ← Порядок байтов от старшего к младшему (Big-endian) → LSB								
BYTE	%QB0	%QB1	%QB2	%QB3	%QB4	%QB5	%QB6	%QB7
	C	P	U	N	E	X	T	O

---

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

WORD	%QW0	%QW2	%QW4	%QW6
	CP	UN	EX	TO
DWORD	%QD0		%QD4	
	CPUN		EXTO	
LWORD	%QL0			
	CPUNEXTO			

Таблица 26: Пример организации памяти и доступа к ней



4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

		SIGNIFICANCE					OVERLAPPING					
		Bit	Byte	Word	DWord	LWord	Byte	Word	DWord			
		%QX0.7										
		%QX0.6										
		%QX0.5										
		%QX0.4	%QB				%QB00					
		%QX0.3	00									
		%QX0.2										
		%QX0.1						%QW				
		%QX0.0		%QW								
		%QX1.7		00				00				
		%QX1.6										
		%QX1.5										
		%QX1.4	%QB				%QB01					
		%QX1.3	01									
MSB		%QX1.2										
		%QX1.1						%QW		%QD		
↑		%QX1.0			%QD							
		%QX2.7			00			01		00		
		%QX2.6										
		%QX2.5										
		%QX2.4	%QB				%QB02					
		%QX2.3	02									
		%QX2.2										
		%QX2.1						%QW		%QD		
		%QX2.0		%QW								
		%QX3.7		02				02		01		
		%QX3.6										
		%QX3.5										
		%QX3.4	%QB				%QB03					
		%QX3.3	03									
		%QX3.2										
		%QX3.1						%QW		%QD		
		%QX3.0				%QL						
		%QX4.7				00		03		02		
		%QX4.6										
		%QX4.5										
		%QX4.4	%QB				%QB04					
		%QX4.3	04									
		%QX4.2						%QW			%QD	
		%QX4.1										
		%QX4.0		%QW								
		%QX5.7		04				04			03	
		%QX5.6										
		%QX5.5										
		%QX5.4	%QB				%QB05					
		%QX5.3	05									
MSB		%QX5.2										
		%QX5.1										
↑		%QX5.0			%QD			%QW			%QD	
		%QX6.7			04			05				04
LSB		%QX6.6										
		%QX6.5										
		%QX6.4	%QB				%QB06					
		%QX6.3	06									
		%QX6.2										
		%QX6.1						%QW				
		%QX6.0		%QW								
		%QX7.7		06				06				
		%QX7.6										
		%QX7.5										
		%QX7.4	%QB				%QB07					
		%QX7.3	07									
		%QX7.2										
		%QX7.1										
		%QX7.0										

Таблица 27: Организация памяти и доступ к ней

В приведенной выше таблице показана организация и доступ к памяти, демонстрирующая значение байтов и расположение других типов переменных, включая перекрытие.

## 4.2. Профили проектов

Профиль проекта в MasterTool IEC XE состоит из шаблона приложения и группы правил верификации, которые направляют разработку приложения, снижая сложность программирования. Приложения могут быть созданы в соответствии со следующими профилями:

- Одиночный (Single)
- Базовый (Basic)
- Обычный (Normal)
- Экспертный (Expert)
- Настраиваемый (Custom)
- Машинный профиль (Machine Profile)

Выбор профиля проекта производится в мастере создания проекта. Каждый профиль проекта определяет шаблон стандартных имен для задач и программ, которые предварительно создаются в соответствии с выбранным профилем проекта. Помимо этого, 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 уклону сложности. Исходя из этих определений, пользователю рекомендуется всегда использовать наиболее простой профиль, отвечающий потребностям его приложения, переходя к более сложному профилю только тогда, когда соответствующие правила становятся больше препятствием для разработки, чем дидактическим упрощением. Важно отметить, что средство программирования позволяет изменять профиль из существующего проекта (см. раздел «Обновление проекта» в Руководстве пользователя MasterTool IEC XE – MU299609). Тем не менее, разработчик должен внести все необходимые изменения, чтобы проект стал совместим с правилами нового выбранного профиля.

**ВНИМАНИЕ:**

При описании профилей проектов упоминаются некоторые типы задач, которые описаны в разделе «Конфигурация задач» руководства пользователя MasterTool IEC XE - MU299609.

**ВНИМАНИЕ:**

При использовании более одной задачи доступ к вводу/выводу может осуществляться только в контексте MainTask («Главная задача»). В случае, если опция «Включить обновление ввода-вывода для каждой задачи» (Enable I/O update per task) не может быть использована, то она присутствует, начиная с версии MasterTool IEC XE 2.01.

### 4.2.1. Одиночный (Single)

В профиле проекта Single («Одиночный») приложение имеет только одну пользовательскую задачу – MainTask («Главная задача»). Данная задача отвечает за выполнение одной единицы программирования типа Program («Программа»), называемой MainPrg. Эта единственная программа может вызывать другие единицы программирования типа Program («Программа»), Function («Функция») или Function Block («Функциональный блок»), однако весь код будет выполняться только MainTask.

В данном профиле MainTask («Главная задача») будет принадлежать к циклическому типу (Cyclic) с приоритетом 13 (тринадцать) и будет запускаться в непрерывном цикле исключительно программу MainPrg. MainTask уже полностью определена, и разработчику необходимо создать программу MainPrg, используя любой из языков стандарта МЭК 61131-3 (IEC 61131-3). Не всегда возможно перевести программу на другой язык, но всегда можно создать новую программу, построенную на другом языке, с тем же именем и заменить ее. Стандартная опция MasterTool IEC XE заключается в использовании стандартного проекта MasterTool, связанного с профилем Single («Одиночный»), в который также входит MainPrg, созданный на языке, выбранном при создании проекта.

В данном типе приложений нет необходимости рассматривать такие вопросы, как согласованность данных, совместное использование ресурсов или механизмы взаимного исключения.

Задача	Структурная единица программы (POU)	Приоритет	Тип	Интервал	Событие
--------	-------------------------------------	-----------	-----	----------	---------

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

MainTask	MainPrg	13	Циклический	20 мс	-
----------	---------	----	-------------	-------	---

Таблица 28: Задача профиля Single («Одиночный»)

#### 4.2.2. Базовый (Basic)

В профиле проекта Basic («Базовый») приложение имеет одну пользовательскую задачу типа Continuous («Непрерывный») под названием MainTask («Главная задача»), которая выполняет программу в непрерывном цикле (без определения времени цикла) с приоритетом, фиксированным на 13 (тринадцать). Данная задача отвечает за выполнение одной структурной единицы программы (POU) единицы программирования под названием MainPrg. Важно отметить, что время цикла может меняться в зависимости от количества используемых коммуникационных задач, поскольку в этом режиме основная задача прерывается коммуникационными задачами.

Данный профиль также позволяет включить две событийные задачи с более высоким приоритетом, которые могут прервать (упредить) основную задачу в любой момент времени: задача с именем ExternInterruptTask00 является событийной задачей типа External («Внешний») с приоритетом 02 (два); задача с именем TimeInterruptTask00 является событийной задачей типа Cyclic («Циклический») с приоритетом 01 (один).

Модель шаблона проекта Basic («Базовый») включает в себя три уже полностью определенные задачи, представленные в таблице, приведенной ниже. Разработчику необходимо только создать соответствующие программы.

Задачи	Структурная единица программы (POU)	Приоритет	Тип	Интервал	Событие
MainTask	MainPrg	13	Непрерывный	-	-
ExternInterruptTask00	ExternInterruptPrg00	02	Внешний	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	20 мс	-

Таблица 29: Задачи профиля Basic («Базовый»)

#### 4.2.3. Обычный (Normal)

В профиле проекта Normal («Обычный») приложение имеет одну пользовательскую задачу типа Cyclic («Циклический»), называемую MainTask. Данная задача отвечает за выполнение одной структурной единицы программы (POU) единицы программирования под названием MainPrg. Данная программа и эта задача аналогичны единственной задаче и единственной программе профиля Single («Одиночный»), но здесь в приложение могут быть интегрированы дополнительные пользовательские задачи. Эти дополнительные задачи, названные CyclicTask00 и CyclicTask01, отвечают за эксклюзивное исполнение соответствующей программы CyclicPrg<nn>. Задачи CyclicTask<nn> всегда имеют циклический тип и приоритет 13 (тринадцать), такой же, как и у MainTask. Эти два типа образуют группу, называемую базовыми задачами, связанные с которыми программы могут вызывать другие структурные единицы программы (POU) типа Program («Программа»), Function («Функция») и Function Block («Функциональный блок»).

Помимо этого, в данный профиль могут входить событийные задачи с более высоким приоритетом, чем у базовых задач, которые могут в любой момент прервать (упредить) выполнение этих задач.

Задача ExternInterruptTask00 представляет собой событийную задачу типа External («Внешний»), выполнение которой инициируется каким-либо внешним событием, например, изменением управляющего сигнала на последовательном порту или изменением цифрового входа на шине NEXTO. Приоритет данной задачи установлен на 02 (два): она отвечает исключительно за выполнение программы ExternInterruptPrg00. Задача под названием TimeInterruptTask00 является событийной задачей циклического типа (Cyclic) с приоритетом 01 (один), отвечающей исключительно за выполнение программы TimeInterruptPrg00.

В модели проекта Normal («Обычный») пять задач и их структурные единицы программы (POU) уже полностью определены, как показано в таблице, приведенной ниже. Разработчику остается только реализовать содержание программ, выбрав в мастере любой из языков стандарта МЭК 61131-3 (IEC 61131-3). Интервал выполнения задач и триггерные события могут быть настроены разработчиком, а ненужные задачи могут быть исключены.

Задачи	Структурная единица программы (POU)	Приоритет	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-

CyclicTask00	CyclicPrg00	13	Циклический	200 мс	-
CyclicTask01	CyclicPrg01	13	Циклический	500 мс	-
ExternInterruptTask00	ExternInterruptPrg00	02	Внешний	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	20 мс	-

Таблица 30: Задачи профиля Normal («Обычный»)

#### 4.2.4. Экспертный (Expert)

Профиль проекта Expert («Экспертный») включает в себя те же базовые задачи CyclicTask<nn>, ExternInterruptTask00 и TimeInterruptTask00 с теми же приоритетами (13, 02 и 01 соответственно), но является расширением по сравнению с предыдущими за счет приема задач с несколькими событиями. Иными словами, в приложение могут быть включены различные задачи ExternInterruptTask<nn> или TimeInterruptTask<nn>, которые выполняют программы ExternInterruptPrg<nn> и TimeInterruptPrg<nn>. Приоритеты дополнительных событийных задач могут быть выбраны произвольно в диапазоне от 08 до 12. В данном профиле, за исключением стандартных программ, каждая задача может выполнять дополнительные программы.

В данном профиле проекта приложение может также включать пользовательскую задачу FreeTask типа Freewheeling («Свободный ход») с приоритетом 31, отвечающую за выполнение программы FreePrg. Поскольку данная задача имеет низкий приоритет, она может быть прервана всеми остальными, чтобы выполнить коды, которые могут быть заблокированы.

Уже полностью определены восемь задач, как показано в таблице, приведенной ниже, а также связанные с ними программы на выбранном языке. Интервалы и триггерные события любой задачи, а также приоритеты событийных задач могут быть настроены пользователем.

При разработке приложения с использованием профиля проекта Expert («Экспертный») особое внимание необходимо уделить масштабированию событийных задач. Если между данными задачами либо между ними и базовыми задачами происходит обмен информацией и ресурсами, настоятельно рекомендуется использовать стратегии, обеспечивающие согласованность данных.

Задачи	Структурная единица программы (POU)	Приоритет	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-
CyclicTask00	CyclicPrg00	13	Циклический	200 мс	-
CyclicTask01	CyclicPrg01	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 31: Задачи профиля Expert («Экспертный»)

#### 4.2.5. Настраиваемый («Custom»)

Профиль проекта Custom («Настраиваемый») позволяет разработчику использовать весь потенциал системы исполнения, реализованной в центральных процессорах серии Nexto. При этом не отключается никакая функциональность, не навязываются приоритеты, ассоциации и номенклатуры задач и программ. Исключение составляет только MainTask, которая всегда должна существовать под таким именем в данном профиле.

Помимо задач реального времени с приоритетом от 00 до 15, которые планируются по приоритету, в данном профиле также можно определить задачи с более низкими приоритетами в диапазоне от 16 до 31. В этом диапазоне используется Completely Fair Scheduler («Полностью честный планировщик») (работа с разделением времени), который необходим для выполнения кодов, которые могут быть заблокированы (например, использование сокетов (разъемов центрального процессора)).

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Разработчик волен частично следовать или не следовать организации, определенной в других профилях проекта, в зависимости от характеристик приложения. С другой стороны, модель Custom («Настраиваемая»), связанная с данным профилем, не нуждается в предопределяющих элементах, таких как задача, программа или параметр, оставляя разработчику возможность самому создавать все элементы, составляющие приложение. Тем не менее, пользователь может генерировать те же элементы, которые доступны для профиля Expert («Экспертный»).

Задачи	Структурная единица программы (POU)	Приоритет	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-
CyclicTask00	CyclicPrg00	13	Циклический	200 мс	-
CyclicTask01	CyclicPrg01	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: Задачи профиля Custom («Настраиваемый»)

#### 4.2.6. Машинный профиль (Machine Profile)

По умолчанию в машинном профиле (Machine Profile) приложение имеет пользовательскую задачу типа Cyclic («Циклический») под названием MainTask. Данная задача отвечает за реализацию единственной структурной единицы программы (POU) типа Program («Программа»), называемого MainPrg. Данная программа может вызывать другие единицы программирования типа Program («Программа»), Function («Функция») и Function Block («Функциональный блок»), но любой пользовательский код будет выполняться исключительно MainTask.

Данный профиль характеризуется тем, что допускает более короткие интервалы в MainTask, что позволяет быстрее выполнять пользовательский код. Подобного рода оптимизация возможна благодаря тому, 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.

Данный профиль может дополнительно включать в себя задачу прерывания, называемую TimeInterruptTask00, которая имеет более высокий приоритет, чем MainTask, и, следовательно, может прервать ее выполнение в любой момент времени.

Задачи	Структурная единица программы (POU)	Приоритет	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	4 мс	-

Таблица 33: Задачи машинного профиля

Данный профиль также поддерживает включение дополнительных задач, связанных с внешними прерываниями.

#### 4.2.7. Общая таблица

Проверки		Профили проекта					
		Одиночный	Машинный	Базовый	Обычный	Экспертный	Настраиваемый
Общее число задач	Количество	01	04	[01..03]	[01..32]	[01..32]	[01..32]
Число программ на одну задачу	Количество	01		01	01	<n>	<n>

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Главная задача	Имя						
	Тип	Циклический	Циклический	Непрерывный	Циклический	Циклический	Циклический
	Приоритет	13	13	13	13	13	13
	Количество	01	01	01	01	01	01
Задача с прерыванием по времени	Имя						
	Тип		Циклический	Циклический	Циклический	Циклический	Циклический
	Приоритет		01	01	01	01 или [08..12]	01 или [08..12]
	Количество		[00..01]	[00..01]	[00..01]	[00..31]	[00..31]
Задача с внешним прерыванием	Имя						
	Тип		Внешний	Внешний	Внешний	Внешний	Внешний
	Приоритет		02	02	02	02 или [08..12]	02 или [08..12]
	Количество		[00..01]	[00..01]	[00..01]	[00..31]	[00..31]
Циклическая задача	Имя				CyclicTask<nn>	CyclicTask<nn>	CyclicTask<n>
	Тип				Циклический	Циклический	Циклический
	Приоритет				13	13	13
	Количество				[00..31]	[00..31]	[00..31]
Свободная задача	Имя					FreeTask	FreeTask
	Тип					Свободный ход	Свободный ход
	Приоритет					31	31
	Количество					[00..01]	[00..01]
Событийная задача	Имя						<n>
	Тип						Событийный
	Приоритет						<n>
	Количество						[00..31]

Таблица 34: Общая таблица 'Профиль x Задачи'

**ВНИМАНИЕ:**

Предлагаемые имена структурных единиц программы (POU), связанные с задачами, не являются постоянными. Они могут быть изменены, если они также были изменены в конфигурациях задач.

#### 4.2.8. Максимальное число задач

Максимальное число задач, которые может создать пользователь, определено только для профиля Custom («Настраиваемый»), единственного, который имеет такое право. В остальных профилях задачи уже созданы и

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

настроены. Тем не менее, создаваемые задачи должны использовать следующие префиксы, в соответствии с типом каждой из них: CyclicTaskxx, TimeInterruptTaskxx, ExternInterruptTaskxx, где xx – номер создаваемой задачи.

В таблице, приведенной ниже, описано максимальное количество задач стандарта МЭК (IEC) для каждого центрального процессора (ЦП) и профиля проекта, где экземпляры протоколов также рассматриваются центральным процессором как коммуникационные задачи.

	Тип задачи	NX3003 / NX3004 / NX3005					
		S	B	N	E	P	M
Задача конфигурирования (задача WHSB)	Циклический	1	1	1	1	1	0
Пользовательские задачи	Циклический	1	1	15	15	15	4
	Срабатывание по событию	0	0	0	0	15	0
	Срабатывание по внешнему событию	0	1	1	14	15	2
	Свободный ход	0	1	0	1	15	0
	Срабатывание по состоянию	0	0	0	0	15	0
NET – экземпляры клиента или сервера	Циклический	4					
COM (n) – экземпляры ведущего или ведомого устройства	Циклический	1					
ИТОГО		16					

Таблица 35: Максимальное число задач по стандарту МЭК (IEC)

Примечания:

Условные обозначения профиля: Буквы S, B, N, E, C и M соответствуют профилям Simple («Простой»), Basic («Базовый»), Normal («Обычный»), Experienced («Экспертный»), Custom («Настраиваемый») и Machine («Машинный») соответственно.

Значения: Числа, определенные для каждого типа задач, представляют собой (максимально допустимые значения).

Задача WHSB: Задача WHSB, являющаяся системной, должна быть рассмотрена таким образом, чтобы она не превышала либо не получила полное значение.

NET – Экземпляры клиента или сервера: Определенное максимальное значение учитывает все системные интерфейсы локальной сети Ethernet, включая модули расширения, если таковые применяются. Примером такого рода задач может служить проектирование протокола MODBUS.

COM (n) – Экземпляры ведущего или ведомого устройства: Символ «n» означает номер последовательного интерфейса. Даже при использовании модулей расширения значение таблицы будет максимальным для каждого интерфейса. Примером такого рода задач может служить проектирование протокола MODBUS.

Итого: Общее значение представляет собой не сумму всех задач профиля, а максимально допустимое значение для каждого центрального процессора (ЦП). Таким образом, пользователь может создать несколько типов задач, при условии, что количество, установленное для каждого из них, и общее значение не превышены.

### 4.3. Конфигурация центрального процессора (ЦП)

Конфигурация центрального процессора (ЦП) Nexto находится в дереве устройств, как показано на рисунке, приведенном ниже, и доступна при двойном щелчке кнопкой мыши по соответствующему объекту. На данной вкладке можно настроить область диагностики, область сохраняющей и постоянной памяти, режим «горячей» замены, а также другие параметры, как описано в разделе «[Конфигурация центрального процессора \(ЦП\)](#)».

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

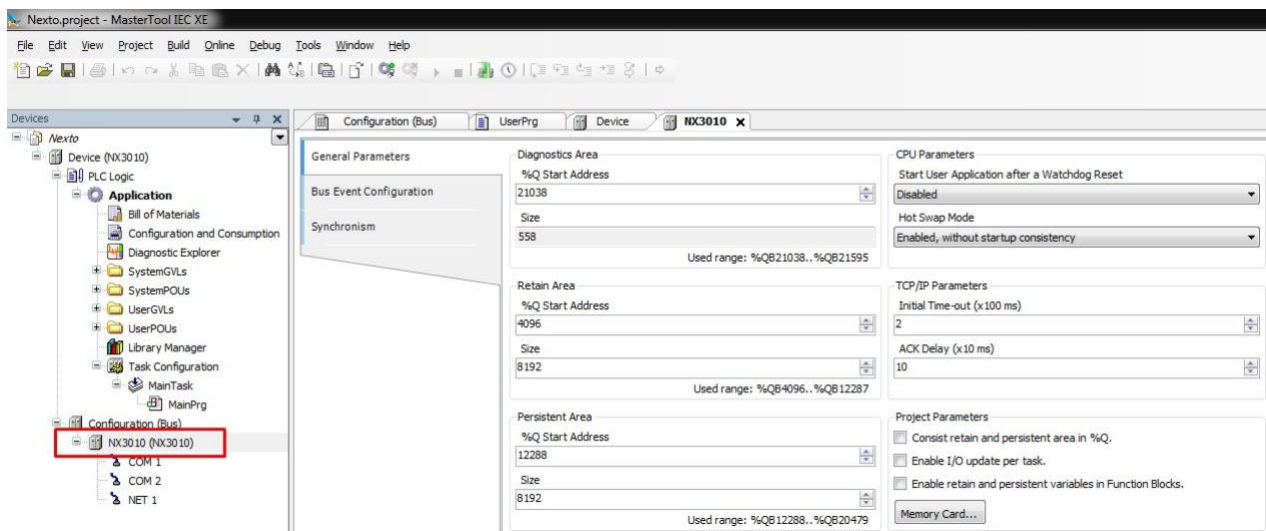


Рисунок 15: Конфигурация центрального процессора (ЦП)

Помимо этого, при выполнении двойного щелчка кнопкой мыши по пиктограмме NET 1 центрального процессора (ЦП) можно настроить интерфейс локальной сети Ethernet, который будет использоваться для связи между контроллером и программным обеспечением MasterTool IEC XE.

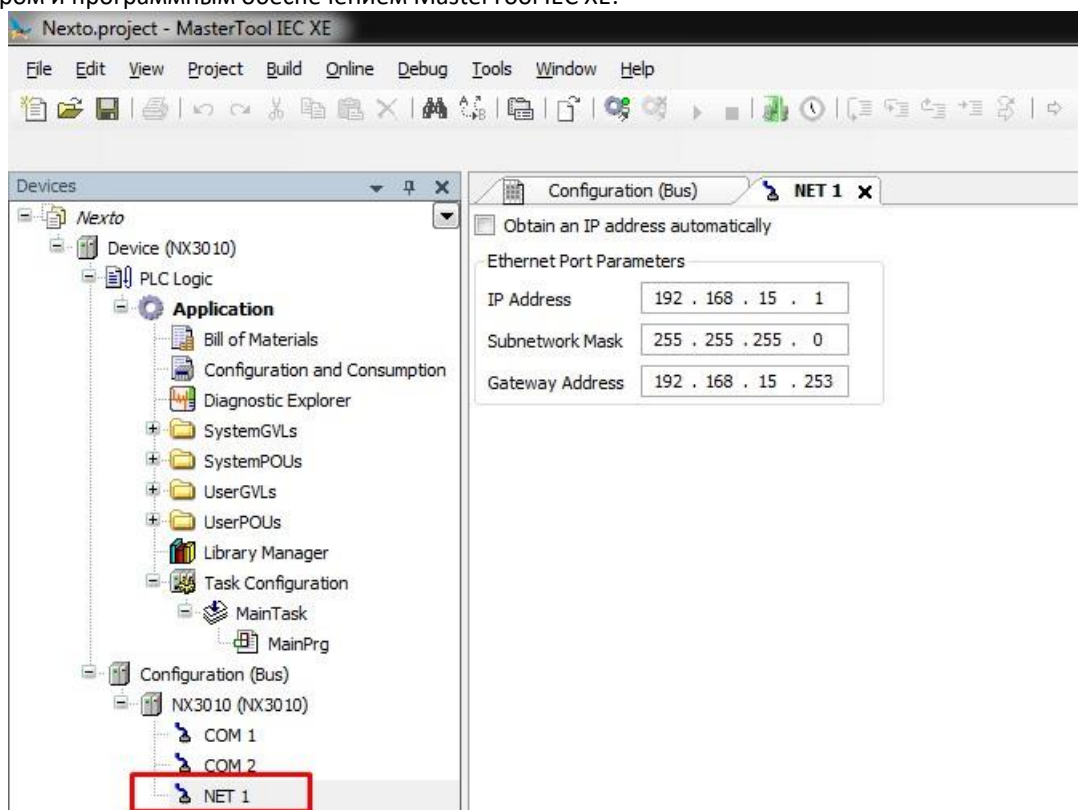


Рисунок 16: Конфигурирование коммуникационного порта ЦП

Конфигурация, заданная на данной вкладке, будет применена к устройству только при отправке приложения на устройство (загрузка), что описано далее в разделах «Поиск устройства» и «Вход в систему».



## 4.4. Библиотеки

Существует несколько ресурсов инструментов программирования, которые доступны через библиотеки. Поэтому данные библиотеки должны быть вставлены в проект, чтобы их использование стало возможным. Процедура вставки и дополнительная информация о доступных библиотеках приведена в руководстве по программированию MasterTool – MP399609.

## 4.5. Вставка экземпляра протокола

Центральные процессоры (ЦП) серии Nexto, как описано в разделе «Протоколы», предлагают несколько коммуникационных протоколов. За исключением протоколов OPC DA и OPC UA, которые имеют другую процедуру конфигурирования, вставка протокола может быть выполнена простым щелчком правой кнопки мыши по нужному коммуникационному интерфейсу, выбором добавления устройства и выполнением конфигурации, как показано в разделе «Конфигурация протоколов». Ниже приведены некоторые примеры.

### 4.5.1. Протокол MODBUS RTU

Первым шагом для конфигурирования MODBUS RTU в режиме ведомого устройства (slave) является включение экземпляра в нужный COM (в данном случае COM 1) путем нажатия правой кнопки мыши на COM и выбора пункта «Add Device...» («Добавить устройство...»), как показано на рисунке, приведенном ниже:

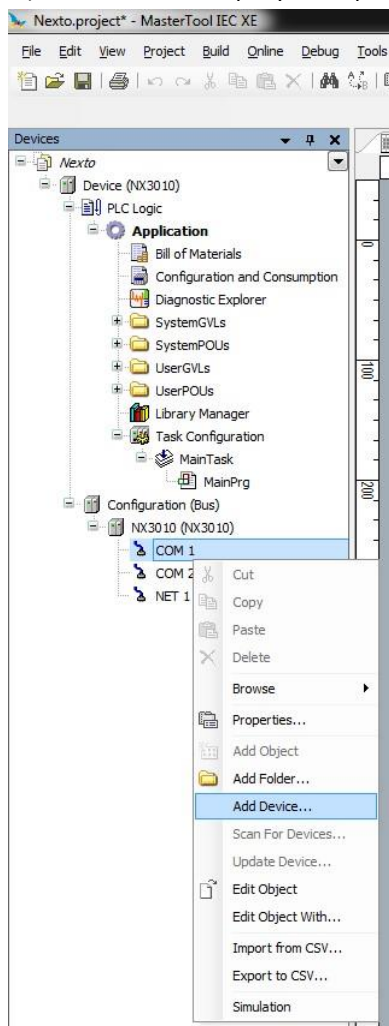


Рисунок 17: Добавление экземпляра

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

После этого на экране появятся доступные пользователю протоколы. Определите режим настройки протокола, выбрав *MODBUS Symbol RTU Slave* («Ведомое устройство MODBUS Symbol RTU»), для настройки *символьного преобразования* либо *MODBUS RTU Slave* («Ведомое устройство MODBUS RTU»), для *прямой адресации (%Q)*, и нажмите кнопку «Add Device» («Добавить устройство»), как показано на рисунке, приведенном ниже.

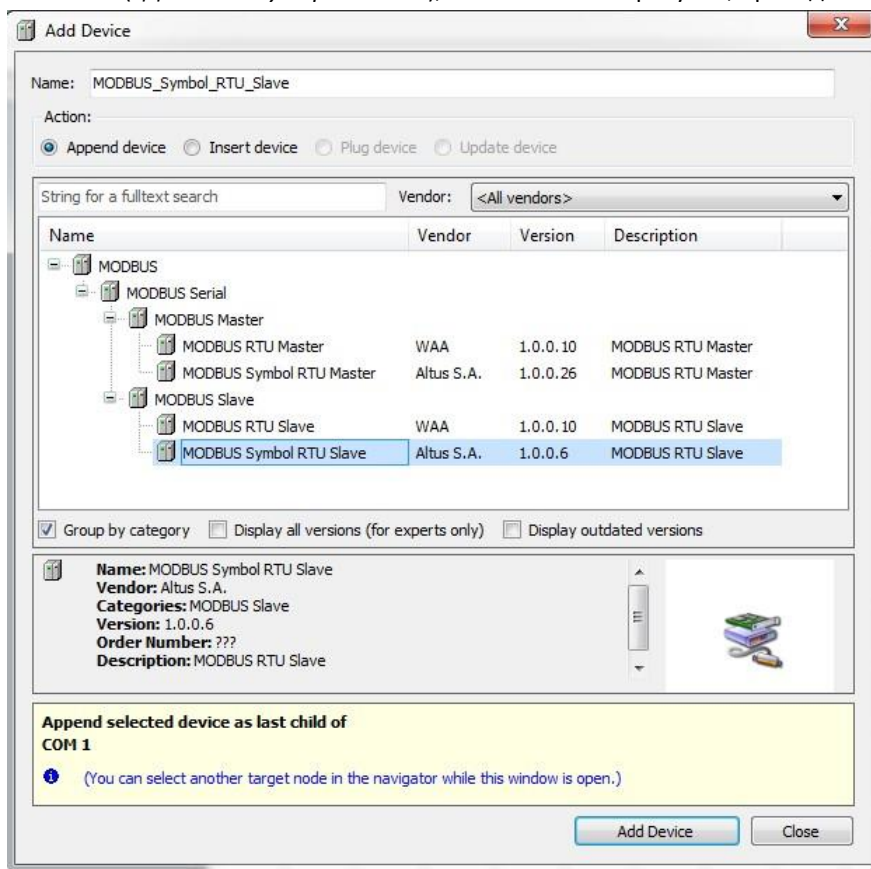


Рисунок 18: Выбор протокола

#### 4.5.2. Протокол MODBUS Ethernet

Первым шагом для конфигурирования протокола MODBUS Ethernet в режиме клиента (client) является включение экземпляра в нужную NET (в данном случае NET 1, поскольку ЦП (центральный процессор) NX3010 имеет только один интерфейс локальной сети Ethernet). Щелкните на NET правой кнопкой мыши и выберите «Add Device...» («Добавить устройство...»), как показано на рисунке, приведенном ниже.

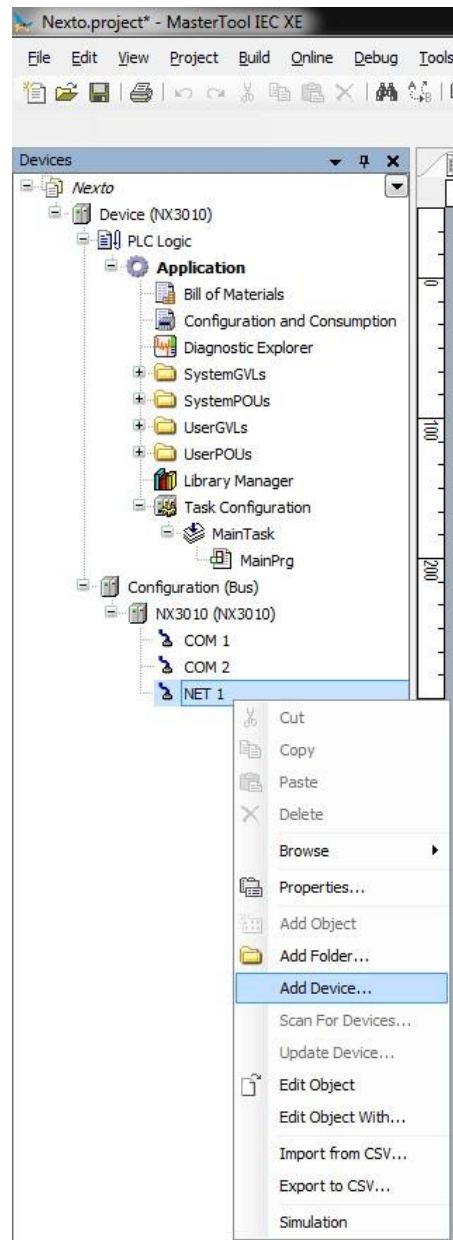


Рисунок 19: Добавление экземпляра

После этого на экране появятся доступные пользователю протоколы. Определите режим конфигурирования протокола, выбрав *MODBUS Symbol Client* («Клиент MODBUS Symbol») для настройки символьного преобразования либо или *MODBUS Client* («Клиент MODBUS») для прямой адресации (%Q), и нажмите кнопку «Add Device» («Добавить устройство»), как показано на рисунке, приведенном ниже.

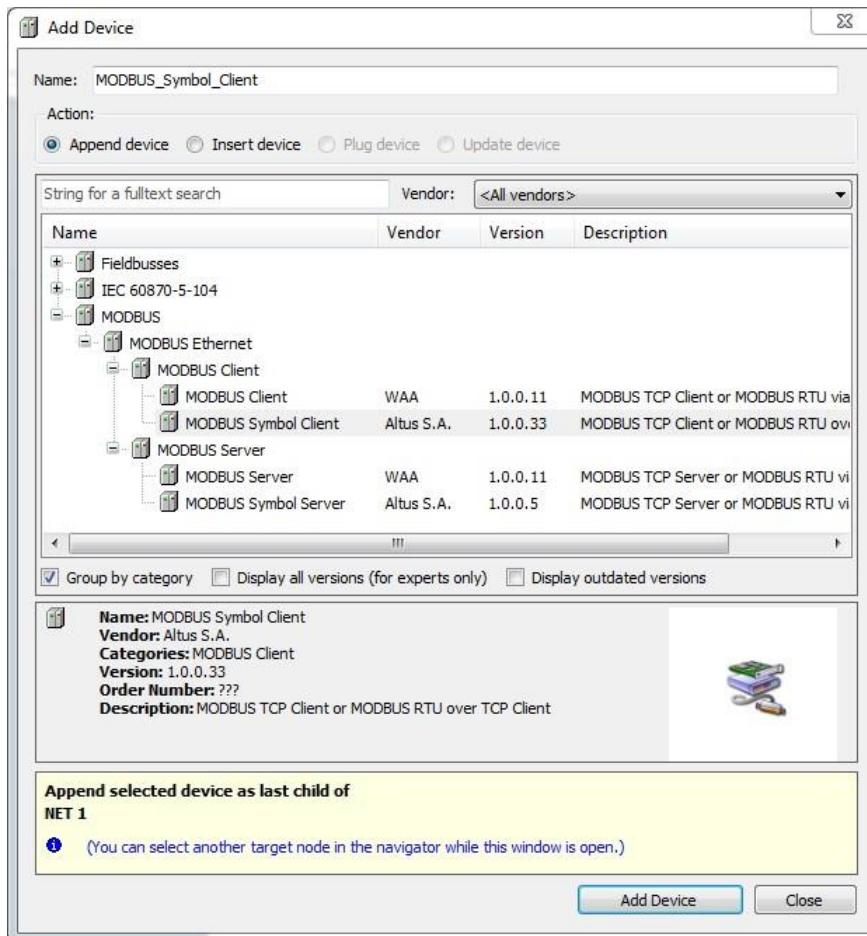


Рисунок 20: Выбор протокола

## 4.6. Поиск устройства

Для установления связи между центральным процессором (ЦП) и MasterTool IEC XE сначала необходимо найти и выбрать нужное устройство. Конфигурация данной связи находится на объекте «Устройство» (*Device*) в дереве устройств, на вкладке «Настройки связи» (*Communication Settings*). На данной вкладке, после выбора шлюза (*Gateway*) и нажатия на кнопку «Сканировать сеть» (*Scan network*), программа MasterTool IEC XE выполняет поиск устройств и показывает центральные процессоры (ЦП), найденные в сети интерфейса Ethernet станции, на которой запущен инструмент.

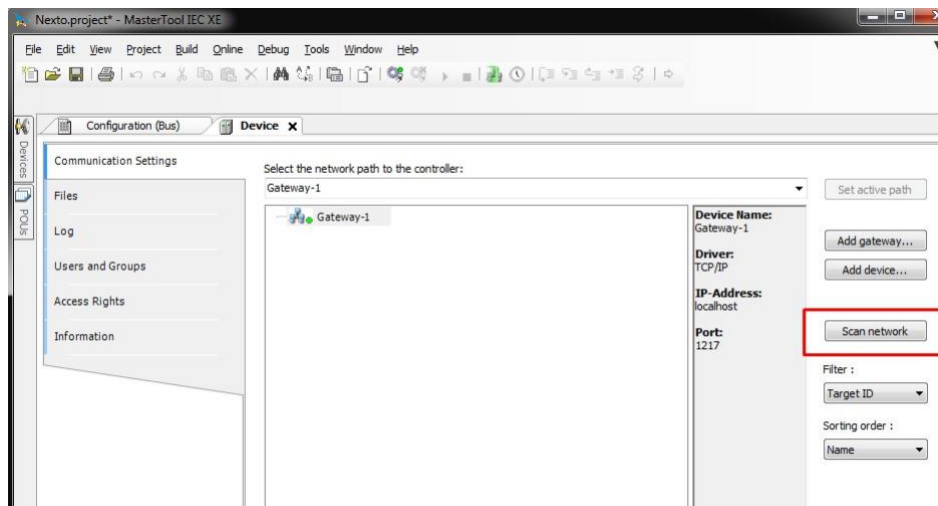


Рисунок 21: Поиск центрального процессора (ЦП)

Если шлюз не был сконфигурирован ранее, его можно включить с помощью кнопки «Добавить шлюз» (*Add gateway*), используя IP-адрес по умолчанию *localhost* для использования шлюза, установленного на станции, или изменив IP-адрес для использования внутреннего шлюза устройства.

Далее необходимо выбрать нужный контроллер, нажав на кнопку «Установить активный путь» (*Set active path*). Это действие позволяет выбрать контроллер и сообщить программному обеспечению конфигурирования, какой контроллер будет использоваться для связи и отправки проекта.

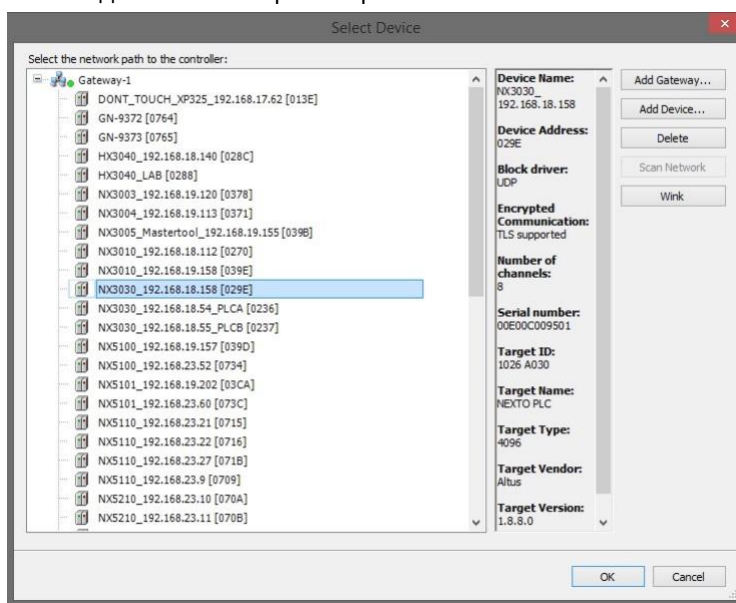


Рисунок 22: Выбор центрального процессора (ЦП)

В дополнение к этому, пользователь может изменить имя устройства, отображаемого по умолчанию. Для этой цели необходимо щелкнуть правой кнопкой мыши на нужном устройстве и выбрать пункт «Изменить имя устройства» (*Change Device Name*). После изменения имени для устройства не сможет быть возвращено имя по умолчанию ни при каких обстоятельствах.

В случае если конфигурация локальной сети Ethernet подключаемого центрального процессора (ЦП) находится в сети, отличной от интерфейса локальной сети Ethernet станции, программа MasterTool IEC XE не сможет найти устройство. В этом случае рекомендуется использовать команду «Простое подключение» (*Easy Connection*), расположенную в меню «Диалоговый режим (Online)».

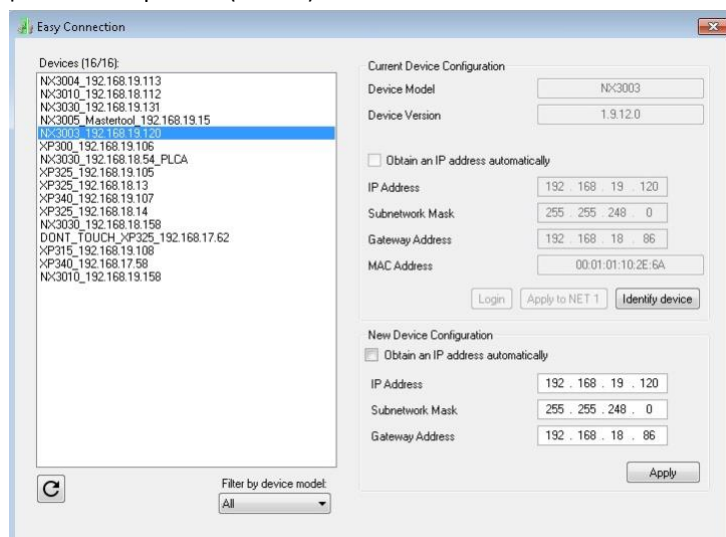


Рисунок 23: Простое подключение

Данная команда осуществляет обмен данными с устройством на уровне MAC, позволяя постоянно изменять конфигурацию интерфейса локальной сети Ethernet центрального процессора (ЦП) независимо от IP-конфигурации станции и от той, которая была ранее сконфигурирована на устройстве. Таким образом, с помощью данной команды можно изменить конфигурацию устройства, поместив его в ту же сеть, что и интерфейс локальной сети Ethernet станции, на которой запущен MasterTool IEC XE, что позволяет найти и выбрать устройство для связи. Полное описание команды «Простое подключение» (*Easy Connection*) приведено в руководстве пользователя MasterTool IEC XE код MU299609.

#### 4.7. Вход в систему

После компиляции приложения и исправления ошибок, которые могут быть обнаружены, настало время отправить проект в центральный процессор (ЦП). Для этого достаточно выбрать команду «Вход в систему» (*Login*), расположенную в меню «Диалоговый режим (*Online*)» программы MasterTool IEC XE, как показано на следующем рисунке. Данная операция может занять несколько секунд, в зависимости от размера сгенерированного файла.

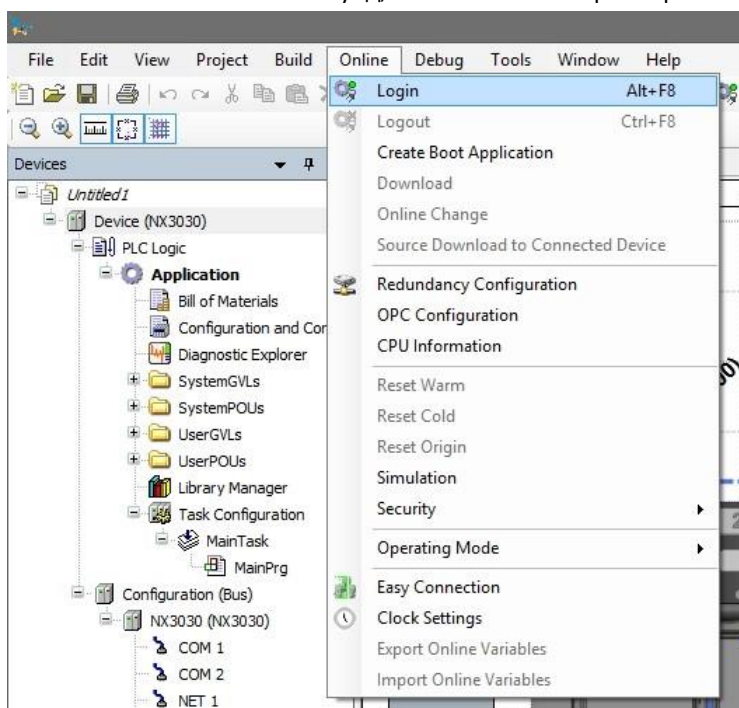


Рисунок 24: Отправка проекта в центральный процессор (ЦП)

После выполнения команды могут появиться некоторые сообщения пользовательского интерфейса, которые представлены по причине различий между старым проектом и отправленным новым проектом, либо просто из-за изменения какой-либо переменной.

Если конфигурация локальной сети Ethernet проекта отличается от конфигурации устройства, связь может быть прервана по окончании процесса загрузки, когда новая конфигурация будет применена на устройстве. В этом случае будет выдано следующее предупреждение, предлагающее пользователю продолжить или отказаться от выполнения данной операции.

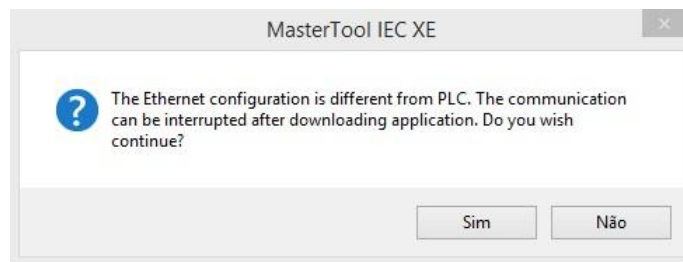


Рисунок 25: Предупреждение о конфигурации IP-адреса

Если на центральном процессоре (ЦП) не присутствует приложение, то будет выдано следующее сообщение.

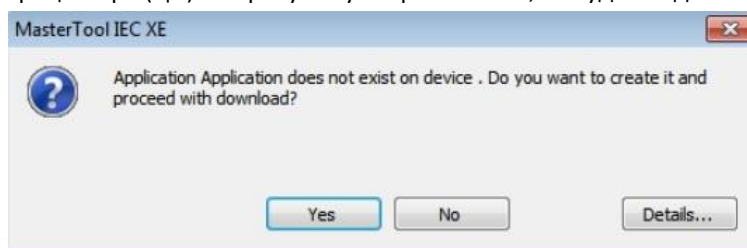


Рисунок 26: Отсутствие приложения на устройстве

В случае, если на центральном процессоре (ЦП) уже присутствует приложение, то в зависимости от различий между проектами будут представлены следующие варианты:

- Вход в систему с внесением изменений в диалоговом режиме («онлайн»): выполнение входа в систему и отправка нового проекта без остановки текущего приложения центрального процессора (ЦП) (см. пункт «Режим выполнения (Run)»), обновление изменений при выполнении нового цикла.
- Вход в систему с выполнением загрузки: выполнение входа в систему и отправка нового проекта с остановленным центральным процессором (ЦП) (см. раздел «Режим останова (Stop)»). Когда приложение будет инициировано, обновление уже будет выполнено.
- Вход в систему без каких-либо изменений: выполняет вход в систему без отправки нового проекта

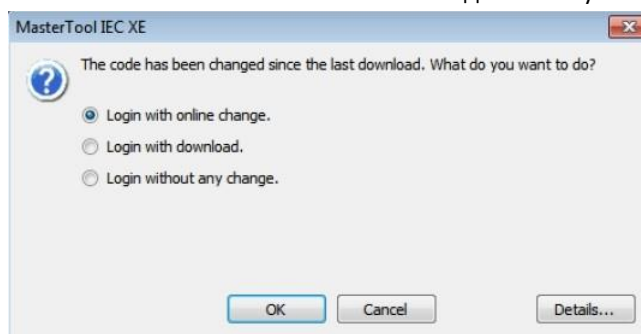


Рисунок 27: Обновление проекта на ЦП

**ВНИМАНИЕ:**

В диалоговом режиме внесения изменений («онлайн») не разрешается связывать преобразование символических переменных из списка глобальных переменных (GVL) и использовать данные переменные в другом списке глобальных переменных (GVL).

В случае если новое приложение содержит изменения в конфигурации, то внесение изменений в диалоговом режиме («онлайн») будет невозможно. В этом случае MasterTool IEC XE запрашивает, следует ли выполнить вход в систему как загрузку (выполнив останов приложения) либо отменить выполнение операции.

---

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Примечание: Кнопка «Подробнее...» (*Details...*) показывает изменения, внесенные в приложение.

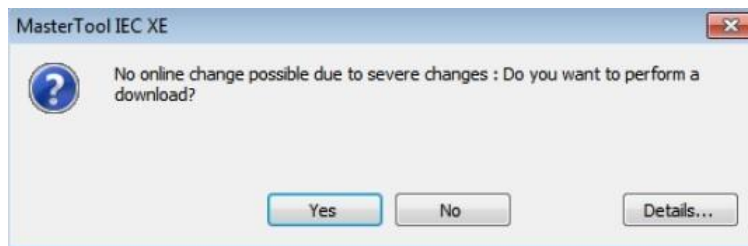


Рисунок 28: Изменения конфигурации

Наконец, по окончании данного процесса MasterTool IEC XE предлагает возможность переноса (загрузки) исходного кода во внутреннюю память устройства, как показано на следующем рисунке:

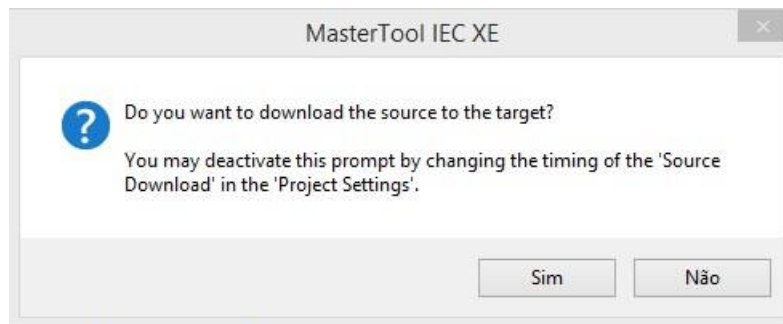


Рисунок 29: Загрузка исходного кода

Передача исходного кода имеет принципиальное значение для обеспечения восстановления проекта в будущем и выполнения модификаций приложения, загруженного в устройство.

### 4.8. Режим выполнения (Run)

Сразу же после отправки проекта в центральный процессор (ЦП) приложение не будет немедленно выполнено (за исключением случая внесения изменений в диалоговом режиме («онлайн»)). Для этого должна быть выполнена команда «Запуск» (*Start*). Таким образом, пользователь может контролировать выполнение приложения, отправленного в центральный процессор (ЦП), позволяя предварительно сконфигурировать начальные значения, которые будут использоваться центральным процессором (ЦП) при первом цикле выполнения.

Для выполнения данной команды достаточно зайти в меню «Отладка» (*Debug*) и выбрать опцию «Запуск» (*Start*), как показано на рисунке, приведенном ниже.



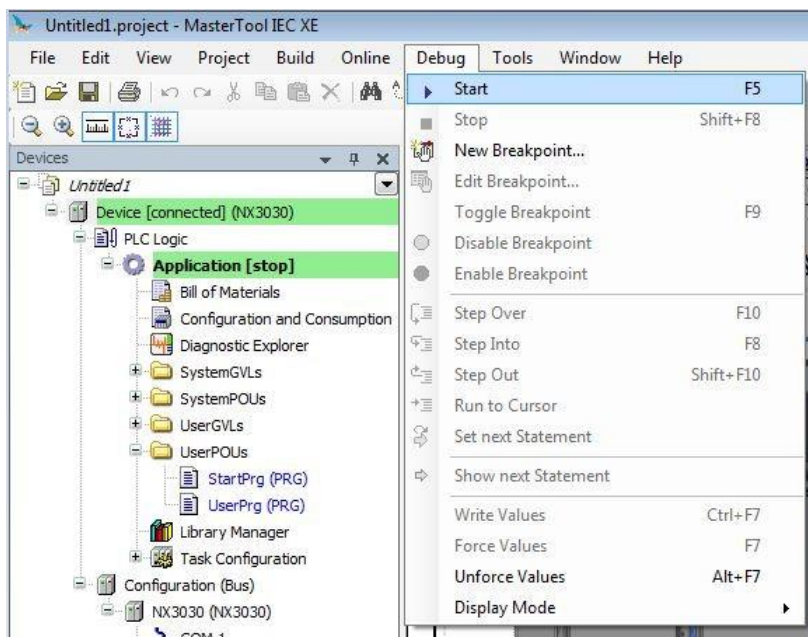


Рисунок 30: Запуск приложения

На приведенном ниже рисунке показано приложение в процессе выполнения. При выборе вкладки «Структурная единица программы (POU)» созданные переменные отображаются в окне мониторинга, в котором их значения могут быть визуализированы и принудительно изменены пользователем.

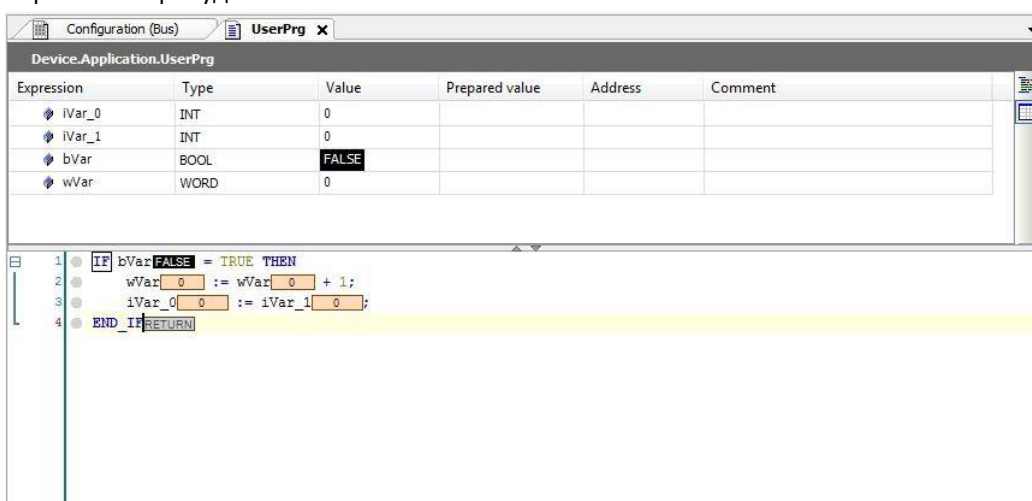


Рисунок 31: Выполнение программы

В случае если центральный процессор (ЦП) уже имеет встроенное приложение загрузки, то при включении питания он автоматически переходит в режим выполнения (Run), без необходимости подачи команды в диалоговом режиме («онлайн») через MasterTool IEC XE.

#### 4.9. Режим останова (Stop)

Чтобы остановить выполнение приложения, пользователь должен выполнить команду «Останов» (Stop), доступную в меню «Отладка» (Debug), как показано на рисунке, приведенном ниже.

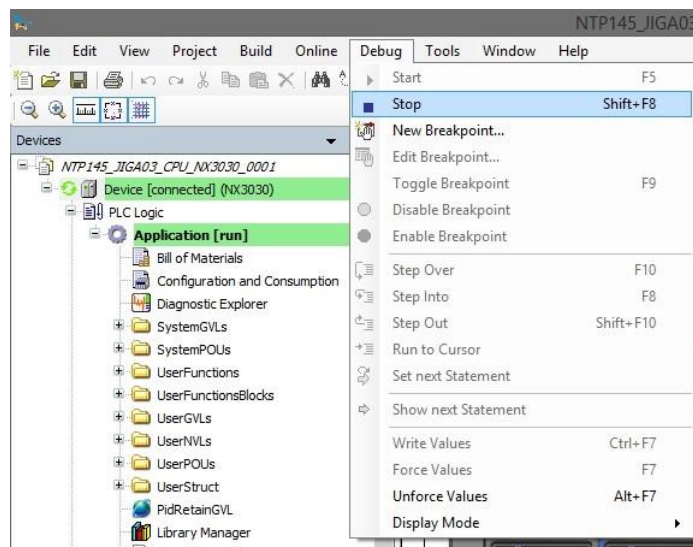


Рисунок 32: Останов приложения

В случае инициализации центрального процессора (ЦП) без сохраненного приложения он автоматически переходит в режим останова (Stop), как это происходит при возникновении исключения программного уровня.

#### 4.10. Запись и форсирование переменных

После выполнения входа в ПЛК пользователь может записать или принудительно задать значения переменной проекта.

Команда записи (**CTRL + F7**) осуществляет запись значения в переменную, и данное значение может быть перезаписано посредством команд, выполняемых в приложении.

Боле того, команда принудительной записи (**F7**) записывает значение в переменную, не позволяя изменить данное значение до тех пор, пока форсированные (принудительно заданные) переменные не будут разблокированы.

Важно отметить, что при использовании ведомого устройства MODBUS RTU (MODBUS RTU Slave) и сервера MODBUS Ethernet (MODBUS Ethernet Server), когда не выбрана опция «Только чтение» (*Read-only*) из сконфигурированных отношений, команда принудительной записи (**F7**) должна быть выполнена в отношении доступных переменных в окне мониторинга, поскольку команда записи (**CTRL + F7**) оставляет переменные перезаписанными при новых считываниях.

##### ВНИМАНИЕ:

Форсирование переменных (принудительное назначение значений переменным) может быть выполнено в диалоговом режиме («онлайн»).

Диагностические переменные не могут быть принудительно заданы, а только записаны, поскольку диагностика осуществляется центральным процессором (ЦП) и будет перезаписана им.

##### ВНИМАНИЕ:

В случае если центральный процессор (ЦП) имеет форсированные (принудительно заданные) переменные и обесточен, то при следующей инициализации переменные утратят свои принудительно заданные значения.

Предельное значение форсированных (принудительно заданных) переменных для центральных процессоров Nexto составляет 128 переменных, независимо от модели или конфигурации используемого центрального процессора.

### 4.11. Выход из системы

Для завершения связи с центральным процессором (ЦП) в диалоговом режиме («онлайн») необходимо выполнить команду «Выход из системы» (*Logout*), расположенную в меню «Диалоговый режим («онлайн»)», как показано на рисунке, приведенном ниже.

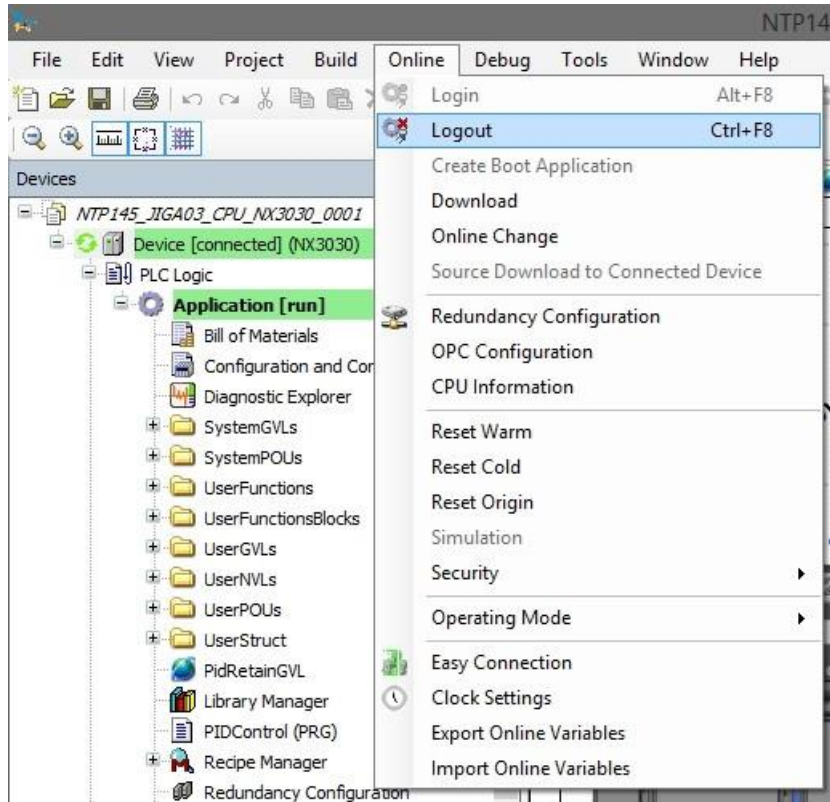


Рисунок 33: Завершение связи с центральным процессором (ЦП) в диалоговом режиме («онлайн»)

### 4.12. Загрузка проекта

Центральные процессоры (ЦП) серии Nexto способны хранить исходный код приложения во внутренней памяти устройства, что позволяет в дальнейшем извлекать (загружать) полный проект и вносить изменения в приложение.

Для восстановления проекта, ранее сохраненного во внутренней памяти центрального процессора (ЦП), необходимо выполнить команду, расположенную в меню «Файл» (*File*), как показано на следующем рисунке.

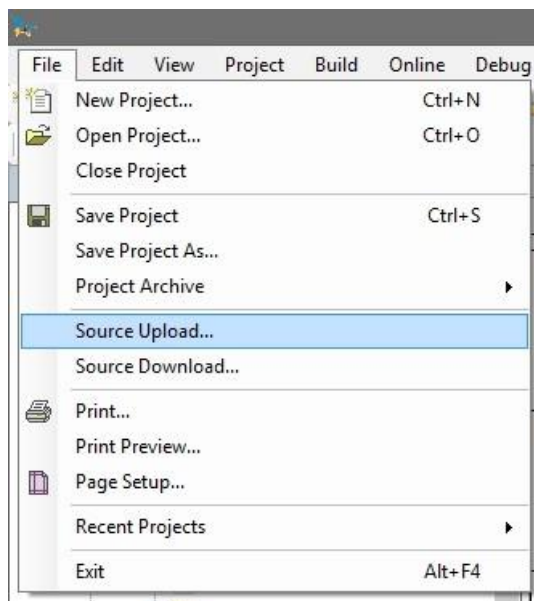


Рисунок 34: Опция загрузки проекта

Далее просто выберите нужный центральный процессор (ЦП) и нажмите кнопку «OK», как показано на рисунке, приведенном ниже.

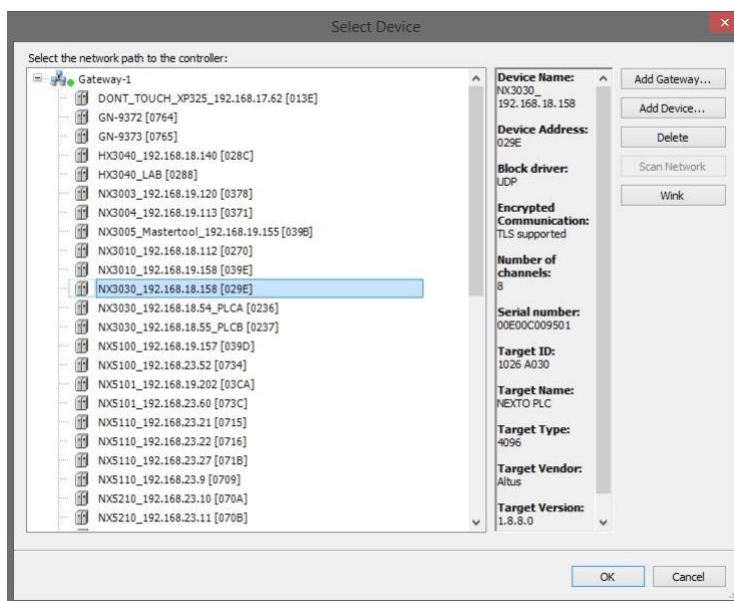


Рисунок 35: Выбор центрального процессора (ЦП)

Чтобы убедиться в том, что проект, загруженный в центральный процессор (ЦП), идентичен и может быть доступен на других рабочих станциях, обратитесь к главе «Способ загрузки/входа в систему без различий в проектах» в руководстве пользователя MasterTool IEC XE MT8500 – MU299609.

**ВНИМАНИЕ:**  
Область емкости памяти для хранения проекта в центральных процессорах (ЦП) Nexto определяется в разделе «Память»

**ВНИМАНИЕ:**

При загрузке восстанавливается последний проект, сохраненный в контроллере, как описано в предыдущих пунктах. Если было загружено только приложение, без передачи его исходного кода, то восстановить его с помощью процедуры загрузки будет невозможно.

### 4.13. Рабочие состояния центрального процессора (ЦП)

#### 4.13.1. Выполнение (Run)

Когда центральный процессор (ЦП) находится в режиме «Выполнение» (*Run*), выполняются все прикладные задачи.

#### 4.13.2. Останов (Stop)

Когда центральный процессор (ЦП) находится в режиме «Останов» (*Stop*), выполнение всех прикладных задач останавливается. Значения переменных в задачах сохраняются с текущим значением, а точки выхода переходят в безопасное состояние.

Когда центральный процессор (ЦП) переходит в режим «Останов» (*Stop*) по причине загрузки приложения, переменные в прикладных задачах будут потеряны, за исключением переменных, относящихся к постоянному типу.

#### 4.13.3. Точка контрольного прерывания (Breakpoint)

При достижении метки отладки в задаче, ее выполнение прерывается. При этом все активные задачи в приложении не прерываются, продолжая свое выполнение. Благодаря данной возможности в *диалоговом* режиме («онлайн») можно пошагово пройти и исследовать поток программы в соответствии с позициями прерываний, включенных через редактор.

Более подробную информацию об использовании точек контрольного прерывания можно найти в руководстве по эксплуатации MasterTool IEC XE – MU299609.

#### 4.13.4. Исключение (Exception)

Когда центральный процессор (ЦП) находится в состоянии «Исключение» (*Exception*), это означает, что в одной из активных задач приложения произошла некорректная операция. Выполнение задачи, вызвавшей состояние «Исключение» (*Exception*), будет приостановлено, а остальные задачи перейдут в режим «Останов» (*Stop*). Вывести задачи из данного состояния и вновь запустить их выполнение можно только после нового запуска центрального процессора (ЦП). Таким образом, только *теплый сброс* (*Reset Warm*), *холодный сброс* (*Reset Cold*), *сброс до исходного состояния* (*Reset Origin*) либо перезапуск центрального процессора (ЦП) снова переведут приложение в режим «Выполнение» (*Run*).

#### 4.13.5. Теплый сброс (Reset Warm)

Данная команда переводит центральный процессор (ЦП) в режим «Останов» (*Stop*) и инициализирует все переменные прикладных задач, за исключением переменных постоянного и сохраняемого типов. Переменные, инициализированные с определенным значением, примут именно это значение, а остальные переменные примут стандартное значение инициализации (ноль).

#### 4.13.6. Холодный сброс (Reset Cold)

Эта команда переводит центральный процессор (ЦП) в режим «Останов» (*Stop*) и инициализирует все переменные прикладных задач, за исключением переменных постоянного типа. Переменные, инициализированные с определенным значением, примут именно это значение, а остальные переменные примут стандартное значение инициализации (ноль).

#### 4.13.7. Сброс до исходного состояния (Reset Origin)

Данная команда удаляет все переменные задачи приложения, включая переменные постоянного типа, удаляет центральный процессор (ЦП) приложения и переводит центральный процессор в режим «Останов» (*Stop*).

Примечания:

---

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Сброс: При выполнении сброса точки контрольного прерывания, определенные в приложении, отключаются.

Команда: Для выполнения команд *теплый сброс*, *холодный сброс* или *сброс до исходного состояния* необходимо, чтобы MasterTool IEC XE находился в *диалоговом* режиме («онлайн») с центральным процессором (ЦП).

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

Проект, созданный MasterTool IEC XE, содержит набор программных модулей (структурные единицы программы (POU)) и списки глобальных переменных, цель которых заключается в том, чтобы облегчить программирование и использование контроллера. В следующих разделах описаны основные элементы, входящие в данную стандартную структуру проекта.

#### 4.14.1. Программа MainPrg

Задача MainTask связана с одной уникальной структурной единицей программы (POU) программного типа под названием MainPrg. Программа MainPrg создается автоматически и не может быть отредактирована пользователем.

Код программы MainPrg на языке структурированного текста (ST) выглядит следующим образом:

```
(*Основная структурная единица программы (POU), связанная с задачей MainTask, которая вызывает StartPrg, UserPrg/ActivePrg и NonSkippedPrg. Данная структурная единица программы (POU) заблокирована для редактирования.*)

ПРОГРАММА MainPrg
VAR isFirstCycle: BOOL := TRUE («ИСТИНА»);
END_VAR

SpecialVariablesPrg();
ЕСЛИ isFirstCycle ТОГДА StartPrg();
    isFirstCycle := FALSE («ЛОЖЬ»);
ELSE
    UserPrg();
END_IF;
```

MainPrg вызывает две другие структурные единицы программы (POU) программного типа, именуемые *StartPrg* и *UserPrg*. В то время как вызов *UserPrg* осуществляется всегда, вызов *StartPrg* происходит только один раз при запуске приложения ПЛК.

В противовес программе *MainPrg*, которая не должна подвергаться внесению изменений, пользователь может изменить программы *StartPrg* и *UserPrg*. Первоначально, при создании проекта с помощью мастера, эти две программы создаются пустыми, но пользователь может вставить в них код.

#### 4.14.2. Программа StartPrg

В данной структурной единице программы (POU) пользователь может создавать логики, циклы, стартовые переменные и т.д., которые будут выполняться только один раз в первом цикле ПЛК, перед первым выполнением структурной единицы программы (POU) *UserPrg* и их вызов больше не будут осуществляться во время выполнения проекта.

В случае загрузки пользователем нового приложения либо при выключении питания ПЛК, а также в условиях сброса до исходного состояния (Reset Origin), холодного сброса (Reset Cold) и теплого сброса (Reset Warm) данная структурная единица программы (POU) будет выполнена повторно.

##### 4.14.3. Программа UserPrg

В данной структурной единице программы (POU) пользователь должен создать главное приложение, отвечающее за управление собственным процессом. Вызов данной структурной единицы программы (POU) осуществляется главной структурной единицей программы (POU) (MainPrg).

Пользователь также может создавать дополнительные структурные единицы программы (POU) (программы, функции или функциональные блоки), а также вызывать их или инстанцировать внутри структурной единицы программы (POU) UserPrg, в конце своей программной команды. Также можно вызывать функции и инстанцировать функциональные блоки, определенные в библиотеках.

##### 4.14.4. Список глобальных переменных (GVL) System\_Diagnostics

Список глобальных переменных (GVL) *System\_Diagnostics* содержит диагностические переменные центрального процессора, коммуникационного интерфейса (Ethernet и PROFIBUS) и всех коммуникационных драйверов. Данный список глобальных переменных (GVL) не редактируется, а при добавлении в проект переменные объявляются автоматически с типом, определяемым устройством, к которому оно относится.

**ВНИМАНИЕ:**

В списке глобальных переменных (GVL) *System\_Diagnostics* также объявлены диагностические переменные прямого представления запросов клиента/ведущего устройства MODBUS (MODBUS Client/Master).

В некоторых устройствах, например в коммуникационном драйвере MODBUS Symbol, диагностика не выделяется в переменных %Q с помощью директивы AT. То же самое происходит и с новейшими коммуникационными драйверами, такими, как сервер МЭК 60870-5-104 (Server IEC 60870-5-104).

На следующем рисунке показан пример представления данного списка глобальных переменных (GVL) в диалоговом режиме («онлайн»).

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

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 36: Список глобальных переменных (GVL) System\_Diagnostics в диалоговом режиме («онлайн»)

#### 4.14.5. Список глобальных переменных (GVL) «Отключения» (Disables)

В списке глобальных переменных (GVL) «Отключения» (Disables) содержатся переменные отключения ведущего устройства / клиента MODBUS (MODBUS Master/Client) по символному преобразованию запроса. Не обязательно, но рекомендуется использовать автоматическую генерацию данных переменных, которая осуществляется нажатием кнопки «Генерирование переменных отключения» (Generate Disabling Variables) на вкладке «запрос устройства» (device requisition). Данные переменные объявляются как относящиеся к питу BOOL и имеют следующую структуру:

Объявление переменных отключения запроса:

```
[Имя устройства]_DISABLE_[Номер запроса] : BOOL;
```

Где:

Имя устройства: имя, которое отображается в режиме просмотра дерева для устройства MODBUS.

Номер запроса: Номер запроса, который был объявлен в таблице запросов устройства MODBUS в последовательности сверху вниз, начиная с 0001.



#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Пример:  
Device.Application.Disables

```
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;
END_VAR
```

Автоматическая генерация с помощью кнопки «Генерирование переменных отключения» (*Generate Disabling Variables*) только создает переменные, но не удаляет их автоматически. Таким образом, в случае удаления какого-либо отношения соответствующая ей отключающая переменная должна быть удалена вручную.

Список глобальных переменных (GVL) «Отключения» (*Disables*) является редактируемым, поэтому переменные отключения заявок могут быть созданы вручную без необходимости следовать модели, созданной автоматическим объявлением, и могут использоваться одновременно в двух вариантах, но всегда должны относиться к типу BOOL. При этом необходимо следить за тем, чтобы не удалять и не изменять автоматически объявленные переменные, поскольку они могут быть использованы для какого-либо устройства MODBUS. Если переменная будет удалена или изменена, то при компиляции проекта будет выдана ошибка. Для корректировки автоматически объявленного имени переменной необходимо следовать приведенной выше модели в соответствии с устройством и запросом, к которому оно относится.

На следующем рисунке показан пример представления данного списка глобальных переменных (GVL) в диалоговом режиме («онлайн»). Если значения переменных равны TRUE («ИСТИНА»), это означает, что запрос, к которому относятся переменные, отключен, и наоборот, если значение переменной равно FALSE («ЛОЖЬ»).









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 37: Список глобальных переменных (GVL) «Отключение» (Disable) в диалоговом режиме («онлайн»)

#### 4.14.6. Список глобальных переменных (GVL) IOQualities

Список глобальных переменных (GVL) *IOQualities* содержит переменные качества модулей входа/выхода, объявленных на шине центрального процессора (ЦП). Данный список глобальных переменных (GVL) не редактируется, а переменные автоматически объявляются как массивы типа *LibDataTypes.QUALITY* и при добавлении в проект имеют размеры, соответствующие количеству входов/выходов модуля, к которому они относятся.

Пример: Device.Application.IOQualities

```
VAR_GLOBAL
QUALITY_NX1001: МАССИВ[0..15] LibDataTypes.QUALITY;
QUALITY_NX2020: МАССИВ [0..15] LibDataTypes.QUALITY;
    QUALITY_NX6000: МАССИВ [0..7]           LibDataTypes.QUALITY;
    QUALITY_NX6100: МАССИВ [0..3]           LibDataTypes.QUALITY;
END_VAR
```

Когда приложение находится в режиме «Выполнение» (*RUN*), можно просмотреть значения переменных качества модулей входа/выхода, которые были добавлены в проект через список глобальных переменных (GVL) *IOQualities*.

#### 4.14.7. Список глобальных переменных (GVL) Module\_Diagnostics

Список глобальных переменных (GVL) *Module\_Diagnostics* содержит диагностические переменные модулей входа/выхода, используемых в проекте, за исключением центрального процессора и коммуникационных драйверов. Данный список глобальных переменных (GVL) не редактируется, и при добавлении в проект переменные автоматически объявляются с типом, заданным модулем, к которому они относятся.

На следующем рисунке показан пример представления данного списка глобальных переменных (GVL) в диалоговом режиме («онлайн»).

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 38: Список глобальных переменных (GVL) Module\_Diagnostics в диалоговом режиме («онлайн»)

#### 4.14.8. Список глобальных переменных (GVL) ReqDiagnostics

В списке глобальных переменных (GVL) *ReqDiagnostics* содержатся диагностические переменные запроса, имеющие символьное преобразование ведущего устройства/клиента MODBUS (MODBUS Master/Client). Не обязательно, но рекомендуется использовать автоматическую генерацию данных переменных, которая осуществляется нажатием кнопки «Генерирование диагностических переменных» (*Generate Diagnostics Variables*) на вкладке «Запросы устройства» (Device Request). Объявление данных переменных имеет следующую структуру:

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Объявление диагностических переменных запроса:

```
[Имя устройства]_REQDG_[ Номер запроса]: [Тип переменной];
```

Где:

Имя устройства: Имя, отображаемое в древовидном представлении для устройства.

Номер преобразования: Номер преобразования, объявленного в таблице преобразования устройств, в последовательности «вверх-вниз», начиная с 0001.

Тип переменной: NXMODBUS\_DIAGNOSTIC\_STRUCTS.T\_DIAG\_MODBUS\_RTU\_MAPPING\_1 к ведущему устройству MODBUS (MODBUS Master) и NXMODBUS\_DIAGNOSTIC\_STRUCTS.T\_DIAG\_MODBUS\_ETH\_MAPPING\_1 к клиенту MODBUS (MODBUS Client).

**ВНИМАНИЕ:**

Переменные диагностики запроса при прямом преобразовании Ведущего устройства/Клиента MODBUS (MODBUS Master/Client) объявляются в списке глобальных переменных (GVL) *System\_Diagnostics*.

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;
  MODBUS_Device_1_REQDG_0002 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_ETH_MAPPING_1;
END_VAR
```

Список глобальных переменных (GVL) *ReqDiagnostics* является редактируемым, в связи с чем диагностические переменные запросов 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. При этом необходимо следить за тем, чтобы не удалять и не изменять автоматически объявленные переменные, поскольку они могут быть использованы каким-либо прибором. Если переменная будет удалена или изменена, то при компиляции проекта будет выдана ошибка. Для корректировки автоматически объявленного имени переменной необходимо следовать приведенной выше модели в соответствии с устройством и запросом, к которому оно относится.

На следующем рисунке показан пример представления данного списка глобальных переменных (GVL) в диалоговом режиме («онлайн»).

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 39: Список глобальных переменных (GVL) ReqDiagnostics в диалоговом режиме («онлайн»)

#### 4.14.9. Функция Prepare\_Start

В данной структурной единице программы (POU) определена функция системного события *PrepareStart*. Она относится к коммуникационной задаче и ее вызов осуществляется перед запуском приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке «Системные события» (*System Events*) объекта «Конфигурация задач» (*Task Configuration*). Каждый раз, когда пользователь запускает приложение, значение счетчика увеличивается.

#### 4.14.10. Функция Prepare\_Stop

В данной структурной единице программы (POU) определена функция системного события *PrepareStop*. Она относится к коммуникационной задаче и ее вызов осуществляется перед выполнением останова приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке «Системные события» (*System Events*) объекта «Конфигурация задач» (*Task Configuration*). Каждый раз, когда пользователь производит останов приложения, значение счетчика увеличивается.

#### 4.14.11. Функция Start\_Done

В данной структурной единице программы (POU) определена функция системного события *StartDone*. Она относится к коммуникационной задаче и ее вызов осуществляется при успешном запуске приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке «Системные события» (*System Events*) объекта «Конфигурация задач» (*Task Configuration*). Каждый раз, когда пользователь успешно запускает приложение, значение счетчика увеличивается.

#### 4.14.12. Функция Stop\_Done

В данной структурной единице программы (POU) определена функция системного события *StopDone*. Она относится к коммуникационной задаче и ее вызов осуществляется при успешном завершении работы приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке «Системные события» (*System Events*) объекта «Конфигурация задач» (*Task Configuration*). Каждый раз, когда пользователь успешно выполняет останов приложения, значение счетчика увеличивается.

## 5. Конфигурация

Конфигурирование и программирование центральных процессоров (ЦП) серии Nexto осуществляется посредством программного обеспечения MasterTool IEC XE. Конфигурирование определяет логику работы и режимы эксплуатации периферийных устройств, а также специальные возможности центрального процессора (ЦП). Программирование представляет собой приложение, разработанное пользователем.

### 5.1. Конфигурация центрального процессора (ЦП)

#### 5.1.1. Общие параметры

Приведенные ниже параметры являются частью конфигурации центрального процессора (ЦП), входящей в состав приложения. Для корректного выполнения проекта каждый из них должен быть проверен надлежащим образом.

Помимо данных параметров, можно изменить имя каждого модуля, вставленного в приложение, нажав правую кнопку на модуле. В пункте «Свойства» (*Properties*) на листе «Общее» (*Common*) измените имя, которое ограничено 24 символами.

Настройки	Описание	Стандарт	Опции
	Область диагностики (%Q)		
Начальный адрес %Q	Начальный адрес диагностики центрального процессора (ЦП) (%Q)	Автоматически выделяемый проект при создании.	от 0 до 32210
Размер	Размер области диагностики в байтах	558	Невозможно изменить размер области диагностики центрального процессора (ЦП)
	Область сохранения (%Q)		
Начальный адрес %Q	Начальный адрес сохраняющей памяти данных (%Q)	4096	от 0 до 32767
Размер	Размер памяти для хранения данных в байтах	7680	от 0 до 7680
	Область постоянной памяти (%Q)		
Начальный адрес %Q	Начальный адрес постоянной памяти данных (%Q)	12288	от 0 до 32767
Размер	Объем постоянной памяти данных в байтах	7680	от 0 до 7680
	Параметры центрального процессора (ЦП)		
Запуск пользовательского приложения после сброса по сторожевому таймеру	Если данная функция включена, то после сброса аппаратного сторожевого таймера или перезапуска среды для выполнения кода (Runtime) запускается пользовательское приложение, но при этом сохраняется диагностическая	Выключено	Включено Выключено

5. КОНФИГУРАЦИЯ

Настройки	Описание	Стандарт	Опции
	индикация через светодиод WD и через переменные.		
Режим «горячей» замены модуля	Режим «горячей» замены модуля	Включено, без согласованности совместимости. (может меняться в зависимости от модели центрального процессора)	<ul style="list-style-type: none"> <li>- Выключено, только для объявленных модулей</li> <li>- Выключено (с согласованностью совместимости)</li> <li>- Выключено, без согласованности совместимости</li> <li>- Включено, с согласованностью совместимости только для объявленных модулей</li> <li>- Включено, с согласованностью совместимости</li> <li>- Включено, без согласованности совместимости</li> </ul>
	Параметры TCP/IP		
Начальная блокировка по превышению лимита времени (x100 мс)	Указывает, через какое время после первой передачи сообщения оно должно быть повторно передано, если предположить, что оно не было получено получателем. При каждой повторной передаче значение блокировки по превышению лимита времени удваивается.	2	от 1 до 75
Задержка отправки подтверждения приема сигнала (АСК) (x10 мс)	Блокировка по превышению лимита времени для отправки команды подтверждения.	10	от 0 до 100
	Параметры проекта		
Соответствующая область сохраняющей и постоянной памяти в %Q	Конфигурация состоит из адресуемой постоянной и сохраняющей памяти.	Не отмечено	<ul style="list-style-type: none"> <li>- Отмечено: Состоит из адресуемой постоянной и сохраняющей памяти</li> <li>- Не отмечено: Не состоит из адресуемой постоянной и сохраняющей памяти.</li> </ul>
Включить обновление входов/выходов на одну задачу	Настройка для 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 в MainTask</li> </ul>

## 5. КОНФИГУРАЦИЯ

Включите сохраняемые и постоянные переменные в функциональном блоке	Настройка, позволяющая использовать в функциональных блоках сохраняемые и постоянные переменные	Не отмечено	<p>- Отмечено: разрешает использование сохраняемых и постоянных переменных в функциональных блоках.</p> <p>- Не отмечено: При запуске может возникнуть ошибка исключения.</p>
---	---	-------------	---

Таблица 36: Настройки центрального процессора (ЦП)

### Примечания:

Соответствующая область сохраняющей и постоянной памяти в %Q: Если в пользовательском приложении объявлены сохраняемые или постоянные переменные, преобразованные по адресам %Q за пределами сконфигурированных областей, то данные переменные не сохраняются после выключения питания, теплого сброса или загрузки. При отмеченном параметре MasterTool будет проверять код и генерировать ошибку во время компиляции для данной ситуации. Настоятельно рекомендуется держать данный параметр отмеченным.

#### ВНИМАНИЕ:

При изменении начального адреса или размера памяти сохраняемых или постоянных данных в пользовательском приложении происходит полное перераспределение памяти, что приводит к очистке области сохраняемых и постоянных переменных. Поэтому пользователь должен быть внимателен, чтобы не потерять сохраненные в памяти данные.

#### ВНИМАНИЕ:

В ситуациях, когда модифицируется символьная постоянная область памяти, устройство программирования (программатор) MasterTool IEC XE выводит сообщение о выборе логики работы для данной области после зарядки модифицированной программы. Выбор данной логики работы не оказывает влияния на постоянную область прямого представления, которая всегда чиста.

#### ВНИМАНИЕ:

Опция «Включить обновление ввода-вывода для каждой задачи» (*Enable I/O update per task*) не поддерживается для ведущих устройств промышленной шины, таких как модуль NX5001. Данная функция применима только для модулей ввода и вывода, присутствующих на локальной шине контроллера (главная стойка и стойки расширения).

#### ВНИМАНИЕ:

Даже если точка ввода/вывода используется в других задачах, при отмеченной опции «Включить обновление ввода-вывода для каждой задачи» (*Enable I/O update per task*), она будет продолжать обновляться и в MainTask. Исключение составляют случаи, когда все точки модуля используются в какой-либо другой задаче: в этом случае они больше не будут обновляться в MainTask.

### 5.1.1.1. «Горячая» замена

В центральных процессорах (ЦП) серии Nexto реализована возможность замены модулей входа/выхода в шине без необходимости выключения системы и без потери информации. Данная возможность известна как «горячая» замена.

#### ОСТОРОЖНО:

Центральные процессоры (ЦП) серии Nexto не гарантируют сохранение постоянных и сохраняемых переменных в случае, если источник питания или даже сам центральный процессор (ЦП) будут извлечены из стойки объединительной платы, подключенной к источнику питания.

При «горячей» замене соответствующая логика работы системы изменяется в соответствии с заданной пользователем таблицей конфигурации, в которой представлены следующие варианты:

## 5. КОНФИГУРАЦИЯ

- Выключено, только для объявленных модулей
- Выключено (с согласованностью при запуске)
- Выключено, без согласованности при запуске
- Включено, с согласованностью при запуске только для объявленных модулей
- Включено, с согласованностью при запуске
- Включено, без согласованности при запуске

Таким образом, пользователь может выбрать логику работы системы в нештатных ситуациях на шине и когда центральный процессор (ЦП) находится в режиме *выполнения работы (Ru)*. В таблице, приведенной ниже, представлены возможные нештатные ситуации на шине.

Ситуация	Возможные причины
Несовместимая конфигурация	- Какой-то модуль, подключенный к шине, отличается от модели, заявленной в конфигурации.
Отсутствие модуля	- Модуль был удален из шины. - Какой-то неисправный модуль не отвечает на запросы центрального процессора (ЦП) - Какая-то позиция шины неисправна.

Таблица 37: Аварийные ситуации на шине

Более подробная информация о диагностике, соответствующей описанным выше ситуациям, приведена в разделе «*Диагностика посредством переменных (Diagnostics via Variables)*».

Если модуль присутствует в определенной позиции, в которой он не должен находиться в соответствии с конфигурационными модулями, то данный модуль считается необъявленным. Опции «горячей» замены «*Выключено, только для объявленных модулей (Disabled, for Declared Modules Only)*» и «*Включено, с согласованностью запуска только для объявленных модулей (Включено, с согласованностью запуска только для объявленных модулей)*» не учитывают модули, находящиеся в таком состоянии.

### 5.1.1.1.1. «Горячая» замена выключена, только для объявленных модулей

В этой конфигурации при возникновении нештатной ситуации на шине (как описано в таблице 37) центральный процессор (ЦП) немедленно переходит в режим *останова (Stop)*. Светодиод DG начинает мигать 4 раза (в соответствии с таблицей 38). В этом случае для того, чтобы центральный процессор (ЦП) вернулся в нормальное состояние «*Выполнение (Run)*», помимо отмены того, что вызвало нештатную ситуацию, необходимо выполнить «*Теплый сброс (Reset Warm)*» или «*Холодный сброс (Reset Cold)*». Если выполняется «*Сброс до исходного состояния (Reset Origin)*», то необходимо выполнить загрузку, чтобы центральный процессор (ЦП) мог вернуться в нормальное состояние «*Выполнение (Run)*». Команды «*Теплый сброс (Reset Warm)*», «*Холодный сброс (Reset Cold)*» и «*Сброс до исходного состояния (Reset Origin)*» могут быть выполнены с помощью MasterTool IEC XE в меню «*Диалоговый режим (онлайн)*».

Центральный процессор (ЦП) останется в нормальном режиме работы даже при обнаружении модуля, не заявленного на шине.

### 5.1.1.1.2. «Горячая» замена выключена

Данная настройка не допускает каких-либо нештатных ситуаций в шине (как показано в таблице 37), включая необъявленные и присутствующие на шине модули. При этом центральный процессор (ЦП) переходит в режим «*Останов (Stop)*», а светодиод DG начинает мигать 4 раза (как показано в таблице 38). В этих случаях для возврата центрального процессора в нормальный режим «*Выполнение (Run)*», помимо устранения причины нештатной ситуации, необходимо выполнить «*Теплый сброс (Reset Warm)*» или «*Холодный сброс (Reset Cold)*». Если выполняется «*Сброс до исходного состояния (Reset Origin)*», то необходимо загрузить проект, чтобы центральный процессор (ЦП) мог вернуться в нормальный режим «*Выполнение (Run)*». Команды «*Теплый сброс (Reset Warm)*», «*Холодный сброс (Reset Cold)*» и «*Сброс до исходного состояния (Reset Origin)*» могут быть выполнены в MasterTool IEC XE в меню «*Диалоговый режим (онлайн)*».



---

## 5. КОНФИГУРАЦИЯ

### 5.1.1.1.3. «Горячая» замена выключена, без согласованности при запуске

Позволяет запустить систему, даже если какой-либо модуль находится в состоянии нештатной ситуации на шине (как показано в таблице 37). О присутствии нештатных ситуациях сообщается посредством диагностики.

Любая модификация шины приведет к тому, что центральный процессор (ЦП) перейдет в режим «Останов» (Stop), а светодиод DG начнет мигать 4 раза (как показано в таблице 38). Для возврата центрального процессора (ЦП) в нормальное состояние «Выполнение» (Run) в этих случаях необходимо выполнить «Теплый сброс» (Reset Warm) или «Холодный сброс» (Reset Cold). Если выполняется «Сброс до исходного состояния» (Reset Origin), то необходимо загрузить центральный процессор (ЦП), чтобы он мог вернуться в нормальное состояние «Выполнение» (Run). Команды «Теплый сброс» (Reset Warm), «Холодный сброс» (Reset Cold) и «Сброс до исходного состояния» (Reset Origin) могут быть выполнены с помощью MasterTool IEC XE в меню «Диалоговый режим («онлайн»)».

### 5.1.1.1.4. Горячая замена включена, с согласованностью при запуске только для объявленных модулей

«Запуск» представляет собой интервал времени между подачей питания на центральный процессор (ЦП) (или команды сброса, либо загрузки приложения) и первым переходом центрального процессора (ЦП) в режим «Выполнение» (Run) после включения. Данная конфигурация проверяет, не возникла ли во время запуска какая-либо нештатная ситуация на шине (как описано в таблице 37). В утвердительном случае центральный процессор (ЦП) переходит в режим «Останов» (Stop), а светодиод DG начинает мигать 4 раза (в соответствии с таблицей 38). После этого, чтобы перевести центральный процессор в режим «Выполнение» (Run), а затем устранить причину нештатной ситуации, необходимо выполнить команду «Теплый сброс» (Reset Warm) или «Холодный сброс» (Reset Cold), которая может быть выполнена с помощью MasterTool IEC XE (меню «Диалоговый режим («онлайн»)»). При выполнении команды «Сброс до исходного состояния» (Reset Origin) необходимо будет выполнить загрузку, чтобы центральный процессор (ЦП) мог вернуться в нормальное состояние («Выполнение» (Run)).

Если после запуска в каком-либо модуле возникнет ситуация, описанная в предыдущей таблице, система продолжит работу в нормальном режиме и направит сигнал о проблеме посредством диагностики.

Если для заявленных модулей нет никаких других отклонений, то центральный процессор (ЦП) перейдет в нормальное состояние («Выполнение» (Run)), даже если на шине присутствует не заявленный модуль.

#### ВНИМАНИЕ:

В данной конфигурации при возникновении (даже временного) сбоя питания, выполнении команды «Теплый сброс» (Reset Warm), «Холодный сброс» (Reset Cold) или «Загрузка» (Download) нового приложения, а также при возникновении нештатной ситуации на шине любого модуля, центральный процессор (ЦП) переходит в режим «Останов» (Stop), а светодиод DG начинает мигать 4x (в соответствии с таблицей 38). Подобная ситуация считается ситуацией запуска.

Данный вариант является наиболее предпочтительным, поскольку гарантирует целостность системы при ее инициализации и позволяет производить замену модулей при работающей системе.

### 5.1.1.1.5. «Горячая» замена включена с согласованностью при запуске

Данная настройка позволяет осуществить проверку того, не возникла ли во время запуска какая-либо нештатная ситуация в шине (как показано в таблице 37), даже если на шине нет заявленных и присутствующих модулей; если да, то центральный процессор (ЦП) переходит в режим «Останов» (Stop), а светодиод DG начинает мигать 4x (как показано в таблице 38). В этих случаях для возврата центрального процессора (ЦП) в нормальный режим работы, помимо устранения причины нештатной ситуации, необходимо выполнить «Теплый сброс» (Reset Warm) или «Холодный сброс» (Reset Cold). Если выполняется «Сброс до исходного состояния» (Reset Origin), то необходимо загрузить проект, чтобы центральный процессор (ЦП) мог вернуться в нормальный режим «Выполнение» (Run). Команды «Теплый сброс» (Reset Warm), «Холодный сброс» (Reset Cold) и «Сброс до исходного состояния» (Reset Origin) могут быть выполнены с помощью MasterTool IEC XE в меню «Диалоговый режим («онлайн»)».

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

Позволяет начать работу системы, даже если модуль находится в состоянии нештатной ситуации на шине (как описано в таблице 37). О нештатных ситуациях сообщается посредством диагностики во время и после запуска.

## 5. КОНФИГУРАЦИЯ

### ВНИМАНИЕ:

Данный вариант рекомендуется использовать на этапе внедрения системы, поскольку он позволяет загружать новые приложения и отключать питание без присутствия всех сконфигурированных модулей.

### 5.1.1.1.7. Как выполнить «горячую» замену

#### ОСТОРОЖНО:

Перед выполнением «горячей» замены необходимо разрядить потенциально возможную статическую энергию, накопившуюся в корпусе. Для этой цели, перед выполнением погрузочно-разгрузочных операций для модулей необходимо прикоснуться (голыми руками) к любой металлической заземленной поверхности. Такая процедура гарантирует, что предельные значения статической энергии модулей не будут превышены.

#### ВНИМАНИЕ:

Для того чтобы гарантировать проверку возвращаемого модулем значения перед его использованием, рекомендуется осуществлять контроль диагностики «горячей» замены в управлении приложениями, разработанном пользователем.

Процедура «горячей» замены описана ниже:

- Отсоедините модуль от стойки объединительной платы, используя предохранительный фиксатор.
- Снимите модуль, сильно потянув за него.
- Вставьте новый модуль в стойку объединительной платы.
- Убедитесь в том, что предохранительный фиксатор полностью подключен.

При необходимости сильнее прижмите модуль к стойке объединительной платы.

В случае выходных модулей удобно отключать точки в процессе замены, чтобы уменьшить образование дуги в разъеме модуля. Для этого необходимо отключить питание или принудительно отключить выходные точки с помощью программных средств. Если нагрузка небольшая, то отключение не требуется.

Важно отметить, что в случаях, когда центральный процессор (ЦП) переходит в режим «Останов» (*Stop*) и светодиод DG начинает мигать 4 раза (в соответствии с таблицей 38, по причине любой нештатной ситуации на шине (как описано в таблице 37), при переходе центрального процессора (ЦП) из режима «Выполнение» (*Run*) в режим «Останов» (*Stop*) работа точек выходных модулей осуществляется в соответствии с конфигурацией модуля. В случае запуска приложения, когда центральный процессор (ЦП) переходит в режим «Останов» (*Stop*) без перехода в режим «Выполнение» (*Run*), выходные модули переводят свои точки в режим защиты от сбоев, другими словами, выключают их (0 В постоянного тока).

Что касается входных модулей, то при извлечении одного модуля из стойки объединительной платы под напряжением состояние логической точки останется в последнем значении. В случае отсоединения разъема состояние логической точки будет переведено в безопасное состояние, что означает нулевое или высокое значение полного сопротивления (импеданса).

#### ВНИМАНИЕ:

Всегда переходите к замене одного модуля за раз, чтобы центральный процессор (ЦП) обновлял состояние модулей.

Ниже, в таблице 38, представлены состояния шины и состояние светодиода DG центрального процессора (ЦП) Nexto. Более подробная информация о состояниях светодиодов диагностики приведена в разделе «[Диагностика посредством светодиодов](#)».

Состояние	Включено, с согласованностью запуска	Включено, с согласованностью запуска только для объявленных модулей	Включено, без согласованности запуска	Выключено	Выключено, только для объявленных модулей	Выключено, без согласованности запуска

## 5. КОНФИГУРАЦИЯ

Необъявленный модуль	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 4х Приложение: Остановлено
Необъявленный модуль (состояние запуска)	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение
Модуль отсутствует	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено
Модуль отсутствует (состояние запуска)	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 2х Приложение: Выполнение
Несовместимая конфигурация	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 2х Приложение: Выполнение	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено
Несовместимая конфигурация (состояние запуска)	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 2х Приложение: Выполнение или Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 2х Приложение: Выполнение
Дублирование адреса слота	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено
Нерабочий модуль	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено	Светодиод DG: Мигает 4х Приложение: Остановлено

Таблица 38: Отношения «горячей» замены и условий

**Примечание:**

Включено, без согласованности запуска: При настройке этого режима «горячей» замены в обычных ситуациях, когда при запуске системы присутствует несовместимый модуль, приложение переходит из режима «*Останов*» (*Stop*) в режим «*Выполнение*» (*Run*). Тем не менее, если данный модуль сконфигурирован как NX5000 или NX5001, а на его месте стоит другой модуль, то приложение останется в состоянии «*Останов*» (*Stop*).

---

## 5. КОНФИГУРАЦИЯ

### 5.1.1.2. Области сохраняющей и постоянной памяти

Центральный процессор (ЦП) Nexto позволяет использовать символические переменные и выходные переменные прямого представления в качестве сохраняемых или постоянных переменных.

Выходные переменные прямого представления, которые будут сохраняемыми либо постоянными, должны быть объявлены в *общих параметрах* центрального процессора (ЦП), как описано в разделе «[Конфигурация центрального процессора \(ЦП\)](#)». Символические имена также могут быть присвоены этим выходным переменным прямого представления с помощью директивы AT, а также с использованием ключевого слова RETAIN («СОХРАНЯЕМЫЙ») или PERSISTENT («ПОСТОЯННЫЙ») при их объявлении. Например, %QB4096 и %QB20480 в ретенционной и персистентной памяти, соответственно:

```
ПРОГРАММА UserPrg
VAR RETAIN
byRetentiveVariable_01 AT %QB4096: BYTE;
END_VAR VAR
PERSISTENT
byPersistentVariable_01 AT %QB20480: BYTE;
END_VAR
```

В случае если символические переменные, объявленные с помощью директивы AT, не находятся в соответствующей сохраняющей и/или постоянной памяти, то при генерации кода в MasterTool могут возникнуть ошибки (как описано в разделе «[Общие параметры](#)», конфигурация «*Соответствующая область сохраняющей и постоянной памяти в %Q*» (*Consist retain and persistent area in %Q*)), информирующие о том, что в области сохраняющей или постоянной памяти определены несохраняемые либо непостоянные переменные.

Что касается символических переменных, которые будут сохраняемыми или постоянными, то только сохраняемые переменные могут быть локальными или глобальными, поскольку постоянные символические переменные всегда должны быть глобальными. Для объявления сохраняющих символических переменных необходимо использовать ключевое слово *RETAIN* («СОХРАНЯЕМЫЙ»). Например, для локальных переменных:

```
ПРОГРАММА UserPrg
VAR RETAIN wLocalSymbolicRetentiveVariable_01: WORD;
END_VAR
```

Или для глобальных переменных, объявленных внутри списка глобальных переменных:

```
VAR_GLOBAL RETAIN wGlobalSymbolicRetentiveVariable_01: WORD;
END_VAR
```

С другой стороны, постоянные символические переменные должны быть объявлены в объекте «*Постоянные переменные*» (*Persistent Variables*), добавляемом в приложение. Данные переменные будут глобальными и будут объявлены в объекте следующим образом:

```
VAR_GLOBAL PERSISTENT RETAIN wGlobalSymbolicPersistentVariable_01 :
    WORD;
END_VAR
```

```
VAR_GLOBAL PERSISTENT RETAIN wGlobalSymbolicPersistentVariable_01 : WORD;
END_VAR
VAR_GLOBAL RETAIN wGlobalSymbolicRetentiveVariable_01 :
    WORD;
END_VAR
```

**ВНИМАНИЕ:**

Для гибкого использования сохраняющей и постоянной памяти необходимо использовать MasterTool IEC XE версии 2.03 и выше.

### 5.1.1.3. Конфигурации TCP

Некоторые из расширенных конфигураций влияют на поддерживаемые протоколы центральных процессоров (ЦП) серии Nexto, поскольку они подключены к сетевому уровню TCP, а именно:

- Начальная блокировка по превышению лимита времени
- Задержка отправки подтверждения приема сигнала (АСК)

Центральный процессор (ЦП) Nexto, прежде чем ответить на любой запрос, как и любое другое оборудование локальной сети Ethernet, использующее транспортный уровень TCP, требует открытия коммуникационного порта, иными словами, установления соединения.

Количество интерфейсных соединений ограничено и после достижения своего лимита просто не устанавливает никаких других соединений. Это может создавать проблемы для установленных соединений в режиме сервера, поскольку закрытие соединений зависит от другого оборудования, клиента.

Транспортный уровень TCP, отвечающий за доставку сообщений от места отправления к месту назначения, имеет некоторые параметры с блокировками по превышению лимита времени, весьма распространенные в общих коммуникационных протоколах. Такие параметры предназначены для восстановления связи после обнаружения сбоев. Пользователь должен быть внимателен при настройке блокировок по превышению лимита времени, поскольку возможны конфликты со значениями, настроенными на прикладном уровне. Поскольку конфигурация TCP является эталонной для всех конфигурируемых экземпляров, то время будет действительным, если оно меньше настроенного внутри протокола:

Начальная блокировка по превышению лимита времени: указывает на то, через какое время после первой передачи сообщения оно должно быть повторно передано, если предположить, что оно не было получено устройством назначения. При каждой повторной передаче значение блокировки по превышению лимита времени удваивается. Количество попыток передачи связано с блокировками по превышению лимита времени связи, настроенным в протоколе. Это будет максимальное время до отказа от доставки сообщения, когда конкретизируется сбой передачи. Помимо этого, важно подчеркнуть, что для центральных процессоров (ЦП) серии Nexto существует максимальное количество попыток. Это количество устанавливается за пять попыток до установления соединения и за три попытки после этого. При достижении максимального количества повторных попыток процесс попытки соединения перезапускается. Более подробная информация об использовании параметров блокировки по превышению лимита времени приведена в разделе «[Конфигурация протоколов](#)», поскольку в зависимости от ситуации они могут быть различными. Важно подчеркнуть, что данный параметр используется только при установке соединения, после чего для расчета новой блокировки по превышению лимита времени используется статистика последних соединений.

Пример параметров начальной блокировки по превышению лимита времени и блокировки по превышению лимита времени связи внутри сервера MODBUS TCP (MODBUS TCP Server) с учетом неполученного подтверждения для следующих значений: 300 для начальной блокировки по превышению лимита времени (=300 мс) и 3000 для блокировки по превышению лимита времени связи (=3000 мс):

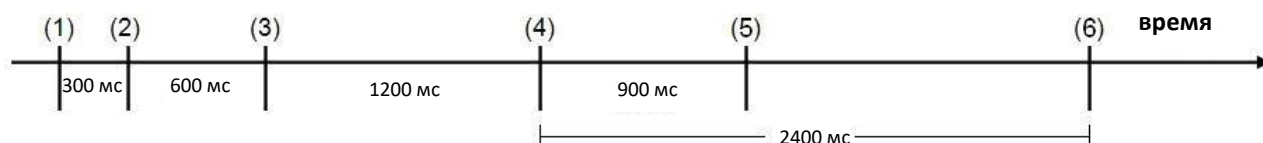


Рисунок 40: Начальная блокировка по превышению лимита времени и блокировка по превышению лимита времени связи

Условные обозначения:

1. Мгновение передачи сообщения.
2. Первая попытка передачи сообщения после начальной блокировки по превышению лимита времени.

## 5. КОНФИГУРАЦИЯ

3. Вторая попытка передачи сообщения, после двукратного превышения значения начальной блокировки по превышению лимита времени.
4. Третья попытка передачи сообщения, после двукратного превышения значения последней блокировки по превышению лимита времени.
5. Прекращение передачи сообщения и индикация сбоя по превышению значения блокировки по превышению лимита времени связи (общее время до сдачи:  $300 + 600 + 1200 + 900 = 3000$  мс).
6. Это будет четвертая попытка передачи сообщения, после двукратного превышения значения последней блокировки по превышению лимита времени, но значение блокировки по превышению лимита времени связи, настроенное в протоколе, превышено, и произошла индикация сбоя.

Задержка отправки подтверждения приема сигнала (ACK): определяет максимальное время ожидания интерфейсом передачи подтверждения приема сигнала (ACK) TCP. Данное подтверждение приема сигнала (ACK) отвечает за подтверждение получения сообщения, в случае MODBUS, устройством назначения. Установка данного поля уменьшает количество сообщений, циркулирующих по сети. Данный механизм объясняется следующим образом:

- Все сообщения запроса, отправляемые клиентом на сервер, должны быть подтверждены сервером посредством сообщения о подтверждении приема сигнала (ACK), передаваемого клиенту.
- Все ответные сообщения, посылаемые клиентом серверу, должны быть подтверждены сервером посредством сообщения о подтверждении приема сигнала (ACK), передаваемого клиенту.
- Если одна из частей не получит сообщение о подтверждении приема сигнала (ACK) в течение времени, определяемого блокировкой по превышению лимита времени TCP, сообщение будет повторно передано по адресу происхождения (см. Параметр «Количество попыток TCP» (TCP number of tries)).
- Сообщение о подтверждении приема сигнала (ACK) не обязательно должно быть эксклюзивным. Подтверждение приема сигнала (ACK), которое сервер должен послать клиенту в момент получения запроса, может быть включено в то же ответное сообщение, а подтверждение приема сигнала (ACK), которое клиент должен послать серверу в момент запроса, может быть включено в то же сообщение, что включает в себя следующий запрос.

На следующих рисунках показана разница между отправкой немедленного и откалиброванного подтверждения приема сигнала (ACK):

**ВНИМАНИЕ:**  
Центральные процессоры (ЦП) NX3003, NX3004 и NX3005 ведут себя несколько иначе. Они не учитывают значение, заданное в параметре «Начальная блокировка по превышению лимита времени» (Initial Time-out, а учитывают только параметр «Блокировка по превышению лимита времени связи» (communication Time-out). Но даже в этом случае значение, сконфигурированное для параметра «Блокировка по превышению лимита времени связи» (communication Time-out), используется, если оно составляет менее 3 секунд, и никогда не будет удваиваться при повторных попытках. Но если «Блокировка по превышению лимита времени связи» (communication Time-out) имеет значение, превышающее 3 секунды, то это значение игнорируется, а начальное значение равно 3 секундам и удваивается при каждой повторной попытке.



Рисунок 41: Пример незамедлительной отправки подтверждения приема (ACK) (=0)

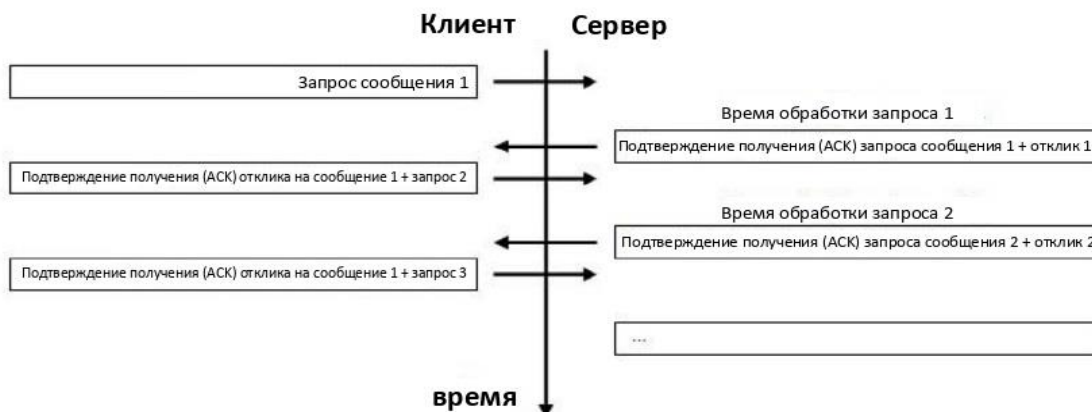


Рисунок 42: Пример откалиброванной отправки подтверждения приема (ACK)

#### 5.1.1.4. Параметры проекта

Параметры проекта центрального процессора (ЦП) связаны с конфигурацией для обновления ввода/вывода при использовании задач проекта и согласованностью области сохраняющей и постоянной памяти в %Q.

Конфигурация	Описание	По умолчанию	Опции
Соответствующая область сохраняющей и постоянной памяти в %Q	Устанавливает соответствие между областями сохраняющей и постоянной памяти в %Q	Не отмечено	- Отмечено - Не отмечено
Включить обновление входа/выхода на задачу	Обновление входных и выходных данных в тех задачах, где они используются	Не отмечено	- Отмечено - Не отмечено
Включить сохраняемые и постоянные переменные в функциональных блоках	Настройка, разрешающая использование сохраняемых и постоянных переменных в функциональных блоках	Не отмечено	- Отмечено - Не отмечено

Таблица 39: Параметры проекта центрального процессора (ЦП)

#### 5.1.2. Конфигурация внешних событий

Внешнее событие представляет собой функцию, присутствующую в центральном процессоре (ЦП), которая позволяет сконфигурированному пользователем цифровому входу при активации запускать выполнение определенной задачи с заданным пользователем кодом. Таким образом, можно посредством данного входа при его срабатывании прервать выполнение основного приложения и запустить установленное приложение в задаче *ExternInterruptTask00*, которая имеет более высокий приоритет по сравнению с другими прикладными задачами. Поскольку обновление входов и выходов происходит в контексте задачи *MainTask*, то в момент вызова задачи «Внешнее событие» (External Event) входные и выходные данные не обновляются. При необходимости следует воспользоваться функциями обновления ввода/вывода.

Важно также отметить, что во избежание генерации нескольких событий за очень короткий промежуток времени было введено ограничение на обработку такого типа событий через каждые 10 мс, т.е. если в течение 10 мс после первого события происходит два или более, то второе и последующие события отбрасываются. Данное ограничение введено для того, чтобы внешнее событие, генерируемое неконтролируемым образом, не препятствовало работе центрального процессора (ЦП), поскольку данная задача имеет более высокий приоритет по сравнению с другими.

Для конфигурирования внешнего события необходимо вставить модуль цифрового входа и выполнить описанные ниже конфигурации в центральном процессоре (ЦП) посредством программного обеспечения программатора (устройства программирования) MT8500.

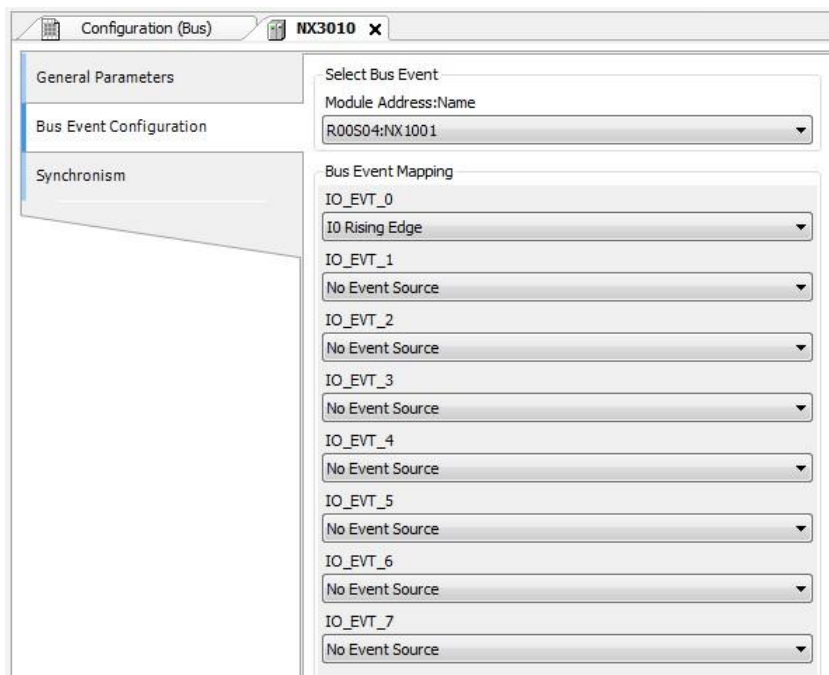


Рисунок 43: Экран конфигурации для внешнего события на центральном процессоре (ЦП)

На вкладке конфигурации «Внешнее событие» в настройках центрального процессора (ЦП) необходимо выбрать, какой модуль будет являться источником прерывания, в поле «Адрес модуля: Имя» (*Module Address: Name*). Затем необходимо выбрать, какой вход данного модуля будет отвечать за генерацию события (*IO\_EVT\_0*). При подобном выборе могут быть выбраны варианты, описанные на рисунке, приведенном ниже.

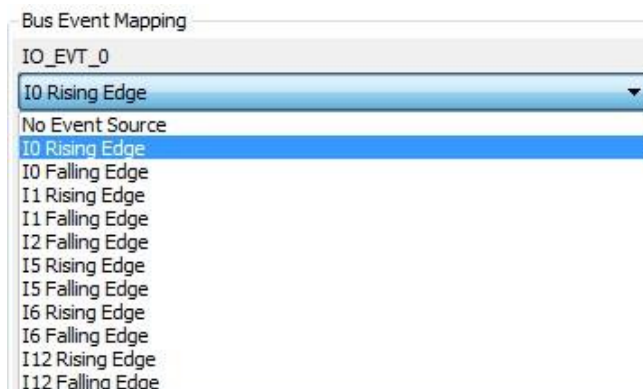


Рисунок 44: Опции внешнего источника событий модуля NX1001

В дополнение к конфигурированию центрального процессора (ЦП) необходимо сконфигурировать задачу, отвечающую за выполнение заданных пользователем действий. В этом случае пользователь должен использовать профиль проекта, поддерживающий внешние события. Более подробная информация приведена в разделе «[Профили проектов](#)». На экране конфигурации задачи *ExternInterruptTask00* (рисунок, приведенный ниже) необходимо выбрать источник события в соответствующем поле. В данном случае следует выбрать *IO\_EVT\_0*, поскольку другие исходные источники (*IO\_EVT\_1 - IO\_EVT\_7*) недоступны. В последовательности действий в поле *структурной единицы программы (POU)* следует проверить, правильно ли выбрана *структурная единица программы (POU)*, поскольку она будет использоваться пользователем для определения действий, которые необходимо выполнить при наступлении внешнего события.



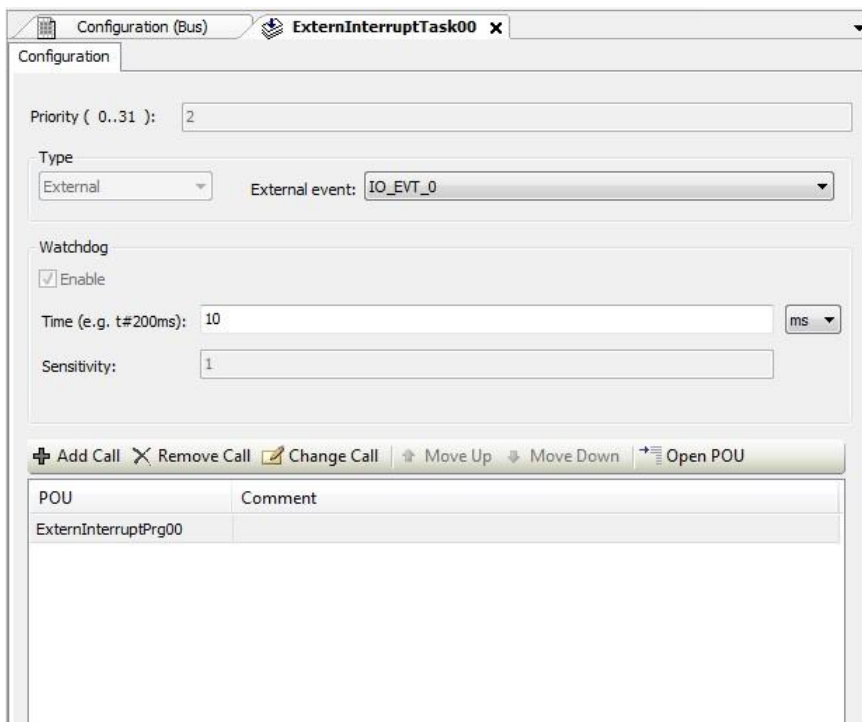


Рисунок 45: Экран конфигурации ExternInterruptTask00

### 5.1.3. Синхронизация времени

Для синхронизации времени центральные процессоры (ЦП) серии Nexto используют протокол SNTP (*Простой Протокол Сетевого Времени*) или синхронизацию по стандарту МЭК 60870-5-104 (IEC 60870-5-104).

Для использования протоколов синхронизации времени пользователю необходимо установить следующие параметры на вкладке «Синхронность» (*Synchronism*), доступ к которой осуществляется через центральный процессор (ЦП) в дереве устройств:

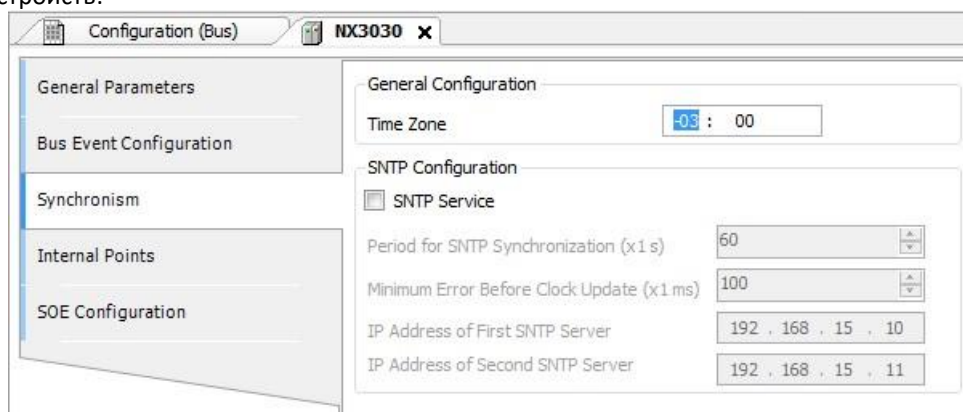


Рисунок 46: Конфигурация простого протокола сетевого времени (SNTP)

Конфигурация	Описание	По умолчанию	Опции
Часовой пояс (чч:мм)	Часовой пояс местонахождения пользователя. Можно вставить часы и минуты.	-3:00	от 12:59 до +13:59

## 5. КОНФИГУРАЦИЯ

Сервис простого протокола сетевого времени (SNTP)	Включает сервис простого протокола сетевого времени (SNTP)	Выключено	Выключено Включено
Период для синхронизации простого протокола сетевого времени (SNTP) (x1 с)	Временной интервал запросов на синхронизацию (секунды).	60	от 1 до 255
Минимальное число ошибок до поправки к эталонному времени (x1 мс)	Значение смещения, допустимое между сервером и клиентом (миллисекунды).	100	от 1 до 65519
IP-адрес первого сервера простого протокола сетевого времени (SNTP)	IP-адрес первичного сервера простого протокола сетевого времени (SNTP)	192.168.15.10	от 1.0.0.1 до 223.255.255.254
IP-адрес второго сервера простого протокола сетевого времени (SNTP)	IP-адрес вторичного сервера простого протокола сетевого времени (SNTP)	192.168.15.11	от 1.0.0.1 до 223.255.255.254

Таблица 40: Конфигурации простого протокола сетевого времени (SNTP)

### Примечания:

Сервер простого протокола сетевого времени (SNTP): Можно задать предпочтительный и другой вторичный адрес для получения доступа к серверу простого протокола сетевого времени (SNTP) и, соответственно, для получения синхронизации времени. Если оба поля пусты, то служба простого протокола сетевого времени (SNTP) остается отключенной.

Часовой пояс: Конфигурация часового пояса используется для преобразования местного времени в формат UTC (всемирное координированное время) и наоборот. В то время как одни источники синхронизации используют местное время (протокол МЭК 60870-5-104 (IEC K 60870-5-104), функция SetDateAndTime), другие - время в формате UTC (простой протокол сетевого времени (SNTP)). Как правило, время в формате UTC используется для снабжения событий метками (DNP3, МЭК 60870-5-104 (IEC 60870-5-104) и файл регистрации устройств MasterTool), а местное время используется функциями других центральных процессоров (ЦП) (функция GetDateAndTime, Информация о дате и времени диагностики одним касанием (OTD)).

В проекте допускается включение более одного источника синхронизации. Тем не менее, устройство не поддерживает синхронизацию от более чем одного источника синхронизации во время работы. Поэтому неявным образом определен механизм приоритетов. Синхронизация по протоколу простого протокола сетевого времени (SNTP) является более приоритетной, чем по протоколу IEC 60870-5-104 МЭК 60870-5-104 (IEC K 60870-5-104). Таким образом, если оба источника включены и присутствует сервер простого протокола сетевого времени (SNTP), то он будет отвечать за синхронизацию часов центрального процессора (ЦП), а любая команда синхронизации от МЭК 60870-5-104 (IEC K 60870-5-104) будет отклонена.

### 5.1.3.1. Простой протокол сетевого времени (SNTP)

При включении данной опции, центральный процессор (ЦП) будет работать как клиент простого протокола сетевого времени (SNTP), то есть отправлять запросы на синхронизацию времени на сервер SNTP/NTP, который может находиться как в локальной сети, так и в Интернете. Клиент простого протокола сетевого времени (SNTP) работает с разрешением 1 мс, но с точностью до 100 мс. Точность синхронизации времени по простому протоколу сетевого времени (SNTP) зависит от конфигурации протокола (минимальная ошибка обновления часов) и особенностей локальной сети Ethernet, в которой он находится, если клиент и сервер находятся в одной сети (локальная) или в разных сетях (удаленная). Как правило, точность составляет порядка десятков миллисекунд.

Центральный процессор (ЦП) посылает циклические запросы на синхронизацию в соответствии со временем, установленным в поле «Период для синхронизации простого протокола сетевого времени (SNTP)» (*Period for SNTP Synchronization*). При первой попытке синхронизации, сразу после запуска сервиса, запрос направляется на первый сервер, заданный в IP-адресе первого сервера. Если он не отвечает, то запросы направляются на второй сервер,

---

## 5. КОНФИГУРАЦИЯ

заданный в IP-адресе второго сервера, что обеспечивает резервирование серверов простого протокола сетевого времени (SNTP). Если и второй сервер не отвечает, то повторяется тот же процесс попытки синхронизации, но уже после истечения периода синхронизации. Другими словами, при каждом периоде синхронизации центральный процессор (ЦП) предпринимает попытку подключения к каждому серверу по одному разу, при этом он пытается подключиться ко второму серверу в случае, если первый не отвечает. Время ожидания ответа от сервера простого протокола сетевого времени (SNTP) по умолчанию определено в 5 с и не может быть изменено.

В случае если после синхронизации разница между текущим временем центрального процессора (ЦП) и временем, полученным сервером, превышает значение, установленное в параметре «Минимальная погрешность перед поправкой к эталонному времени» (*Minimum Error Before Clock Update*), то время центрального процессора (ЦП) обновляется. Простой протокол сетевого времени (SNTP) использует время в формате UTC (всемирное координированное время), поэтому необходимо правильно установить параметр «Часовой пояс» (*Time Zone*), чтобы время, считанное простым протоколом сетевого времени (SNTP), было правильно переведено в местное время.

Процесс выполнения клиента простого протокола сетевого времени (SNTP) можно проиллюстрировать следующими шагами:

1. Попытка синхронизации через первый сервер. В случае успешной синхронизации центральный процессор (ЦП) выжидает время для новой синхронизации («Период для синхронизации простого протокола сетевого времени (SNTP)» (*Period for SNTP Synchronization*)) и снова синхронизируется с данным сервером, используя его в качестве основного. В случае неудачи (сервер не отвечает менее чем за 5 с) выполняется шаг 2.
2. Попытка синхронизации через второй сервер. В случае успешной синхронизации центральный процессор (ЦП) выжидает время для новой синхронизации («Период для синхронизации простого протокола сетевого времени (SNTP)» (*Period for SNTP Synchronization*)) и пытается синхронизироваться с данным сервером, используя первичный сервер. В случае неудачи (сервер не отвечает менее чем за 5 с) выжидается время, соответствующее Периоду синхронизации, и снова выполняется шаг 1.

Поскольку время ожидания отклика сервера простого протокола сетевого времени (SNTP) составляет 5 с, пользователь должен обратить внимание на значение параметра «Период синхронизации» (*Synchronization Period*), составляющее меньше 10 с. В случае если первичный сервер не отвечает, время синхронизации составит минимум 5 с (ожидание отклика первичного сервера и попытка синхронизации со вторичным сервером). В случае если ни первичный, ни вторичный сервер не отвечают, время синхронизации составит не менее 10 с (ожидание ответа обоих серверов и попытка нового соединения с первым сервером).

В зависимости от подсети сервера простого протокола сетевого времени (SNTP) клиент будет использовать для запросов синхронизации интерфейс локальной сети Ethernet, находящийся в соответствующей подсети. Если в той же подсети, что и сервер, не сконфигурирован ни один интерфейс, запрос может быть сделан любым интерфейсом, который может найти маршрут к серверу.

### ВНИМАНИЕ:

Служба простого протокола сетевого времени (SNTP) зависит в своей конфигурации только от пользовательского приложения. Таким образом, выполнение данной службы будет осуществляться даже когда центральный процессор (ЦП) находится в режимах «Останов» (*STOP*) или «Точка контрольного прерывания» (*BREAKPOINT*), если в центральном процессоре (ЦП) присутствует приложение с включенным и правильно настроенным клиентом простого протокола сетевого времени (SNTP).

### 5.1.3.2. Скорректированное летнее время (DST)

Настройка скорректированного летнего времени (DST) должна осуществляться косвенно через функцию *SetTimeZone*, которая изменяет часовой пояс, применяемый к часам реального времени (RTC). В начале применения скорректированного летнего времени (DST) необходимо использовать функцию для увеличения часового пояса на один час. В конце применения скорректированного летнего времени (DST) она используется для уменьшения часового пояса на один час.

Более подробную информацию см. в разделе «[Часы реального времени \(RTC\)](#)».

## 5.2. Конфигурация последовательных интерфейсов

### 5.2.1. COM 1

Коммуникационный интерфейс COM 1 состоит из разъема типа мама (с внутренней резьбой) DB9 для стандартов RS-422 и RS-485. Он позволяет осуществлять связь «точка-точка» или сетевую связь по открытым протоколам ведомого устройства MODBUS RTU (MODBUS RTU slave) или ведущего устройства MODBUS RTU (MODBUS RTU master).

При использовании протокола ведущего / ведомого устройства MODBUS некоторые из данных параметров (например, «Последовательный режим» (Serial Mode), «Биты данных» (Data Bits), «Пороговое значение UART RX» (UART RX Threshold) и «Последовательные события» (Serial Events)) автоматически настраиваются программой MasterTool для корректной работы данного протокола.

Ниже описаны параметры, которые должны быть сконфигурированы для правильного функционирования приложения.

Конфигурация	Описание	По умолчанию	Опции
Последовательный тип	Конфигурация типа последовательного канала	RS-485	RS-422 и RS-485
Скорость передачи данных в бодах	Конфигурация скорости порта последовательной связи	115200	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит / сек
Контроль по четности	Конфигурация четности последовательного порта	Нет	Нечетный Четный Метка Сигнал логического нуля Нет
Биты данных	Устанавливает количество битов данных в каждом символе последовательной связи	8	5, 6, 7 и 8
Стоповые биты	Устанавливает стоповые биты последовательного порта	1	1, 1.5 и 2
Режим поочередной передачи данных	Устанавливает конфигурацию режима работы последовательного порта	Нормальный режим	- Расширенный режим: Расширенный режим работы последовательной связи, при котором передается информация о полученном фрейме данных (см. примечание к разделу «COM 1»). - Нормальный режим: Нормальный режим работы последовательной связи.

Таблица 41: Конфигурация стандартного последовательного интерфейса RS-485/RS-422

Конфигурация «Биты данных» (Data Bits) последовательных интерфейсов ограничивает поля «Стоповые биты» (Stop Bits) и «Четность связи» (Communication Parity). Поэтому количество стоповых битов и метод контроля четности будут зависеть от количества бит данных.

В приведенной ниже таблице представлены допустимые конфигурации интерфейсов.

Биты данных	Стоповые биты	Четность
5	1, 1.5	БЕЗ ЧЕТНОСТИ, НЕЧЕТНЫЕ, ЧЕТНЫЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА ЕДИНИЦЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА НУЛЮ

## 5. КОНФИГУРАЦИЯ

6	1, 2	БЕЗ ЧЕТНОСТИ, НЕЧЕТНЫЕ, ЧЕТНЫЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА ЕДИНИЦЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА НУЛЮ
7	1, 2	БЕЗ ЧЕТНОСТИ, НЕЧЕТНЫЕ, ЧЕТНЫЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА ЕДИНИЦЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА НУЛЮ
8	1, 2	БЕЗ ЧЕТНОСТИ, НЕЧЕТНЫЕ, ЧЕТНЫЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА ЕДИНИЦЕ, ЧЕТНОСТЬ ВСЕГДА РАВНА НУЛЮ

Таблица 42: Конкретные конфигурации

### 5.2.1.1. Расширенные конфигурации

Расширенные конфигурации связаны с управлением последовательной связью, т.е. когда необходимо использовать более точное управление передачей и приемом данных.

Конфигурация	Описание	По умолчанию	Опции
Пороговое значение UART RX	Количество байтов, которое должно быть получено для генерации нового прерывания универсального асинхронного приёмопередатчика (UART). Низкие значения делают МЕТКУ ВРЕМЕНИ (TIMESTAMP) более точной при использовании режима EXTENDED MODE (РАСШИРЕННЫЙ РЕЖИМ) и сводят к минимуму ошибки превышения. Однако, если значения слишком малы, они могут вызвать несколько прерываний, что может привести к задержке центрального процессора (ЦП).	8	1, 4, 8 и 14

Таблица 43: Расширенные конфигурации стандартного последовательного интерфейса RS-485/RS-422

## 5.3. Конфигурация интерфейсов локальной сети Ethernet

В центральных процессорах (ЦП) Nexto может быть предусмотрено большее количество мест расположения интерфейсов сети Ethernet. Центральный процессор (ЦП) NX3004 имеет только интерфейс NET 1.

### 5.3.1. Локальные интерфейсы сети Ethernet

#### 5.3.1.1. NET 1

Интерфейс NET 1 состоит из коммуникационного разъема RJ45 стандарта 10/100Base-TX. Он позволяет осуществлять связь «точка-точка» или сетевую связь по следующим открытым протоколам, например: клиент MODBUS TCP (MODBUS TCP Client), MODBUS RTU через клиент TCP (MODBUS RTU via TCP Client), сервер MODBUS TCP (MODBUS TCP Server) и MODBUS RTU через сервер TCP (MODBUS RTU via TCP Server).

Ниже описаны параметры, которые должны быть сконфигурированы для правильного функционирования приложения.

Конфигурация	Описание	По умолчанию	Опции
Автоматическое получение IP-адреса	Включение функции DHCP Client (Клиент DHCP) на устройстве для автоматического назначения IP-адресов	Не отмечено	Отмечено или Не отмечено

## 5. КОНФИГУРАЦИЯ

IP-адрес	IP-адрес контроллера в шине сети Ethernet	192.168.15.1	От 1.0.0.1 до 223.255.255.254
Маска подсети	Маска подсети контроллера в шине сети Ethernet	255.255.255.0	От 128.0.0.0 до 255.255.255.252
Адрес шлюза	Адрес шлюза контроллера в шине сети Ethernet	192.168.15.253	От 1.0.0.1 до 223.255.255.254

Таблица 44: Конфигурация локальной сети NET 1

### 5.3.2. Зарезервированные порты TCP

Следующие порты TCP интерфейсов сети Ethernet, как локальные, так и удаленные, используются службами центрального процессора (ЦП) (в зависимости от доступности согласно таблице «[Протоколы](#)») и, следовательно, зарезервированы и не должны использоваться пользователем.

Сервис	Порт
Веб-страница диагностики	80
SNTP	123
SNMP	161
MODBUS TCP	502*
MasterTool MT8500	1217* / 1740 / 1741 / 1742 / 1743
Сервис	Порт
SQL Server	1433
MQTT	1883* / 8883*
EtherNet/IP	2222 / 44818
IEC 60870-5-104	2404*
OPC UA	4840
WEBVISU	8080
CODESYS ARTI	11740

Таблица 45: Зарезервированные порты TCP

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

### 5.4. Конфигурация протоколов

Независимо от протоколов, используемых в каждом конкретном приложении, для центральных процессоров (ЦП) серии Nexто установлены некоторые максимальные ограничения для каждой модели центрального процессора. В основном существует два различных типа коммуникационных протоколов: символьное преобразование и преобразование прямого представления. Максимальное количество преобразований, а также максимальное количество протоколов (экземпляров) определено в следующей таблице:

	NX3004
Преобразованные токи	20480
Преобразования	5120
Запросы	512
NET – Экземпляры клиента или сервера (на NET / итого)	4
COM (n) – Экземпляры ведущего или ведомого устройства	1

Центры управления	-
-------------------	---

Таблица 46: Предельные значения протоколов для отдельно взятого центрального процессора (ЦП)

Примечания:

Преобразованные точки: Обозначает максимальное количество точек преобразования, поддерживаемое центральным процессором (ЦП). Каждое преобразование поддерживает 1 или несколько точек преобразования, в зависимости от размера данных при использовании с переменными типа ARRAY («МАССИВ»).

Преобразования: «Преобразование» представляет собой связь между внутренней переменной приложения и объектом прикладного протокола. В данном поле указывается максимальное количество преобразований, поддерживаемых центральным процессором (ЦП). Оно соответствует сумме всех преобразований, выполненных в рамках экземпляров коммуникационных протоколов и соответствующих им устройств.

Запросы: Сумма запросов протоколов связи, заявленных на устройствах, не может превышать максимального количества запросов, поддерживаемых центральным процессором (ЦП).

NET – Экземпляры клиента или сервера: В этом поле задается максимальное количество экземпляров протокола на один интерфейс сети Ethernet, а также общее максимальное количество, распределенное по всем интерфейсам сети Ethernet системы.

COM (n) – Экземпляры ведущего или ведомого устройства: В силу своих особенностей каждый последовательный интерфейс поддерживает только один экземпляр протокола связи. Примеры экземпляров, совместимых с последовательными интерфейсами: ведущее устройство MODBUS RTU (MODBUS RTU Master) и ведомое устройство MODBUS RTU (MODBUS RTU Slave).

Центры управления: «Центры управления» подразумевают все клиентские устройства, подключенные к центральному процессору (ЦП) по протоколу МЭК 60870-5-104 (IEC 60870-5-104). Данное поле сообщает максимальное количество клиентских устройств типа «Центр управления», поддерживаемых центральным процессором (ЦП). Соответствует сумме всех клиентских устройств протокола связи сервера МЭК 60870-5-104 (IEC 60870-5-104 Server) (не включает в себя ведущие устройства или клиентов из протоколов ведомого устройства MODBUS RTU (MODBUS RTU Slave), сервера MODBUS (MODBUS Server) и сервера DNP3 (DNP3 Server)).

Ограничения протокола MODBUS для прямого представления и символьного преобразования для центральных процессоров можно увидеть в таблицах 47 и 48 соответственно.

Ограничения	Ведущее устройство MODBUS RTU	Ведомое устройство MODBUS RTU	Клиент MODBUS Ethernet	Сервер MODBUS Ethernet
Преобразования на каждый экземпляр	128	32	128	32
Устройства на каждый экземпляр	64	1 <sup>(1)</sup>	64	64 <sup>(2)</sup>
Преобразования на устройства	32	32	32	32
Одновременные запросы на экземпляр	-	-	128	64
Одновременные запросы на устройство	-	-	8	64

Таблица 47: Ограничения протокола MODBUS для прямого представления

Примечания:

Устройства на каждый отдельно взятый экземпляр:

- Протоколы ведущего устройства или клиента: количество ведомых устройств или серверов, поддерживаемых каждым экземпляром протокола ведущего или ведомого устройства.

## 5. КОНФИГУРАЦИЯ

- Протокол ведущего устройства MODBUS RTU (MODBUS RTU Slave): сообщенное ограничение <sup>(1)</sup> 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 либо возможности объявлять или конфигурировать ведущее устройство в экземпляре протокола ведомого устройства MODBUS RTU. Ведущее устройство будет иметь доступ ко всем преобразованиям, выполненным непосредственно на экземпляре протокола ведомого устройства MODBUS RTU.
- Протокол сервера MODBUS RTU: информированное ограничение <sup>(2)</sup> относится к интерфейсам сети Ethernet, которые ограничивают количество соединений, которые могут быть установлены с другими устройствами через один интерфейс сети Ethernet. Нет необходимости и возможности объявлять или конфигурировать клиентские устройства в экземпляре протокола сервера MODBUS. Все клиентские устройства будут иметь доступ ко всем связям, созданным непосредственно в экземпляре протокола сервера MODBUS.

Преобразования на одно устройство: Максимальное количество преобразований на устройство, несмотря на то, что оно указано выше, также ограничено максимальным количеством преобразований протокола. Также следует учитывать максимальное количество преобразований на центральный процессор (ЦП), как показано в таблице 46.

Одновременные запросы на один экземпляр: Количество запросов, которые могут быть одновременно переданы каждым экземпляром протокола клиента либо получены каждым экземпляром протокола сервера. Экземпляры протокола MODBUS RTU, ведущего или ведомого устройства, не поддерживают одновременные запросы.

Одновременные запросы на устройство: Количество запросов, которые могут одновременно передаваться каждому устройству сервера MODBUS либо одновременно приниматься каждым устройством клиента MODBUS. Устройства MODBUS RTU, ведущего или ведомого устройства, не поддерживают одновременные запросы.

Ограничения	Ведущее устройство MODBUS RTU	Ведомое устройство MODBUS RTU	Клиент MODBUS Ethernet	Сервер MODBUS Ethernet
Устройств на каждый экземпляр	64	1 <sup>(1)</sup>	64	64 <sup>(2)</sup>
Запросов на устройство	32	-	32	-
Одновременных запросов на экземпляр	-	-	128	64
Одновременных запросов на устройство	-	-	8	64

Таблица 48: Ограничения протокола MODBUS для символьных преобразований

Примечания:

Устройства на каждый отдельно взятый экземпляр:

- Протоколы ведущего устройства или клиента: количество ведомых устройств или серверов, поддерживаемых каждым экземпляром протокола ведущего устройства или клиента.
- Протокол ведомого устройства MODBUS RTU (MODBUS RTU Slave): сообщенное ограничение <sup>(1)</sup> 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 возможности объявлять или конфигурировать ведущее устройство в экземпляре протокола ведомого устройства MODBUS RTU. Ведущее устройство будет иметь доступ ко всем преобразованиям, выполненным непосредственно на экземпляре протокола ведомого устройства MODBUS RTU.
- Протокол сервера MODBUS RTU: информированное ограничение <sup>(2)</sup> относится к интерфейсам сети Ethernet, которые ограничивают количество соединений, которые могут быть установлены с другими устройствами через один интерфейс сети Ethernet. Нет необходимости и возможности объявлять или конфигурировать клиентские устройства в экземпляре протокола сервера MODBUS. Все клиентские устройства будут иметь доступ ко всем связям, созданным непосредственно в экземпляре протокола сервера MODBUS.

Запросы по устройствам: Количество запросов, таких как считывание или запись регистров временного хранения, которые могут быть сконфигурированы для каждого из устройств (ведомых или серверов) из экземпляров протоколов



## 5. КОНФИГУРАЦИЯ

ведущего устройства или клиента. Данный параметр не относится к экземплярам протоколов ведомого устройства или сервера.

Одновременные запросы на один экземпляр: Количество запросов, которые могут быть одновременно переданы каждым экземпляром протокола клиента либо получены каждым экземпляром протокола сервера. Экземпляры протокола MODBUS RTU, ведущего или ведомого устройства, не поддерживают одновременные запросы.

Одновременные запросы на устройство: Количество запросов, которые могут одновременно передаваться каждому устройству сервера MODBUS либо одновременно приниматься каждым устройством клиента MODBUS. Устройства MODBUS RTU, ведущего или ведомого устройства, не поддерживают одновременные запросы.

### 5.4.1. Логика работы протокола x состояние центрального процессора (ЦП)

В приведенной ниже таблице подробно представлена логика работы каждого конфигурируемого протокола в центральных процессорах (ЦП) серии Nexto в каждом состоянии работы.

Протокол	Тип	Рабочее состояние центрального процессора (ЦП)					
		«Останов» (STOP)			«Выполнение» (RUN)		
		После загрузки; перед запуском приложения	После перехода приложения в режим «ОСТАНОВ (ПАУЗА)»	После исключения	Не резервный или активный	Резервирование в резервном режиме	После точки контрольного прерывания в MainPrg
MODBUS Symbol	Ведомое устройство /Сервер	✓	✓	✓	✓	✓	✓
	Ведущее устройство /Клиент	✗	✗	✗	✓	✓	✓
MODBUS	Ведомое устройство /Сервер	✗	✗	✗	✓	✓	✗
	Ведущее устройство /Клиент	✗	✗	✗	✓	✓	✓
SOE (DNP3)	Удаленная станция	✓	✓	✓	✓	✗	✓
МЭК 60870-5-104	Сервер	✗	✗	✗	✓	✗	✓
EtherCAT	Мастер	✗	✗	✗	✓	Не применимо	✗
OPC DA	Сервер	✓	✓	✓	✓	✗	✓
OPC UA	Сервер	✓	✓	✓	✓	✓	✓
SNTP	Клиент	✓	✓	✓	✓	✓	✓
HTTP	Сервер	✓	✓	✓	✓	✓	✓
SNMP	Агент	✓	✓	✓	✓	✓	✓
EtherNet/IP	Сканер	✓	✓	✗	✓	Не применимо	✗
	Адаптер	✗	✓	✗	✓	Не применимо	✗

Таблица 49: Логика работы протокола x состояние центрального процессора (ЦП)

Примечания:

Символ ✓: Протокол остается активным и работает в нормальном режиме.

Символ ✗: Протокол отключен.

EtherCAT: Испытания, выполняемые с использованием MainTask в качестве задачи цикла шины EtherCAT. Если используется другая задача, то протокол останется активным при наличии точки останова в MainPrg. Более подробная информация о протоколе EtherCAT приведена в руководстве пользователя MasterTool IEC XE MT8500 – MU299609.

Ведомое устройство / сервер MODBUS Symbol (MODBUS Symbol Slave/Server): Чтобы протокол продолжал взаимодействовать, когда центральный процессор (ЦП) не находится в состоянии «Выполнение» (RUN) или после точки контрольного прерывания, необходимо отметить опцию «Поддерживать связь при останове центрального процессора (ЦП)» («Keep the communication running on CPU stop»).

### 5.4.2. Ведущее устройство MODBUS RTU (MODBUS RTU Master)

Данный протокол доступен для центральных процессоров (ЦП) серии Nexto в их последовательных каналах. При выборе данной опции в MasterTool IEC XE центральный процессор (ЦП) становится ведущим устройством связи MODBUS, что позволяет ему в режиме исполнения (режим «Выполнение» (*Run*)) обращаться к другим устройствам с тем же протоколом.

Для данного протокола существует два режима конфигурирования. В одном из них используется прямое представление (%Q), при котором переменные определяются по их адресу. В другом, называемом «Символьное преобразование» (*Symbolic Mapping*), переменные определяются по имени.

Независимо от режима конфигурирования, шаги по вставке экземпляра протокола и настройке последовательного интерфейса одинаковы. Процедура вставки экземпляра протокола подробно описана в руководстве пользователя MasterTool IEC XE – MU299609 или в разделе «Вставка экземпляра протокола». Остальные шаги по конфигурированию описаны ниже для каждого режима.

- Добавьте экземпляр протокола ведущего устройства MODBUS RTU (MODBUS RTU Master) в последовательный канал COM 1 или COM 2 (или в оба, если речь идет о двух сетях связи). Для выполнения данной процедуры см. раздел «Вставка экземпляра протокола».
- Настроить последовательный интерфейс, выбрав скорость передачи, логика работы сигналов сигналов RTS/CTS, четность, стоповые биты канала и другие конфигурации двойным щелчком мыши на последовательном канале COM 1 или COM 2. См. раздел «Конфигурация последовательных интерфейсов».

#### 5.4.2.1. Конфигурирование протокола ведущего устройства MODBUS с помощью символьного преобразования

Для конфигурации данного протокола с использованием символьного преобразования необходимо выполнить следующие действия:

- Сконфигурируйте общие параметры протокола ведущего устройства MODBUS (MODBUS Master), такие как: время задержки передачи и минимальный межкадровый интервал (интерфрейм), как показано на рисунке 47.
- Добавьте и сконфигурируйте устройства на вкладке «Общие параметры» (General Parameters), определив адрес ведомого устройства, блокировку по превышению лимита времени связи и количество повторных попыток связи, как показано на рисунке 48.
- Добавьте и сконфигурируйте преобразования протокола MODBUS на вкладке «Преобразования» (Mappings), как показано на рисунке 49, указав имя переменной, тип данных и начальный адрес данных; размер и диапазон данных заполняются автоматически.
- Добавьте и сконфигурируйте запросы MODBUS, как показано на рисунке 50, указав функцию, время сканирования запроса, начальный адрес (считывание/запись), размер данных (считывание/запись), а также сформируйте диагностические переменные и отключите запрос с помощью кнопок в нижней части окна.

##### 5.4.2.1.1. Общие параметры протокола ведущего устройства MODBUS - Конфигурация символьного преобразования

Общие параметры, представленные на начальном экране протокола MODBUS (рисунок ниже), определяются следующим образом:

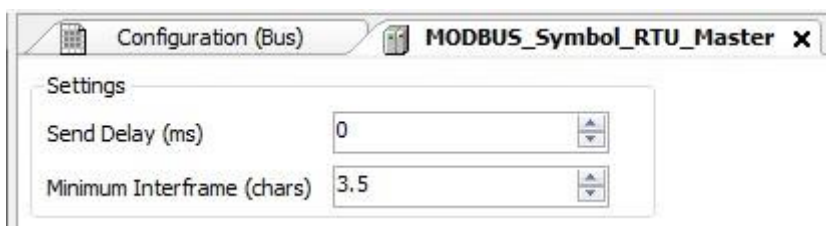


Рисунок 47: Экран конфигурации ведущего устройства MODBUS RTU

## 5. КОНФИГУРАЦИЯ

Конфигурация	Описание	По умолчанию	Опции
Задержка отправки (мс)	Задержка при передаче отклика	0	от 0 до 65535
Минимальный межкадровый интервал (интерфрейм) (символов)	Минимальное время паузы между различными фреймами (кадрами).	3.5	от 3,5 до 100,0

Таблица 50: Общие конфигурации ведущего устройства MODBUS RTU

Примечания:

**Задержка отправки:** Ответ на протокол MODBUS в некоторых моментах может вызвать проблемы, как, например, в полудуплексном интерфейсе RS-485 или других. Иногда возникает задержка между временем отклика ведомого устройства и паузой физической линии (задержка ведомого устройства на установку RTS в ноль и перевод RS-485 в высокоимпедансное состояние). Для решения данной проблемы ведомое устройство может выждать определенное в данном поле время перед отправкой нового запроса. В противном случае первые байты, переданные ведущим устройством, могут быть потеряны.

**Минимальный межкадровый интервал (интерфрейм):** Стандарт MODBUS определяет это время как 3,5 символа, но данный параметр настраивается для того, чтобы обслуживать устройства, которые не следуют стандарту.

**Диагностика и команды протокола MODBUS, сконфигурированные либо по символьному преобразованию, либо по прямому представлению, хранятся в переменных `T_DIAG_MODBUS_RTU_MASTER_1`. При преобразовании прямого представления они также находятся в 4 байтах и 8 словах, которые описаны в таблице, приведенной ниже (где «n» - сконфигурированное значение в поле «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area)):**

Переменная прямого представления	Диагностическая переменная <code>T_DIAG_MODBUS_RTU_MASTER_1.*</code>	Размер	Описание
Диагностические биты:			
<code>%QX(n).0</code>	<code>tDiag. bRunning</code>	BIT	Ведущее устройство работает
<code>%QX(n).1</code>	<code>tDiag. bNotRunning</code>	BIT	Ведущее устройство не работает (см. бит: <code>bInterruptedByCommand</code> ).
<code>%QX(n).2</code>	<code>tDiag. bInterruptedByCommand</code>	BIT	Бит <code>bNotRunning</code> был включен, поскольку работа ведущего устройства была прервана пользователем с помощью битов команд
<code>%QX(n).3</code>	<code>tDiag. bConfigFailure</code>	BIT	Прекращение диагностики.
<code>%QX(n).4</code>	<code>tDiag. bRXFailure</code>	BIT	Прекращение диагностики.
<code>%QX(n).5</code>	<code>tDiag. bTXFailure</code>	BIT	Прекращение диагностики.
<code>%QX(n).6</code>	<code>tDiag. bModuleFailure</code>	BIT	Указывает на наличие сбоя в модуле или его отсутствие.
<code>%QX(n).7</code>	<code>tDiag. bDiag_7_reserved</code>	BIT	Зарезервировано

5. КОНФИГУРАЦИЯ

Коды ошибок:

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS _RTU_MASTER_1.*	Размер	Описание
%QB(n+1)	eErrorCode	SERIAL_STATUS (BYTE)	<p>0: ошибок нет</p> <p>1: недопустимый последовательный порт</p> <p>2: недопустимый режим последовательного порта</p> <p>3: недопустимая скорость передачи данных в бодах</p> <p>4: недопустимые биты данных</p> <p>5: недопустимая четность</p> <p>6: недопустимые стоповые биты</p> <p>7: недопустимый параметр сигнала модема</p> <p>8: недействительный параметр порогового значения универсального асинхронного приемопередатчика (УАПП) RX (UART RX Threshold)</p> <p>9: неверный параметр блокировки по превышению лимита времени</p> <p>10: последовательный порт занят</p> <p>11: аппаратная ошибка универсального асинхронного приемопередатчика (УАПП)</p> <p>12: аппаратная ошибка удаленного оборудования</p> <p>20: недопустимый размер буфера передачи</p> <p>21: недопустимый метод сигнального модема</p> <p>22: блокировка по превышению лимита времени CTS = true («истина»)</p> <p>23: блокировка по превышению лимита времени CTS = false («ложь»)</p> <p>24: ошибка блокировки по превышению лимита времени передачи</p> <p>30: недопустимый размер буфера приема</p> <p>31: ошибка блокировки по превышению лимита времени приема</p> <p>32: управление потоком настроено иначе, чем вручную</p> <p>33: недопустимое управление потоком для сконфигурированного последовательного порта</p>

## 5. КОНФИГУРАЦИЯ

			<p>34: прием данных запрещен в нормальном режиме</p> <p>35: прием данных запрещен в расширенном режиме</p> <p>36: прерывание DCD не разрешено</p> <p>37: прерывание CTS не разрешено</p> <p>38: прерывание DSR не разрешено</p> <p>39: последовательный порт не сконфигурирован</p> <p>50: внутренняя ошибка в последовательном порту</p>
Биты команд, автоматически инициализируемые:			
%QX(n+2).0	tCommand. bStop	BIT	Останов ведущего устройства.
%QX(n+2).1	tCommand. bRestart	BIT	Перезапуск ведущего устройства.
%QX(n+2).2	tCommand. bResetCounter	BIT	Статистика диагностики перезапуска (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 представления	Диагностическая переменная T_DIAG_MODBUS _RTU_MASTER_1.*	Размер	Описание
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Зарезервировано
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано
Статистика коммуникаций:			
%QW(n+4)	tStat. wTXRequests	WORD	Счетчик запросов, переданных ведущим устройством (от 0 до 65535).
%QW(n+6)	tStat. wRXNormalResponses	WORD	Счетчик нормальных откликов, полученных ведущим устройством (от 0 до 65535).
%QW(n+8)	tStat. wRXExceptionResponses	WORD	Счетчик откликов с кодами исключений, полученных ведущим устройством (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalResponses	WORD	Счетчик недействительных откликов, полученных ведущим устройством - неправильный синтаксис, недостаточное количество принятых

## 5. КОНФИГУРАЦИЯ

			байт, неправильный CRC - (от 0 до 65535).
%QW(n+12)	tStat. wRXOverrunErrors	WORD	Счетчик ошибок превышения во время приема - UART FIFO (считывание универсального асинхронного приемопередатчика (УАПП) в порядке очереди) или линия RX - (от 0 до 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	WORD	Счетчик откликов с конструктивными недоработками, ошибками четности или сбоем во время приема (от 0 до 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	WORD	Счетчик ошибки блокировки по превышению лимита времени CTS, использующий синхронизацию интерфейса и периферийного оборудования RTS/CTS, во время передачи (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Зарезервировано

Таблица 51: Диагностика ведущего устройства MODBUS RTU

Примечание:

Счетчики: Все счетчики диагностики ведущего устройства MODBUS RTU (MODBUS RTU Master) возвращаются к нулю при превышении предельного значения 65535.

### 5.4.2.1.2. Конфигурация устройств - конфигурация символьного преобразования

Конфигурация устройств, представленная на приведенном ниже рисунке, соответствует следующим параметрам:

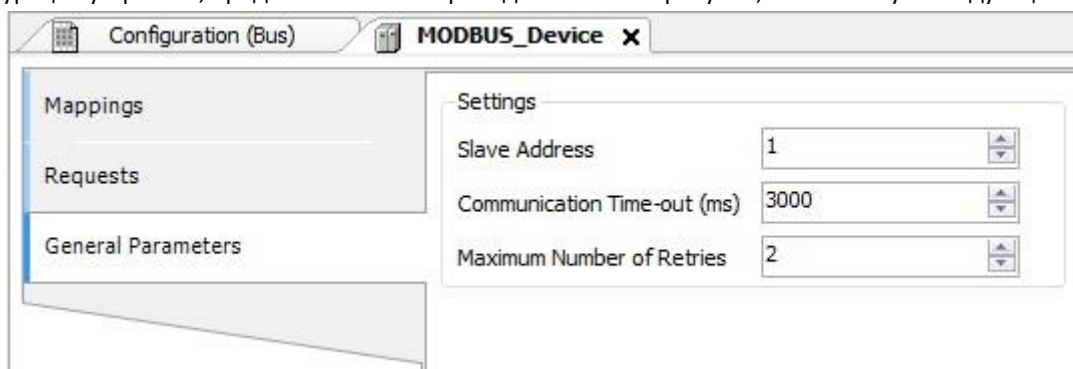


Рисунок 48: Настройки общих параметров устройства

Конфигурация	Описание	По умолчанию	Опции
Адрес ведомого устройства	Адреса ведомых устройств MODBUS	1	от 0 до 255
Блокировка по превышению лимита времени связи (мс)	Определяет блокировку по превышению лимита времени на уровне приложения	3000	от 10 до 65535

## 5. КОНФИГУРАЦИЯ

Максимальное количество повторных попыток	Определяет количество повторных попыток, прежде чем сообщить об ошибке связи	2	от 0 до 9
---	--	---	-----------

Таблица 52: Конфигурации устройства

Примечания:

Адрес ведомого устройства: Согласно стандарту MODBUS, допустимые адреса ведомых устройств составляют от 0 до 247, при этом адреса от 248 до 255 являются резервными. Когда ведущее устройство посылает команду записи с адресом, сконфигурированным как ноль, он осуществляет широковещательный запрос в сети.

Блокировка по превышению лимита времени связи: Блокировка по превышению лимита времени связи представляет собой время, в течение которого ведущее устройство ожидает ответа от ведомого устройства на запрос. Для ведущего устройства MODBUS RTU необходимо учитывать как минимум следующие системные переменные: время передачи фрейма (кадра) ведомым устройством (в зависимости от скорости передачи данных в бодах), время обработки запроса ведомым устройством и задержку отправки ответа, если она задана в ведомом устройстве. Рекомендуется, чтобы блокировка по превышению лимита времени была равна или превышала время передачи фрейма (кадра) плюс задержка отправки ответа и удвоенное время обработки запроса. Дополнительную информацию см. в разделе «Производительность связи» ([Communication Performance](#)).

Максимальное количество повторных попыток: Устанавливает количество повторных попыток перед отправкой сообщения об ошибке связи. Например, если ведомое устройство не отвечает на запрос, а ведущее устройство настроено на отправку трех повторных запросов, то при выполнении этих трех повторных запросов номер счетчика ошибок увеличивается на одну единицу. После увеличения ошибки связи при попытке перезапуска процесса и при повторном достижении числа повторных попыток новая ошибка увеличит значение счетчика.

### 5.4.2.1.3. Конфигурация преобразований – Настройки символьного преобразования

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

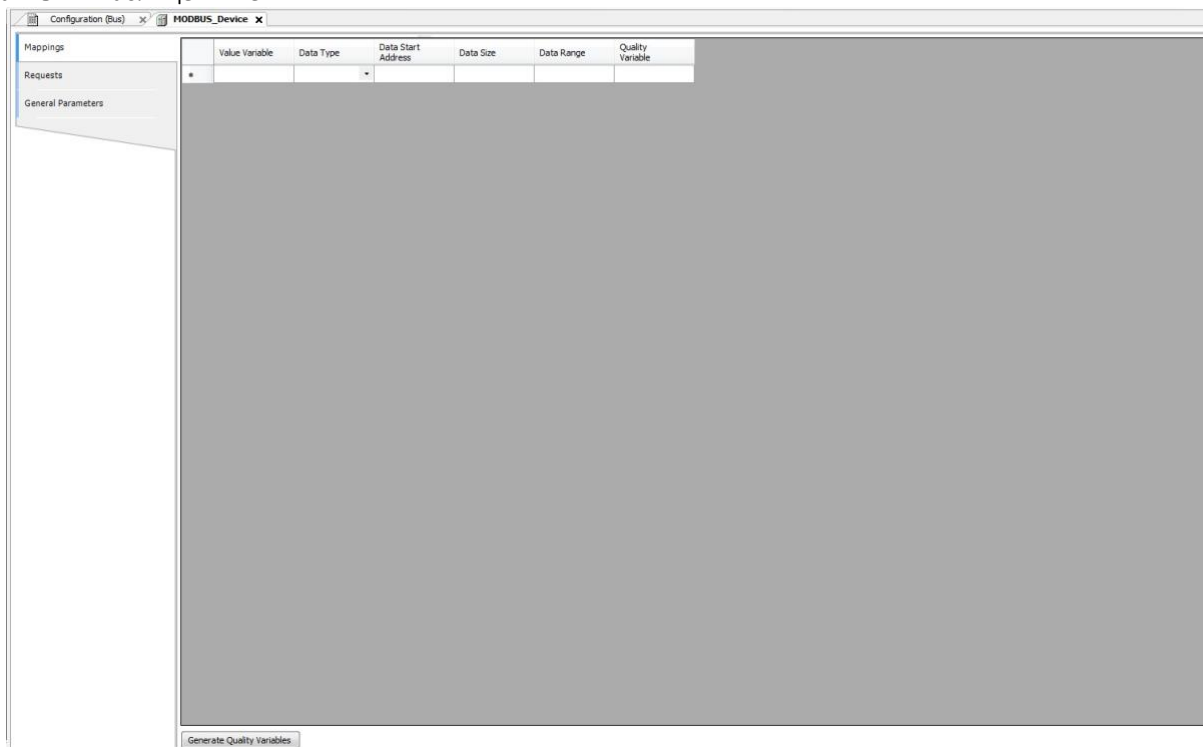


Рисунок 49: Экран преобразований данных MODBUS

## 5. КОНФИГУРАЦИЯ

Конфигурация	Описание	По умолчанию	Опции
Переменная значения	Имя символической переменной	-	Имя переменной, объявленной в программе или списке глобальных переменных (GVL)
Тип данных	Тип данных MODBUS	-	Регистр флагов - Запись (1 бит) Регистр флагов - Чтение (1 бит) Регистр временного хранения - Запись (16 бит) Регистр временного хранения - Чтение (16 бит) Регистр временного хранения - маска «И» (16 бит) Регистр временного хранения - маска «ИЛИ» (16 бит) Регистр входа (16 бит) Состояние входа (1 бит)
Начальный адрес данных	Начальный адрес данных MODBUS	-	от 1 до 65536
Размер данных	Размер данных MODBUS	-	от 1 до 65536
Диапазон данных	Диапазон адресов сконфигурированных данных	-	-

Таблица 53: Настройки преобразований протокола MODBUS

Примечания:

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

Тип данных: данное поле используется для указания типа данных, используемого в отношении MODBUS.

Тип данных	Размер [биты]	Описание
Регистр флагов - Запись	1	Запись цифрового выхода.
Регистр флагов - Чтение	1	Считывание цифрового выхода.
Регистр временного хранения - Запись	16	Запись аналогового выхода.
Регистр временного хранения - Чтение	16	Считывание аналогового выхода.
Регистр временного хранения - маска «И»	16	Аналоговый выход, который может быть считан или записан с помощью маски «И».
Регистр временного хранения - маска «ИЛИ»	16	Аналоговый выход, который может быть считан или записан с помощью маски «ИЛИ».
Регистр входа	16	Аналоговый вход, который может быть только считан.
Состояние входа	1	Цифровой вход, который может быть только считан.

Таблица 54: Типы данных, поддерживаемые в MODBUS



## 5. КОНФИГУРАЦИЯ

Начальный адрес данных: Начальный адрес данных в преобразовании MODBUS.

Размер данных: Значение size («размер») определяет максимальный объем данных, к которому может обратиться интерфейс MODBUS, начиная с начального адреса. Таким образом, для считывания непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном интерфейсе. Данное поле зависит от настроенного типа данных MODBUS.

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

### 5.4.2.1.4. Конфигурация запросов – Настройки символьного преобразования

Конфигурация запросов MODBUS, представленная на приведенном ниже рисунке, соответствует параметрам, описанным в таблице:

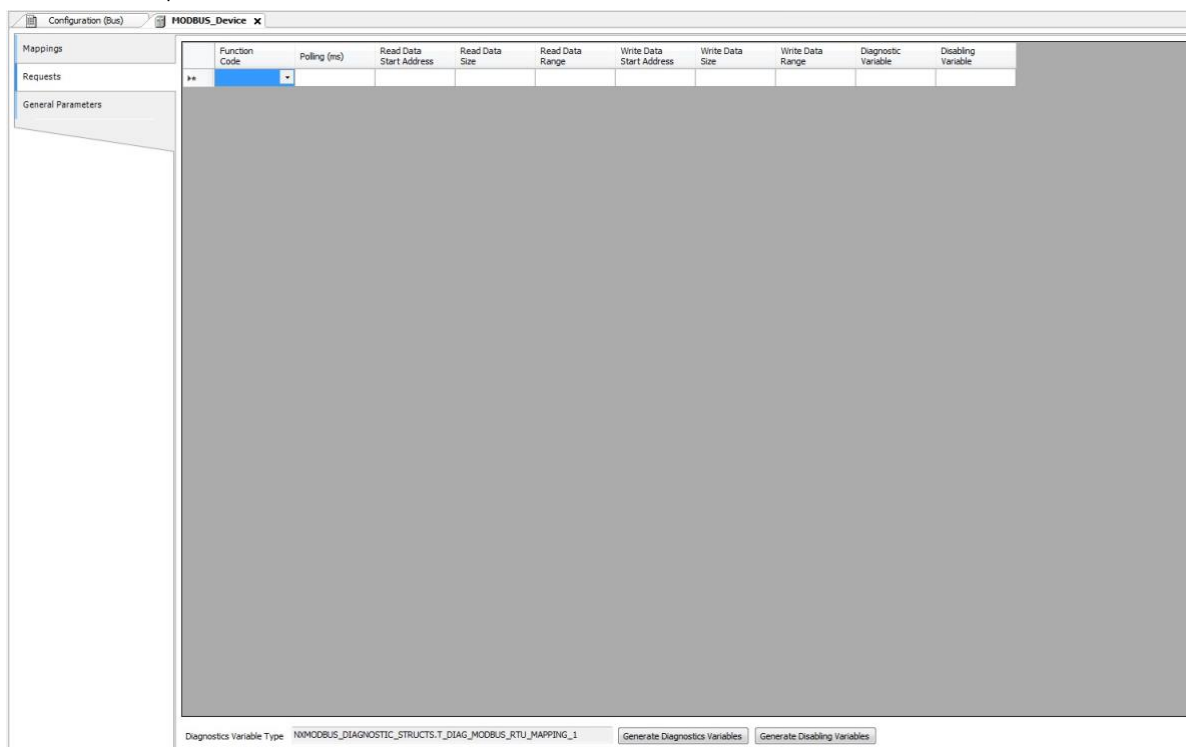


Рисунок 50: Экран запросов данных ведущего устройства MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Код функции	Тип функции MODBUS	-	01 - Считывание регистров флагов 02 - Считывание состояния входа 03 - Считывание регистров временного хранения 04 - Считывание регистров входов 05 - Запись регистра флагов 06 - Запись регистра временного хранения 15 - Запись нескольких регистров флагов 16 - Запись нескольких регистров временного хранения 22 - Маска записи регистра временного хранения 23 - Чтение/запись нескольких регистров временного хранения

## 5. КОНФИГУРАЦИЯ

Упорядоченный опрос (мс)	Период связи (мс)	100	от 0 до 3600000
Считывание начального адреса данных	Начальный адрес данных для считывания MODBUS	-	от 1 до 65536
Считывание размера данных	Размер данных для считывания MODBUS	-	В зависимости от используемой функции
Считывание диапазона данных	Диапазон адреса данных для считывания MODBUS	-	От 0 до 2147483646
Запись начального адреса данных	Начальный адрес для записи данных MODBUS	-	От 1 до 65536
Запись размера данных	Размер данных для записи MODBUS	-	В зависимости от используемой функции
Запись диапазона данных	Диапазон адреса данных для записи MODBUS	-	От 0 до 2147483647
Конфигурация	Описание	Значение по умолчанию	Опции
Диагностическая переменная	Наименование диагностической переменной	-	Имя переменной, объявленной в программе или списке глобальных переменных (GVL)
Переменная отключения	Переменная, используемая для отключения отношения MODBUS	-	Поле для символической переменной, используемой для индивидуального отключения сконфигурированных запросов MODBUS. Данная переменная должна относиться к типу BOOL. Переменная может быть простым элементом или элементом массива и может находиться в структурах.

Таблица 55: Конфигурация отношений протокола MODBUS

Примечания:

Уставка: количество заводских уставок параметров по умолчанию и значения для столбца «Опции» (Options) могут отличаться в зависимости от типа данных и функции MODBUS (FC).

Код функции: Доступны следующие функции MODBUS (FC):

Тип функции	Код		Описание
	DEC	HEX	
Доступ к переменным	1	0x01	Считывание регистров флагов (FC 01)
	2	0x02	Считывание состояния входа (FC 02)
	3	0x03	Считывание регистров временного хранения (FC 03)
	4	0x04	Считывание регистров входов (FC 04)
	5	0x05	Запись одного регистра флагов (FC 05)
	6	0x06	Запись одного регистра временного хранения (FC 06)
	15	0x0F	Запись нескольких регистров флагов (FC 15)
	16	0x10	Запись нескольких регистров временного хранения (FC 16)
	22	0x16	Маска записи регистра временного хранения (FC 22)

## 5. КОНФИГУРАЦИЯ

	23	0x17	Считывание/Запись нескольких регистров временного хранения (FC 23)
--	----	------	--

Таблица 56: Функции MODBUS, поддерживаемые центральными процессорами (ЦП) Nexto

Упорядоченный опрос: данный параметр указывает на то, как часто должен выполняться обмен данными, установленный для данного запроса. По окончании обмена данными, время ожидания будет равно значению, заданному в поле «Упорядоченный опрос» (polling), после чего будет выполнен новый обмен данными.

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

Размер считываемых данных: минимальное значение размера считываемых данных равно 1, в то время как максимальное зависит от используемой функции MODBUS (FC), как показано ниже:

- Считывание регистров флагов (FC 01): 2000
- Считывание состояния входа (FC 02): 2000
- Считывание регистров временного хранения (FC 03): 125
- Считывание регистров входов (FC 04): 125
- Считывание/запись нескольких регистров временного хранения (FC 23): 121

Диапазон считываемых данных: данное поле показывает диапазон считываемых данных MODBUS, сконфигурированный для каждого запроса. Начальный адрес вместе с размером считываемых данных приводит к диапазону считываемых данных для каждого запроса.

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

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

- Запись одного регистра флагов (FC 05): 1
- Запись одного регистра (FC 06): 1
- Запись нескольких регистров флагов (FC 15): 1968
- Запись нескольких регистров (FC 16): 123
- Маскирование регистра записи (FC 22): 1
- Считывание/запись нескольких регистров (FC 23): 121

Диапазон данных записи: данное поле показывает диапазон данных записи MODBUS, сконфигурированный для каждого запроса. Начальный адрес вместе с размером считываемых данных приводит к диапазону данных записи для каждого запроса.

Диагностическая переменная: Диагностика запросов MODBUS, сконфигурированная по символьному преобразованию или по прямому представлению, хранится в переменных типа *T\_DIAG\_MODBUS\_RTU\_MAPPING\_1* для ведущих устройств и *T\_DIAG\_MODBUS\_ETH\_CLIENT\_1* для устройств клиента, а преобразование по прямому представлению - в 4-байтовых и 2-словных, которые описаны в таблице 57 (n - значение, заданное в поле «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area)).

Переменная прямого представления	Диагностическая переменная типа <i>T_DIAG_MODBUS_RTU_MAPPING_1.*</i>	Размер	Описание
Биты состояния связи:			
%QX(n).0	byStatus. bCommIdle	BIT	Неактивная связь (ожидание выполнения).
%QX(n).1	byStatus. bCommExecuting	BIT	Активная связь.

5. КОНФИГУРАЦИЯ

%QX(n).2	byStatus. bCommPostponed	BIT	Установление связи откладывается, поскольку достигнуто максимальное количество одновременных запросов. Отложенное установление связи будет выполнено в той же последовательности, в которой они были необходимы, чтобы избежать неопределенности. Время, проведенное в этом состоянии, не учитывается при определении блокировки по превышению лимита времени. Биты bCommIdle и bCommExecuting ложны (false), если бит bCommPostponed истинен (true).
%QX(n).3	byStatus. bCommDisabled	BIT	Связь отключена. В этом состоянии бит bCommIdle перезапускается.
%QX(n).4	byStatus. bCommOk	BIT	Связь, завершенная ранее, была успешной.
%QX(n).5	byStatus. bCommError	BIT	В ранее завершенной связи возникла ошибка. Проверьте код ошибки.
%QX(n).6	byStatus. bCommAborted	BIT	Не используется в ведущем устройстве MODBUS RTU (MODBUS RTU Master).
%QX(n).7	byStatus. bDiag_7_reserved	BIT	Зарезервировано
Код последней ошибки (включается, когда bCommError = TRUE («ИСТИНА»)):			
%QB(n+1)	eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Информирует о возможной причине последней ошибки в части преобразования MODBUS. Более подробная информация приведена в таблице 80.
Последний код исключения, полученный ведущим устройством:			
%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)*
Статистика коммуникаций:			
Direct Representation Variable Переменная прямого представления	Диагностическая переменная типа T_DIAG_MODBUS_RTU_MAPPING_1.*	Размер	Описание
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.

## 5. КОНФИГУРАЦИЯ

%QW(n+4)	wCommCounter	WORD	Счетчик связи завершен (с ошибками или без). Пользователь может проверить, когда связь была завершена, путем проверки изменения данного счетчика. 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 57: Диагностика отношений протокола MODBUS

### Примечания:

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

**Переменная отключения:** переменная булевого типа, используемая для индивидуального отключения запросов MODBUS, сконфигурированных на вкладке запросов с помощью кнопки в нижней части окна. Запрос отключен, если переменная, соответствующая запросу, равна 1, в противном случае запрос включен.

**Код последней ошибки:** Ниже приведены коды возможных ситуаций, приводящих к возникновению ошибки при обмене данными по протоколу MODBUS:

Код	Счетное значение	Описание
1	ERR_EXCEPTION	Ответ находится в коде исключения (см. eLastExceptionCode = Exception Code («Код исключения»)).
2	ERR_CRC	Ответ с недействительным CRC.
3	ERR_ADDRESS	Адрес MODBUS не найден. Адрес, ответивший на запрос, отличается от ожидаемого.
4	ERR_FUNCTION	Неверный код функции. Код функции ответа отличается от ожидаемого.
5	ERR_FRAME_DATA_COUNT	Объем данных в ответе оказался иным, чем ожидалось.
7	ERR_NOT_ECHO	Ответ не является эхоотображением (воспроизведением) запроса (FC 05 и 06).
8	ERR_REFERENCE_NUMBER	Неверный номер ссылки (FC 15 и 16).
9	ERR_INVALID_FRAME_SIZE	Ответ оказался короче, чем ожидалось.
20	ERR_CONNECTION	Ошибка при установлении соединения.
21	ERR_SEND	Ошибка на этапе передачи данных.
22	ERR_RECEIVE	Ошибка на этапе приема данных.
40	ERR_CONNECTION_TIMEOUT	Блокировка по превышению лимита времени на уровне приложения во время соединения.
41	ERR_SEND_TIMEOUT	Блокировка по превышению лимита времени на уровне приложения при передаче данных.
42	ERR_RECEIVE_TIMEOUT	Блокировка по превышению лимита времени на уровне приложения во время ожидания отклика.

## 5. КОНФИГУРАЦИЯ

43	ERR_CTS_OFF_TIMEOUT	Блокировка по превышению лимита времени при ожидании CTS = false («ЛОЖЬ») при передаче.
44	ERR_CTS_ON_TIMEOUT	Блокировка по превышению лимита времени при ожидании CTS =TRUE («ИСТИНА») при передаче.
128	NO_ERROR	С момента запуска ошибок нет.

Таблица 58: Коды ошибок отношений протокола MODBUS

**ВНИМАНИЕ:**

В отличие от других прикладных задач, при достижении метки очистки в MainTask выполнение задачи экземпляра ведущего устройства MODBUS RTU и любой другой задачи протокола MODBUS будет прекращено в тот момент, когда таковой будет предпринята попытка производства записи в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока не выполняется MainTask.

### 5.4.2.2. Конфигурация протокола ведущего устройства MODBUS (MODBUS Master) для прямого представления (%Q)

Для конфигурирования данного протокола с использованием прямого представления (%Q) необходимо выполнить следующие действия:

- Конфигурирование общих параметров протокола MODBUS, таких как: время связи и переменные прямого представления (%Q) для получения диагностики.
- Добавление и конфигурирование устройств путем задания адреса, переменных прямого представления (%Q) для отключения отношений, блокировок по превышению лимита времени связи и т.д.
- Добавление и конфигурирование отношений MODBUS с указанием типа данных и функции MODBUS, блокировок по превышению лимита времени, переменных прямого представления (%Q) для получения диагностики отношения и других для приема/записи данных, количества передаваемых данных и упорядоченного опроса отношений.

Описание каждой конфигурации приведено ниже в данном разделе.

#### 5.4.2.2.1. Общие параметры протокола ведущего устройства MODBUS (MODBUS Master) - настройка посредством прямого представления (%Q)

Общие параметры, расположенные на главном экране конфигурации протокола MODBUS (рисунок, приведенный ниже), определяются следующим образом:

## 5. КОНФИГУРАЦИЯ

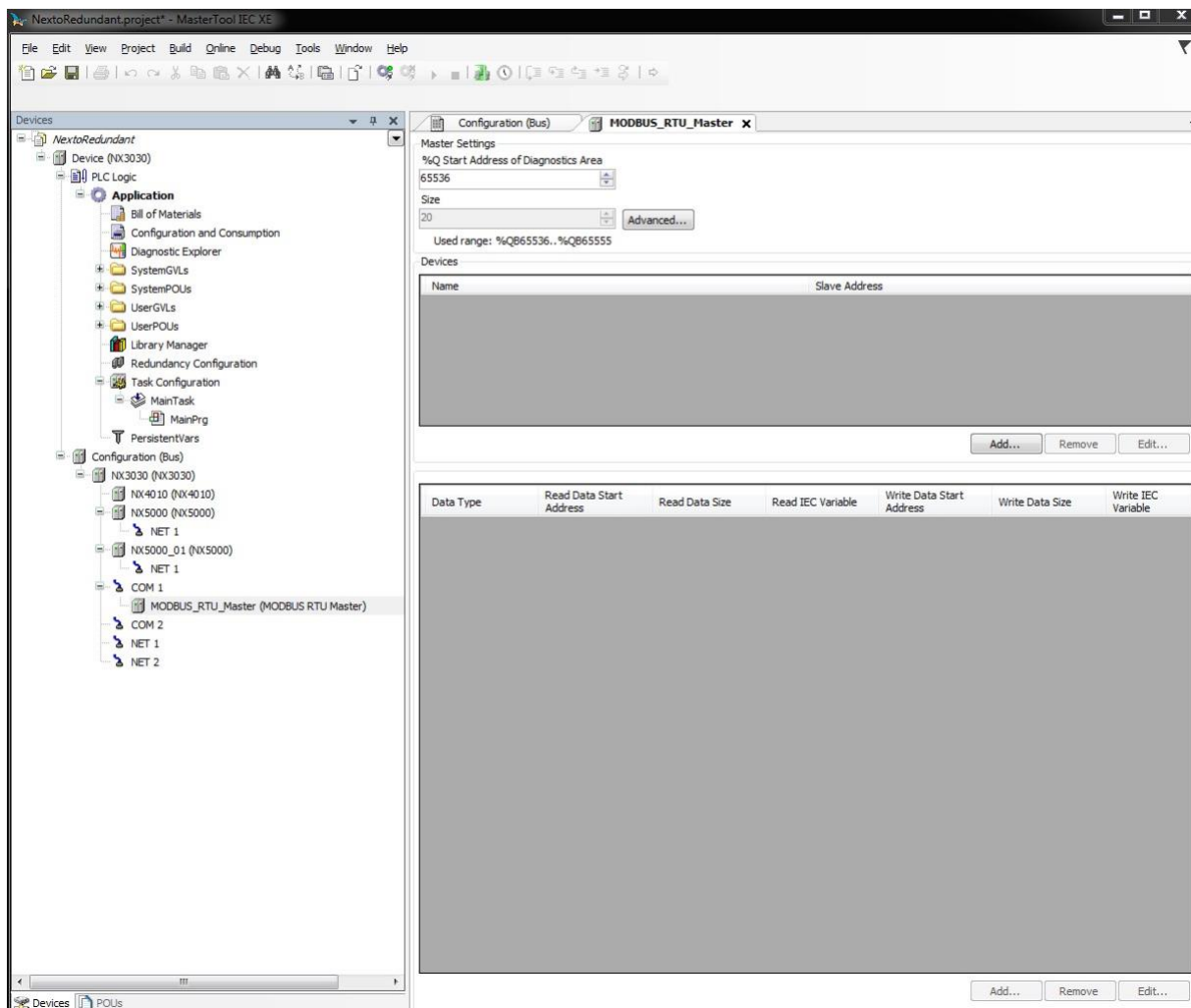


Рисунок 51: Экран настройки ведущего устройства MODBUS RTU

Переменные прямого представления (%Q) для диагностики протокола:

Конфигурация	Описание	Значение по умолчанию	Опции
Начальный адрес %Q области диагностики	Начальный адрес диагностических переменных	-	от 0 до 2147483628
Размер	Размер области диагностики	20	Выключено для редактирования

Таблица 59: Конфигурация ведущего устройства MODBUS RTU

Примечания:

Начальный адрес диагностики в %Q: это поле ограничено размером адресуемой памяти переменных выходов (%Q) каждого центрального процессора (ЦП), который можно найти в разделе «Память».

Значение по умолчанию: заводское значение по умолчанию для поля «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area) не может быть установлено, поскольку создание экземпляра протокола может быть проведено в любой момент разработки приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), которое еще не используется.

Команды диагностики и протокола MODBUS описаны в таблице 51.

## 5. КОНФИГУРАЦИЯ

Время связи по протоколу ведущего устройства MODBUS Master (MODBUS Master), определяемое по кнопке «Дополнительно...» (*Advanced...*) на экране конфигурации, подразделяется на *задержку отправки (Send Delay)* и *минимальный межкадровый интервал интерфрейм (Minimum Interframe)*. С дополнительной информацией вы можете ознакомиться в разделе «Общие параметры протокола ведущего устройства MODBUS - Конфигурация символического преобразования».

### 5.4.2.2.2. Конфигурация устройств - Конфигурация для прямого представления (%Q)

Конфигурация устройств, представленная на приведенном ниже рисунке, включает в себя следующие параметры:

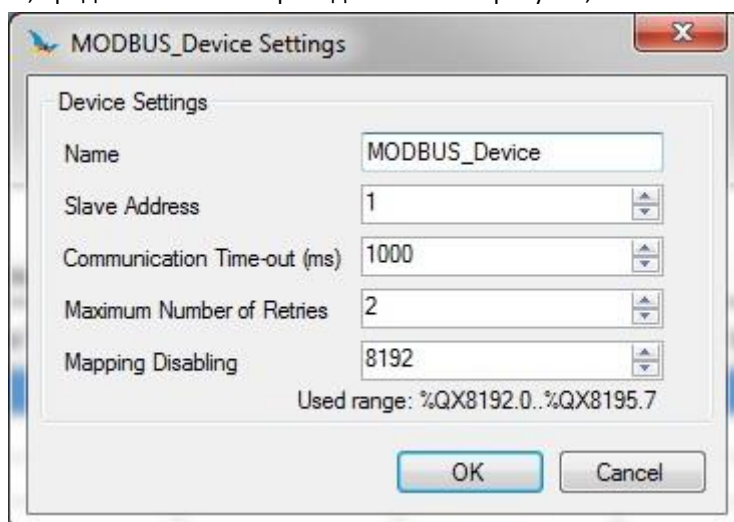


Рисунок 52: Конфигурация устройств

Конфигурация	Описание	Значение по умолчанию	Опции
Имя	Наименование экземпляра	MODBUS_Device	Идентификатор, в соответствии с МЭК 61131-3
Адрес ведомого устройства	Адрес ведомого устройства MODBUS	1	От 0 до 255
Блокировка по превышению лимита времени связи (мс)	Устанавливает блокировку по превышению лимита времени на уровне приложения	1000	От 10 до 65535
Конфигурация	Описание	Значение по умолчанию	Опции
Максимальное количество повторных попыток	Устанавливает количество повторных попыток перед сообщением об ошибке связи	2	От 0 до 9
Выключение преобразования	Начальный адрес, используемый для отключения отношения MODBUS	-	От 0 до 2147483644

Таблица 60: Конфигурация устройств – ведущее устройство MODBUS



---

## 5. КОНФИГУРАЦИЯ

Примечания:

Наименование экземпляра: данное поле является идентификатором устройства, который проверяется в соответствии с МЭК 61131-3 (IEC 61131-3), т.е. не допускает пробелов, специальных символов и начинается с числового знака. Оно ограничено 24 символами.

Выключение преобразования: состоит из 32 битов, используемых для отключения по отдельности 32 отношений MODBUS, сконфигурированных в пространстве «Преобразования устройств» (*Device Mappings*). Отношение отключено, если бит, соответствующий отношению, равен 1, в противном случае преобразование включено. Данное поле ограничено размером адресуемой памяти переменных выходов (%Q) каждого центрального процессора (ЦП), который можно найти в разделе «Память».

Значение по умолчанию: заводское значение по умолчанию в поле «Выключение преобразования» (*Mapping Disabling*) не может быть установлено, поскольку создание экземпляра протокола может быть проведено в любой момент разработки приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), которое еще не используется.

Более подробную информацию о параметрах «Адрес ведомого устройства» (*Slave Address*), «Блокировка по превышению лимита времени связи» (*Communication Time-out*) и «Максимальное количество повторных попыток» (*Maximut Number of Retries*) см. В примечаниях в разделе «Конфигурация устройств - Конфигурация символьного преобразования».

### 5.4.2.2.3. Конфигурация преобразований - Конфигурация для прямого представления (%Q)

Настройки отношений MODBUS, представленные на приведенных ниже рисунках, соответствуют параметрам, описанным в таблице:

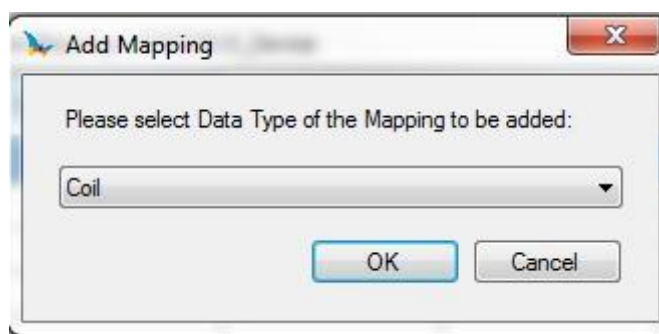


Рисунок 53: Тип данных протокола MODBUS

5. КОНФИГУРАЦИЯ

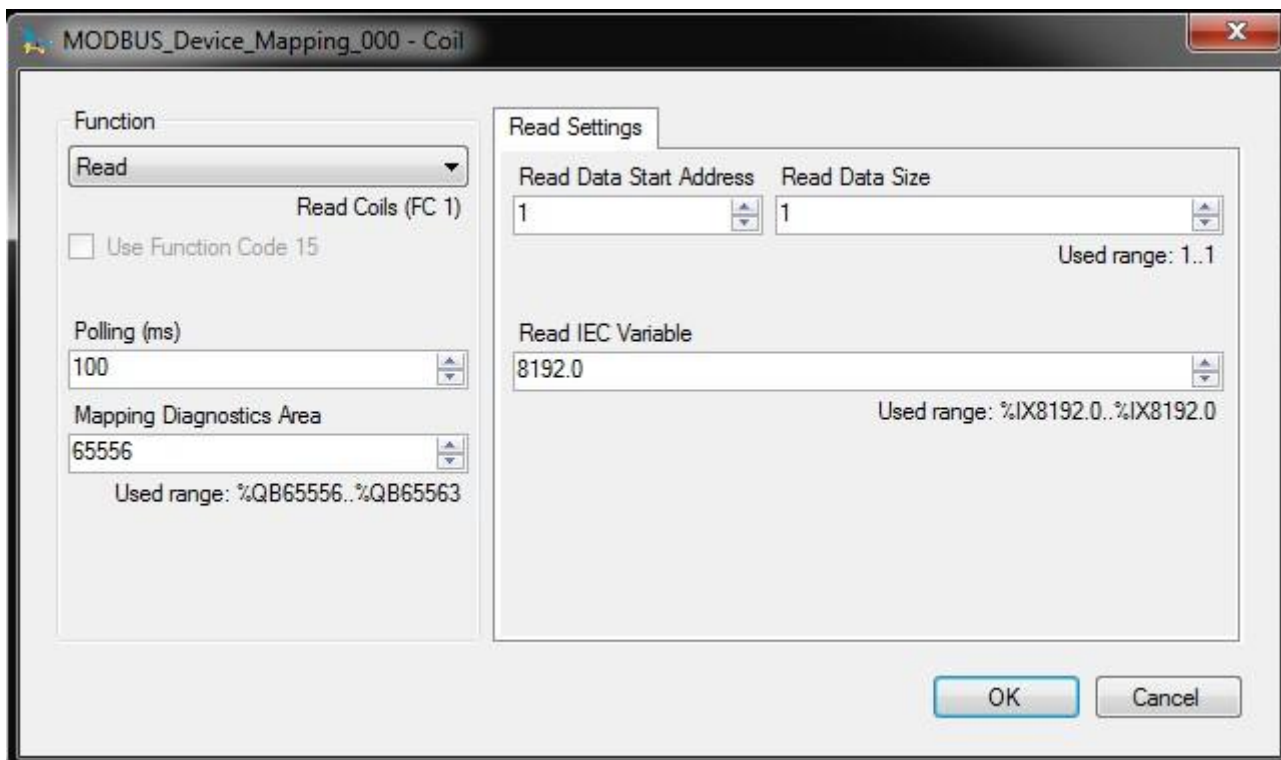


Рисунок 54: Функция протокола MODBUS

В приведенной ниже таблице количество заводских уставок по умолчанию и значения для столбца «Опции» (Options) могут отличаться в зависимости от типа данных и функции MODBUS (FC).

Конфигурация	Описание	По умолчанию	Опции
Функция	Тип функции MODBUS	Считывание	Считывание Запись Считывание/Запись Маска записи
Упорядоченный опрос (мс)	Период связи (мс)	100	от 0 до 3600000
Область диагностики преобразования	Начальный адрес диагностики связи MODBUS (%Q)	-	от 0 до 2147483640
Считывание начального адреса данных	Начальный адрес данных считывания MODBUS	1	от 1 до 65536
Считывание размера данных	Число считываемых данных MODBUS	-	В зависимости от используемой функции
Считывание переменной IEC	Начальный адрес считываемых переменных (%I)	-	от 0 до 2147483646
Запись начального адреса данных	Начальный адрес данных записи MODBUS	1	от 1 до 65536
Запись размера данных	Число данных записи MODBUS	-	В зависимости от используемой функции

## 5. КОНФИГУРАЦИЯ

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

Таблица 61: Преобразование устройства

Примечания:

Функция: доступные типы данных подробно описаны в таблице 80, а функции MODBUS (FC) – в таблице 78.

Упорядоченный опрос: данный параметр указывает на то, как часто должен выполняться обмен данными, установленный для данного отношения. По окончании обмена данными, время ожидания будет равно значению, заданному значению упорядоченного опроса, после чего как можно скорее будет выполнен новый обмен данными.

Область диагностики преобразования: данное поле ограничено размером адресуемой памяти выходных переменных (%Q) на центральном процессоре (ЦП), который можно найти в разделе «Память». Конфигурируемая диагностика отношений MODBUS описана в таблице 57.

Размер данных при считывании/записи: подробности о размере данных, поддерживаемых каждой функцией, описаны в примечаниях раздела «Конфигурация запросов – Настройки символьного преобразования».

**ВНИМАНИЕ:**

При обращении к памяти данных связи между устройствами с различным порядком следования байтов многобайтного объекта в памяти или при передаче (эндианизм) (Little-Endian (порядок байтов от младшего к старшему) и Big-Endian (порядок байтов от старшего к младшему)) может произойти инверсия данных при считывании/записи. В этом случае пользователь должен скорректировать данные в приложении.

Считывание переменной IEC: если тип данных MODBUS – *Coil* («Регистр флагов») либо *Input Status* («Состояние входов») (бит), то начальный адрес переменных считывания IEC будет иметь, например, формат %IX10.1. Если же тип данных MODBUS – *Holding Register* («Регистр временного хранения») или *Input Register* («Регистр входов») (16 бит), то начальный адрес переменных считывания IEC будет иметь формат %IW. Данное поле ограничено размером памяти адресуемых входных переменных (%I) в центральном процессоре (ЦП), который можно найти в разделе «Память».

Запись переменной IEC: если тип данных MODBUS – *Coil* («Регистр флагов»), то начальный адрес переменных записи IEC будет иметь, например, формат %QX10.1. Если же тип данных MODBUS – *Holding Register* («Регистр временного хранения») (16 бит), то начальный адрес записи переменных IEC будет иметь формат %QW. Данное поле ограничено размером адресуемой памяти выходных переменных (%Q) в центральном процессоре (ЦП), который можно найти в разделе «Память».

Запись маски: функция «Запись маски» (*Mask Write*) (FC 22) использует логику между уже записанным значением и двумя словами, которые конфигурируются в данном поле, используя %QW(0) для маски AND («логическая операция И») и %QW(2) для маски OR («логическая операция ИЛИ»); что позволяет пользователю обрабатывать слово. Данное поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого центрального процессора (ЦП), который можно найти в разделе «Память».

Значение по умолчанию: заводское значение по умолчанию для полей «Область диагностики преобразований» (*Mapping Diagnostics Area*), «Считывание переменной IEC» (*Read IEC Variable*), «Запись переменной IEC» (*Write IEC Variable*) и «Маскировка записи переменных IEC» (*Mask Write IEC Variables*) установить нельзя, так как создание отношения может быть выполнено в любой момент при разработке приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), которое пока не используется. Для полей «Размер считываемых данных/данных записи» (*Read/Write Data Size*) нельзя установить заводское значение, поскольку они будут изменяться в зависимости от выбранного типа данных MODBUS.

**ВНИМАНИЕ:**

В отличие от других задач приложения, при достижении метки при отладке MainTask выполнение задачи экземпляра ведущего устройства MODBUS RTU либо любой другой задачи протокола MODBUS будет прекращено в тот момент, когда она попытается произвести запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока MainTask не выполняется.

### 5.4.3. Ведомое устройство MODBUS RTU (MODBUS RTU Slave)

Данный протокол доступен для серии Nexto на ее последовательных каналах. При выборе данной опции в MasterTool IEC XE центральный процессор (ЦП) становится ведомым устройством связи MODBUS, что позволяет подключаться к ведущим устройствам MODBUS RTU. Данный протокол доступен только в режиме исполнения (режим «Выполнение» (Run)).

Существует два способа конфигурирования данного протокола. Первый – через прямое представление (%Q), при котором переменные определяются своим адресом. Второй – через символьное преобразование, при котором переменные определяются по имени.

Независимо от режима конфигурирования, шаги по вставке экземпляра протокола и конфигурированию последовательного интерфейса одинаковы. Процедура вставки экземпляра протокола подробно описана в руководстве пользователя MasterTool IEC XE – MU299609. Остальные шаги конфигурирования описаны ниже для каждого режима:

- Добавьте экземпляр протокола ведомого устройства MODBUS RTU (MODBUS RTU Slave) в последовательный канал COM 1 или COM 2 (или в оба, если речь идет о двух сетях связи). Для выполнения данной процедуры см. раздел «[Вставка экземпляра протокола](#)».
- Настроить последовательный интерфейс, выбрав скорость передачи данных, логику работы сигналов RTS/CTS, четность, канал стоповых битов и др. См. раздел «[Конфигурация последовательных интерфейсов](#)».

#### 5.4.3.1. Конфигурирование протокола ведомого устройства MODBUS (MODBUS Slave) с помощью символьного преобразования

Для настройки данного протокола с использованием символьного отображения необходимо выполнить следующие действия:

- Настройте общие параметры протокола MODBUS, такие как: адрес ведомого устройства и время связи (доступно при нажатии кнопки «*Расширенные конфигурации ведомого устройства*» (*Slave advanced configurations*)).
- Добавление и настройка связей MODBUS с указанием имени переменной, типа данных MODBUS и начального адреса данных.

Автоматически будут заполнены размер и диапазон данных в соответствии с объявленным типом переменной.

##### 5.4.3.1.1. Общие параметры протокола ведомого устройства MODBUS (MODBUS Slave) – Конфигурирование посредством символьного преобразования

Общие параметры, представленные на начальном экране протокола MODBUS (рисунок, приведенный ниже), определяются следующим образом.

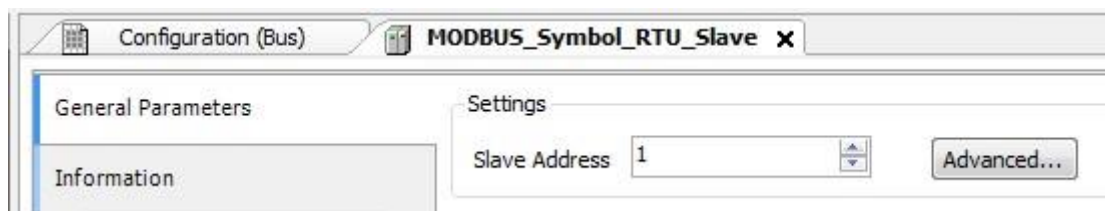


Рисунок 55: Экран конфигурации ведомого устройства MODBUS RTU

Конфигурация	Описание	По умолчанию	Опции
Адрес ведомого устройства	Адрес ведомого устройства MODBUS	1	от 1 до 255

Таблица 62: Конфигурации ведомого устройства

Время связи по протоколу ведомого устройства MODBUS, определяемое по кнопке «*Дополнительно...*» (*Advanced...*) на экране конфигурации, подразделяется на *цикл задач (Task Cycle)*, *задержку отправки (Send Delay)* и

## 5. КОНФИГУРАЦИЯ

минимальный межкадровый интервал интерфрейм (*Minimum Interframe*), как показано на приведенном ниже рисунке и в представленной ниже таблице.

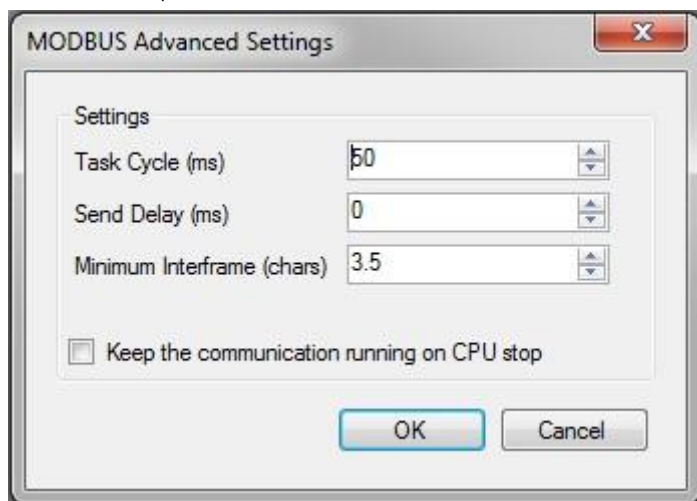


Рисунок 56: Расширенные конфигурации ведомого устройства Modbus

Конфигурация	Описание	По умолчанию	Опции
Цикл задач (мс)	Время выполнения экземпляра в рамках цикла без учета его собственного времени выполнения	50	От 20 до 100
Задержка отправки (мс)	Задержка на передачу ответа	0	От 0 до 65535
Минимальный межкадровый интервал (интерфрейм) (символов)	Минимальное время паузы между различными фреймами (кадрами).	3.5	От 3,5 до 100,0
Конфигурация	Описание	По умолчанию	Опции
Поддержка связи при останове центрального процессора (ЦП)	Разрешить работу ведомого устройства MODBUS Symbol Slave во время останова центрального процессора или после точки контрольного прерывания	Не отмечено	Отмечено или не отмечено

Таблица 63: Расширенные конфигурации ведомого устройства Modbus

### Примечания:

**Цикл задач:** пользователь должен быть осторожен при изменении данного параметра, поскольку он напрямую влияет на время отклика, объем данных для сканирования и, главным образом, на баланс ресурсов центрального процессора (ЦП) между коммуникациями и другими задачами.

**Задержка отправки:** Ответ на протокол MODBUS в некоторых моментах может вызвать проблемы, как, например, в полудуплексном интерфейсе RS-485 или других полудуплексных интерфейсах. Иногда возникает задержка между временем отклика ведомого устройства и паузой физической линии (задержка ведомого устройства на установку RTS в ноль и перевод RS-485 в высокоимпедансное состояние). Для решения данной проблемы ведущее устройство может

## 5. КОНФИГУРАЦИЯ

выждать определенное в данном поле время перед отправкой нового запроса. В противном случае первые байты, переданные ведущим устройством, могут быть потеряны.

Минимальный межкадровый интервал (интерфрейм): Стандарт MODBUS определяет это время как 3,5 символа, но данный параметр настраивается для того, чтобы обслуживать устройства, которые не следуют стандарту.

Диагностика и команды протокола ведомого устройства MODBUS (MODBUS Slave), сконфигурированные либо по символному преобразованию, либо по прямому представлению, хранятся в переменных *T\_DIAG\_MODBUS\_RTU\_SLAVE\_1*. При преобразовании прямого представления они также находятся в 4 байтах и 8 словах, которые описаны в приведенной ниже таблице (где «n» - сконфигурированное значение в поле «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area)):

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_SLAVE_1.*	Размер	Описание
Диагностические биты:			
%QX(n).0	tDiag. bRunning	BIT	Ведомое устройство находится в режиме выполнения.
%QX(n).1	tDiag. bNotRunning	BIT	Ведомое устройство не находится в процессе выполнения (см. бит: bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	Бит bNotRunning был включен, поскольку ведомое устройство было прервано пользователем с помощью задающих битов
%QX(n).3	tDiag. bConfigFailure	BIT	Прекращение диагностики.
%QX(n).4	tDiag. bRXFailure	BIT	Прекращение диагностики.
%QX(n).5	tDiag. bTXFailure	BIT	Прекращение диагностики.
%QX(n).6	tDiag. bModuleFailure	BIT	Прекращение диагностики.
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Зарезервировано.
Коды ошибок:			

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_SLAVE_1.*	Размер	Описание
%QB(n+1)	eErrorCode	SERIAL_STATUS (BYTE)	0: ошибок нет 1: недопустимый последовательный порт 2: недопустимый режим последовательного порта 3: недопустимая скорость передачи данных в бодах 4: недопустимые биты данных 5: недопустимая четность

5. КОНФИГУРАЦИЯ

			<p>6: недопустимые стоповые биты          7: недопустимый параметр сигнала модема          8: недействительный параметр порогового значения универсального асинхронного приемопередатчика (УАПП) RX (UART RX Threshold)          9: неверный параметр блокировки по превышению лимита времени          10: последовательный порт занят          11: аппаратная ошибка универсального асинхронного приемопередатчика (УАПП)          12: аппаратная ошибка удаленного оборудования          20: недопустимый размер буфера передачи          21: недопустимый метод сигнального модема          22: блокировка по превышению лимита времени CTS = true («истина»)          23: блокировка по превышению лимита времени CTS = false («ложь»)          24: ошибка блокировки по превышению лимита времени передачи          30: недопустимый размер буфера приема          31: ошибка блокировки по превышению лимита времени приема          32: управление потоком настроено иначе, чем вручную          33: недопустимое управление потоком для сконфигурированного последовательного порта          34: прием данных запрещен в нормальном режиме          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 сконфигурирован          50: внутренняя ошибка в последовательном порту</p>
<p>Биты команд, автоматически инициализируемые:</p>			

5. КОНФИГУРАЦИЯ

%QX(n+2).0	tCommand. bStop	BIT	Останов ведомого устройства.
%QX(n+2).1	tCommand. bRestart	BIT	Перезапуск ведомого устройства.
%QX(n+2).2	tCommand. bResetCounter	BIT	Статистика диагностики перезапуска (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 представления	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.
Статистика коммуникаций:			
%QW(n+4)	tStat. wRXRequests	WORD	Счетчик нормальных запросов, полученных ведомым устройством и получивших нормальный отклик. В случае широковещательной команды значение данного счетчика увеличивается, но не передается (от 0 до 65535).
%QW(n+6)	tStat. wTXExceptionResponses	WORD	Счетчик нормальных запросов, полученных ведомым устройством и отвеченных с кодом исключения. В случае широковещательной команды данный счетчик увеличивается, но не передается (от 0 до 65535). Коды исключений: 1: код функции (FC) является законным, но не поддерживается. 2: связь не найдена в этих данных MODBUS. 3: недопустимое значение для данного поля. 128: ведущее устройство/клиент не имеет права на запись или чтение. 129: связь MODBUS отключена.



## 5. КОНФИГУРАЦИЯ

%QW(n+8)	tStat. wRXFrames	WORD	Счетчик кадров, полученных ведомым устройством. Считается кадром то, что обрабатывается и за ним следует минимальная межкадровая пауза, другими словами, также вычисляется недействительное сообщение (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalRequests	WORD	Счетчик недействительных запросов. Это кадры, которые начинаются с адреса 0 (широковещательный) или с адреса ведомого устройства MODBUS, но не являются действительными запросами – неправильный синтаксис, меньшие кадры, неправильный CRC - (от 0 до 65535).
%QW(n+12)	tStat. wRXOverrunErrors	WORD	Счетчик кадров с ошибками превышения во время приема - UART FIFO (считывание универсального асинхронного приемопередатчика (УАПП) в порядке очереди) или линия RX - (от 0 до 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	WORD	Счетчик кадров с конструктивными недоработками, ошибками четности или сбоями во время приема (от 0 до 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	WORD	Счетчик ошибок блокировки по превышению лимита времени CTS, использующий синхронизацию интерфейса и периферийного оборудования RTS/CTS, во время передачи (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Зарезервировано.

Таблица 64: Диагностика ведомого устройства MODBUS RTU

Примечание:

Счетчики: Все счетчики диагностики ведомого устройства MODBUS RTU (MODBUS RTU Slave) возвращаются к нулю при превышении предельного значения 65535.

### 5.4.3.1.2. Конфигурация отношений - настройка символьного преобразования

Конфигурация отношений протокола MODBUS, показанная на рисунке, приведенном ниже, соответствует параметрам, описанным в таблице ниже:

Mappings						
	Value Variable	Data Type	Data Start Address	Absolute Data Start Address	Data Size	Data Range
*						

Рисунок 57: Экран преобразований данных протокола MODBUS

Конфигурация	Описание	По умолчанию	Опции
Переменная значения	Имя символической переменной	-	Имя переменной, объявленной в программе или списке глобальных переменных (GVL)
Тип данных	Тип данных MODBUS	-	Регистр флагов Состояние входа Регистр временного хранения Регистр входа
Начальный адрес данных	Начальный адрес данных MODBUS	-	От 1 до 65536
Абсолютный начальный адрес данных	Абсолютный начальный адрес данных MODBUS в соответствии с их типом	-	-
Размер данных	Размер данных MODBUS	-	От 1 до 65536
Диапазон данных	Сконфигурированный диапазон адресов данных	-	-

Таблица 65: Конфигурации преобразования протокола MODBUS

Примечания:

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

Тип данных: данное поле используется для указания типа данных, используемого в отношении MODBUS.

Тип данных	Размер [биты]	Описание
Регистр флагов	1	Цифровой выход, который может быть считан или записан.
Состояние входа	1	Цифровой вход (только чтение).
Регистр временного хранения	16	Аналоговый выход, который может быть считан или записан.
Регистр входа	16	Аналоговый вход (только чтение).

Таблица 66: Типы данных MODBUS, поддерживаемые центральными процессорами (ЦП) Nexto

Начальный адрес данных: Начальный адрес данных отношения MODBUS.

Размер данных: Значение размера данных определяет максимальный объем данных, к которому может получить доступ отношение MODBUS, начиная с начального адреса. Таким образом, для считывания непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном отношении. Данное поле изменяется в зависимости от настроенного типа данных MODBUS.

Диапазон данных: Данное поле показывает пользователю диапазон адресов памяти, используемый отношением MODBUS.

**ВНИМАНИЕ:**

В отличие от других прикладных задач, при достижении метки очистки в MainTask выполнение задачи экземпляра ведомого устройства MODBUS RTU и любой другой задачи MODBUS будет прекращено в тот момент, когда будет предпринята попытка производства записи в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока не выполняется MainTask.

## 5. КОНФИГУРАЦИЯ

### 5.4.3.2. Конфигурирование протокола ведомого устройства MODBUS (MODBUS Slave) через прямое представление (%Q)

Для конфигурирования данного протокола с использованием прямого представления (%Q) необходимо выполнить следующие действия:

- Настройка общих параметров протокола ведомого устройства MODBUS, таких как: время связи, адрес и переменные прямого представления (%Q) для получения диагностики и отношений управления.
- Добавление и конфигурирование отношений MODBUS с указанием типа данных MODBUS, переменных прямого представления (%Q) для приема/записи данных и количества данных для передачи.

Описание каждой настройки приведено ниже, в данном разделе.

#### 5.4.3.2.1. Общие параметры протокола ведомого устройства (MODBUS) – Конфигурация через прямое представление (%Q)

Общие параметры, расположенные на главном экране конфигурации протокола MODBUS (рисунок, приведенный ниже), определяются следующим образом:

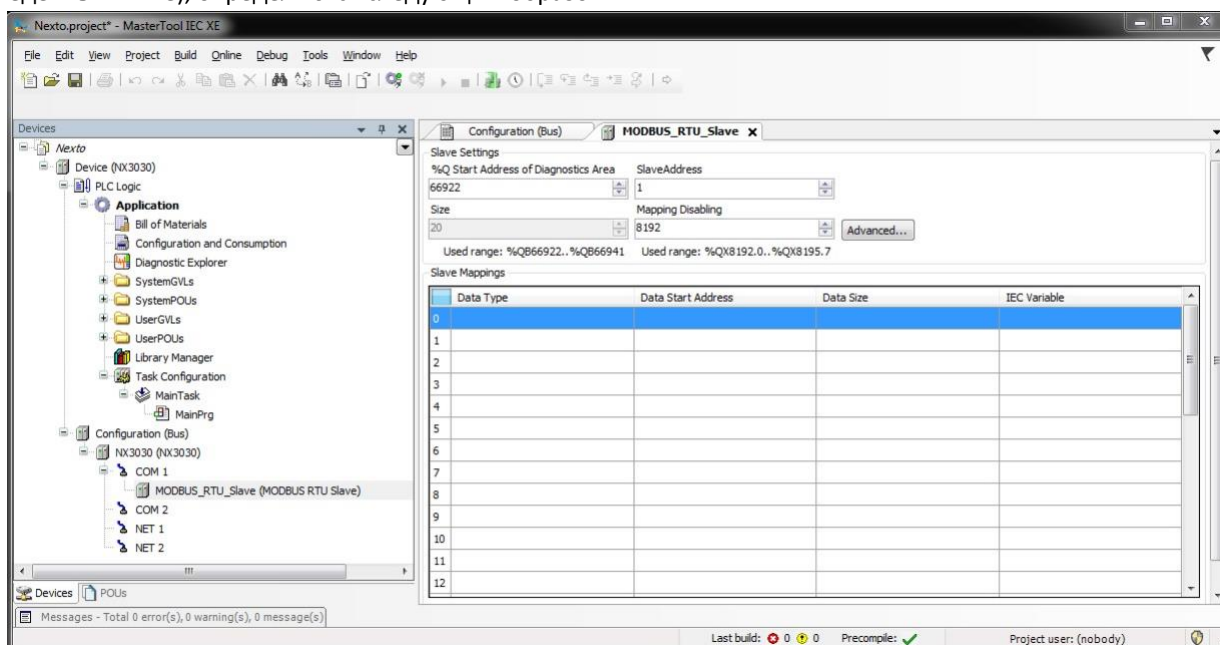


Рисунок 58: Экран конфигурации ведомого устройства MODBUS RTU

Адрес и переменные прямого представления (%Q) для отношений управления и диагностики:

Конфигурация	Описание	Значение по умолчанию	Опции
Начальный адрес %Q области диагностики	Начальный адрес диагностических переменных	-	От 0 до 2147483628
Размер	Размер области диагностики	-	Выключено для редактирования
Адрес ведомого устройства	Адрес ведомого устройства MODBUS	1	От 1 до 255
Выключение преобразования	Начальный адрес, используемый для отключения отношений MODBUS	-	От 0 до 2147483644

Таблица 67: Настройки адресных переменных и переменных прямого представления

## 5. КОНФИГУРАЦИЯ

Примечания:

Начальный адрес %Q области диагностики: данное поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого центрального процессора, который можно найти в разделе «Память».

Адрес ведомого устройства: важно отметить, что ведомое устройство принимает широковещательные запросы, когда ведущее устройство посылает команду с адресом, установленным на ноль. Помимо этого, в соответствии со стандартом MODBUS, диапазон допустимых адресов для ведомых устройств составляет от 1 до 247. Адреса с 248 по 255 зарезервированы.

Выключение преобразования: состоит из 32 битов и используется для отключения по отдельности 32 отношений MODBUS, сконфигурированных в пространстве «Преобразования ведомого устройства» (*Slave Mappings*). Отношение отключено, если соответствующий бит равен 1, в противном случае преобразование включено. Данное поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого центрального процессора (ЦП), который можно найти в разделе «Память».

Значение по умолчанию: для полей «Начальный адрес %Q области диагностики» (*%Q Start Address of Diagnostics Area*) и «Выключение преобразования» (*Mapping Disabling*) не может быть установлено заводское значение по умолчанию, поскольку создание связи может быть выполнено в любой момент при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), пока неиспользуемое.

Диагностика и команды протокола MODBUS описаны в таблице 64.

Время обмена данными по протоколу ведомого устройства MODBUS (MODBUS Slave), которое можно найти по кнопке «Дополнительно...» (*Advanced...*) экрана конфигурации, описано в таблице 63.

### 5.4.3.2.2. Конфигурация преобразований - Конфигурация через прямое представление (%Q)

Настройки отношений MODBUS, представленные на приведенных ниже рисунках, соответствуют параметрам, описанным в таблице:

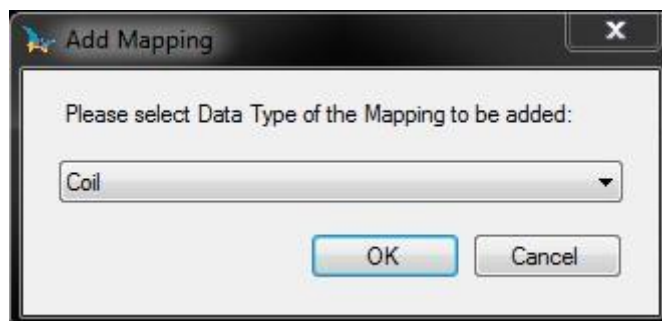


Рисунок 59: Добавление отношений протокола MODBUS

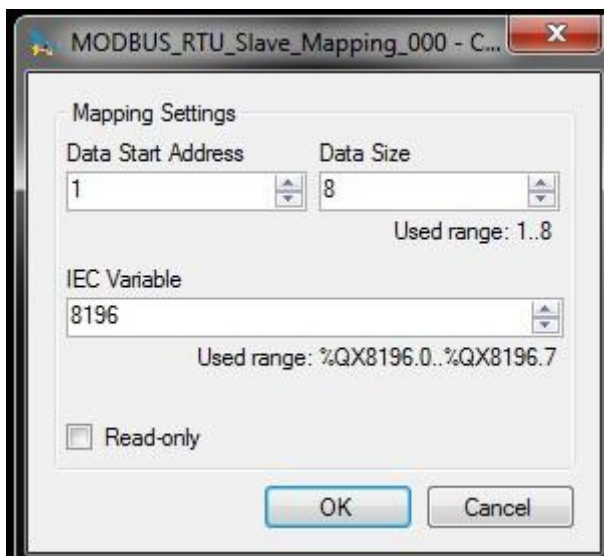


Рисунок 60: Конфигурирование отношения протокола MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Тип данных	Тип данных MODBUS	Регистр флагов	Регистр флагов (1 бит) Регистр временного хранения (16 бит) Входной регистр (16 бит) Состояние входов (1 бит)
Начальный адрес данных	Начальный адрес данных MODBUS	1	От 1 до 65536
Размер	Число данных MODBUS	-	От 1 до 65536
Переменная IEC	Начальные адреса переменных (%Q)	-	От 0 до 2147483647
Только чтение	Разрешено только чтение	Выключено	Включено или выключено

Таблица 68: Преобразования ведомого устройства

Примечания:

Опции: значения, записанные в столбце «Опции» (*Options*), могут изменяться в зависимости от конфигурации данных MODBUS.

Размер данных: значение «Размер данных» (*Data Size*) определяет максимальный объем данных, к которому может получить доступ отношение MODBUS, начиная с начального адреса. Таким образом, для считывания непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном интерфейсе.

Данное поле зависит от настроенного типа данных MODBUS, т.е. при выборе *Coil* («Регистр флагов») или *Input Status* («Состояние входов») поле «Размер данных» (*Data Size*) должно быть кратно восьми. Помимо этого, максимальный размер не должен превышать размер выходной адресуемой памяти и не должен присваивать те же значения, которые используются в приложении.

**ВНИМАНИЕ:**

При обращении к памяти данных связи между устройствами с различным порядком следования байтов многобайтного объекта в памяти или при передаче (эндианизм) (*Little-Endian* (порядок байтов от младшего к старшему) и *Big-Endian* (порядок байтов от старшего к младшему)) может произойти инверсия данных при считывании/записи. В этом случае пользователь должен скорректировать данные в приложении.

## 5. КОНФИГУРАЦИЯ

Переменная IEC: если тип данных MODBUS – *Coil* («Регистр флагов») или *Input Status* («Состояние входов») (бит), то начальный адрес переменных IEC будет в формате %QX10.1. Если же тип данных MODBUS – *Holding Register* («Регистр временного хранения») или *Input Register* («Регистр входов») (16 бит), то начальный адрес переменных IEC будет в формате %QW. Данное поле ограничено объемом памяти адресуемых выходных переменных (%Q) от каждого центрального процессора, который можно посмотреть в разделе «Память».

Только чтение: если данная опция включена, она позволяет ведущему устройству связи только считывать данные переменных. Выполнение записи не разрешено. Данная опция действительна только для функций записи.

Значение по умолчанию: значение по умолчанию не может быть определено для поля «Переменная IEC» (*IEC Variable*), поскольку создание отношения может быть выполнено в любой момент разработки приложения. Программное обеспечение MasterTool IEC XE самостоятельно выделяет значение из диапазона выходных переменных прямого представления (%Q), пока неиспользуемое. Для поля «Размер данных» (*Data Size*) нельзя определить значение по умолчанию, поскольку оно будет изменяться в зависимости от выбранного типа данных MODBUS.

В ранее определенных отношениях максимальный размер данных MODBUS может составлять 65535 (максимальное значение, конфигурируемое в поле «Размер данных» (*Data Size*)). Тем не менее, запрос, поступающий в ведомое устройство MODBUS RTU (MODBUS RTU Slave), должен обращаться к подгруппе данного преобразования, и данная группа должна иметь, максимум, размер данных, зависящий от кода функции, который определен ниже:

- Считывание регистров флагов (FC 1): 2000
- Считывание состояния входа (FC 2): 2000
- Считывание регистров временного хранения (FC 3): 125
- Считывание регистров входов (FC 4): 125
- Запись одного регистра флагов (FC 5): 1
- Запись одного регистра временного хранения (FC 6): 1
- Форсирование нескольких регистров флагов (FC 15): 1968
- Запись регистров временного хранения (FC 16): 123
- Маскирование регистра записи (FC 22): 1
- Считывание/запись регистров временного хранения (FC 23):
  - Считывание: 121
  - Запись: 121

### ВНИМАНИЕ:

В отличие от других прикладных задач, при достижении метки очистки в MainTask выполнение задачи экземпляра ведомого устройства MODBUS RTU и любой другой задачи MODBUS будет прекращено в тот момент, когда будет предпринята попытка производства записи в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока не выполняется MainTask.

### 5.4.4. Протокол MODBUS Ethernet

Связь с применением нескольких ведущих устройств позволяет центральным процессорам (ЦП) Nexto считывать или записывать переменные MODBUS в других контроллерах или на ЧМИ, совместимых с протоколом MODBUS TCP или MODBUS RTU через TCP. Центральный процессор (ЦП) Nexto может быть одновременно и клиентом, и сервером в одной коммуникационной сети, или даже иметь несколько экземпляров, связанных с интерфейсом сети Ethernet. При этом не имеет значения, относятся ли они к MODBUS TCP или MODBUS RTU через TCP, как описано в таблице 46.

На приведенном ниже рисунке представлены некоторые возможности связи с использованием протокола MODBUS TCP одновременно с протоколом MODBUS RTU через TCP.

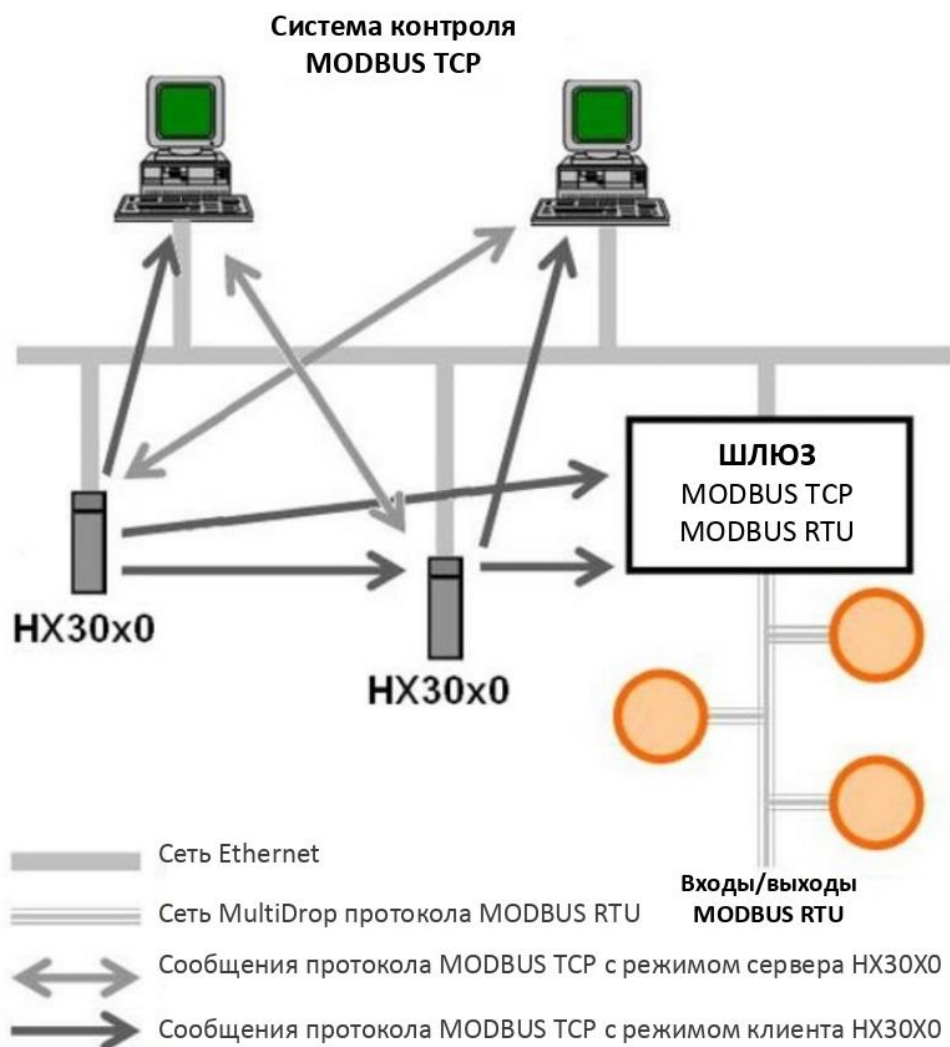


Рисунок 61: Коммуникационная сеть MODBUS TCP

Объединение переменных MODBUS с символическими переменными центрального процессора (ЦП) осуществляется пользователем путем определения отношений с помощью конфигуратора MasterTool IEC XE. В режиме сервера можно сконфигурировать до 32 связей, а в вериже клиента – до 128. При этом отношения в режиме клиента должны соответствовать максимальному размеру данных функции MODBUS: 125 регистров (регистры входов или регистры временного хранения) либо 2000 бит (регистры или состояние входов). Данная информация представлена в подробностях в описании каждого протокола.

Все отношения в режиме клиента или сервера могут быть отключены через переменные прямого представления (%Q), которые в MasterTool IEC XE идентифицируются как «Переменные отключения» (Disabling Variables). Отключение может происходить через общие биты, которые влияют на все отношения режима работы, или через специфические биты, влияющие на конкретные отношения.

Для отношений серверного режима работы кластеры IP-адресов могут быть определены с разрешением на запись и чтение, называемыми фильтрами. Это делается через определение сетевого IP-адреса и маски подсети, в результате чего образуется группа клиентских IP-адресов, которые могут считывать и записывать в переменные отношения. Функции считывания/записи фильтруются, другими словами, они не могут быть запрошены ни одним клиентом, независимо от IP-адреса. Данная информация подробно описана в протоколе сервера MODBUS Ethernet (MODBUS Ethernet Server).

При использовании протокола MODBUS TCP в режиме клиента для ускорения обмена данными с серверами можно использовать функцию многократных запросов через одно и то же TCP-соединение. Если данная функция не нужна или

## 5. КОНФИГУРАЦИЯ

не поддерживается сервером, ее можно отключить (действие на уровне отношения). Важно подчеркнуть, что максимальное количество TCP-соединений между клиентом и сервером составляет 63. При изменении некоторых параметров неактивные связи могут быть закрыты, что позволит открыть новые соединения.

В приведенных ниже таблицах приведен, соответственно, полный перечень функций передачи данных и MODBUS, поддерживаемых центральными процессорами (ЦП) Nexto.

Тип данных	Размер [биты]	Описание
Регистр флагов	1	Цифровой выход с возможностью чтения или записи.
Состояние входов	1	Цифровой вход (только чтение).
Регистр временного хранения	16	Аналоговый выход, который может быть считан или записан.
Регистр входов	16	Аналоговый вход (только чтение).

Таблица Типы данных протокола MODBUS, поддерживаемые центральными процессорами (ЦП) Nexto

Тип функции	Код		Описание
	DEC	HEX	
Доступ к переменным	1	0x01	Считывание регистров флагов (FC 01)
	2	0x02	Считывание состояния входа (FC 02)
	3	0x03	Считывание регистров временного хранения (FC 03)
	4	0x04	Считывание регистров входов (FC 04)
	5	0x05	Запись одного регистра флагов (FC 05)
	6	0x06	Запись одного регистра временного хранения (FC 06)
	15	0x0F	Запись нескольких регистров флагов (FC 15)
	16	0x10	Запись нескольких регистров временного хранения (FC 16)
	22	0x16	Маска записи регистра временного хранения (FC 22)
	23	0x17	Считывание/Запись нескольких регистров временного хранения (FC 23)

Таблица 70: Функции протокола MODBUS, поддерживаемые центральными процессорами (ЦП) Nexto

Независимо от режима конфигурирования, шаги по вставке экземпляра протокола и настройке интерфейса сети Ethernet одинаковы. Остальные шаги конфигурирования описаны ниже для каждого режима.

- Добавьте в канал сети Ethernet один или несколько экземпляров протокола клиента или сервера MODBUS Ethernet. Для выполнения данной процедуры обратитесь к разделу «[Вставка экземпляра протокола](#)».
- Настройте интерфейс сети Ethernet. Для выполнения данной процедуры обратитесь к разделу «[Конфигурация интерфейсов сети Ethernet](#)».

### 5.4.5. Клиент MODBUS Ethernet (MODBUS Ethernet Client)

Данный протокол доступен для всех центральных процессоров (ЦП) серии Nexto на их каналах сети Ethernet. При выборе данной опции в MasterTool IEC XE центральный процессор (ЦП) становится клиентом связи MODBUS, что позволяет ему в режиме исполнения (режим «*Выполнение*» (Run)) получать доступ к другим устройствам с тем же протоколом.

Существует два способа конфигурирования данного протокола. Первый – через *прямое представление (%Q)*, при котором переменные определяются своим адресом. Второй – через *символьное преобразование*, при котором переменные определяются по имени.

Процедура вставки экземпляра протокола подробно описана в руководстве пользователя MasterTool IEC XE – MU299609 или в разделе «[Вставка экземпляра протокола](#)».



## 5. КОНФИГУРАЦИЯ

### 5.4.5.1. Конфигурирование клиента MODBUS Ethernet через символьное преобразование

Для конфигурирования данного протокола с использованием *символьного преобразования* необходимо выполнить следующие шаги:

- Сконфигурируйте общие параметры клиента протокола MODBUS, с протоколом управления передачей (TCP) или RTU через TCP.
- Добавьте и сконфигурируйте устройства, задав IP-адрес, порт, адрес ведомого устройства и блокировку по превышению лимита времени связи (доступно по кнопке «Дополнительные настройки» (Advanced Settings) устройства).
- Добавьте и сконфигурируйте преобразования MODBUS, указав имя переменной, тип данных, начальный адрес данных, размер данных и переменную, которая будет получать данные о качестве.

Добавьте и сконфигурируйте запрос MODBUS, указав желаемую функцию, время сканирования запроса, начальный адрес (считывание/запись), размер данных (считывание/запись), переменную, которая будет получать данные о качестве, и переменную, отвечающую за отключение запроса.

#### 5.4.5.1.1. Общие параметры протокола клиента MODBUS - конфигурирование через символьное преобразование

Общие параметры, представленные на начальном экране конфигурации протокола MODBUS (рисунок, приведенный ниже), определяются следующим образом:

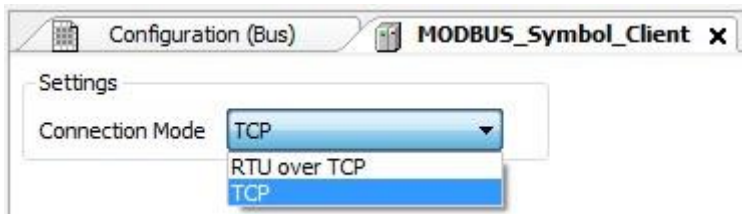


Рисунок 62: Экран конфигурации общих параметров клиента MODBUS

Конфигурация	Описание	По умолчанию	Опции
Режим соединения	Выбор протокола	TCP	RTU через TCP TCP

Таблица 71: Общие конфигурации клиента MODBUS

Диагностика и команды протокола клиента MODBUS (MODBUS Client), сконфигурированные либо по символьному преобразованию, либо по прямому представлению, хранятся в переменных *T\_DIAG\_MODBUS\_ETH\_CLIENT\_1*. При преобразовании прямого представления они также находятся в 4 байтах и 8 словах, которые описаны в таблице, приведенной ниже (где «n» - сконфигурированное значение в поле «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area)):

Переменная прямого представления	Диагностическая переменная <i>T_DIAG_MODBUS_ETH_CLIENT_1.*</i>	Размер	Описание
Диагностические биты:			
%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).

5. КОНФИГУРАЦИЯ

%QX(n).2	tDiag. bInterruptedByCommand	BIT	Бит bNotRunning был включен, поскольку клиент был прерван пользователем через задающие биты.
%QX(n).3	tDiag. bConfigFailure	BIT	Прекращение диагностики.
%QX(n).4	tDiag. bRXFailure	BIT	Прекращение диагностики.
%QX(n).5	tDiag. bTXFailure	BIT	Прекращение диагностики.
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_CLIENT_1.*	Размер	Описание
%QX(n).6	tDiag. bModuleFailure	BIT	Указывает на наличие сбоя в модуле или его отсутствие.
%QX(n).7	tDiag. bAllDevicesCommFailure	BIT	Указывает на то, что все устройства, сконфигурированные в клиенте, находятся в состоянии отказа.
%QB(n+1)	byDiag_1_reserved	BYTE	Зарезервировано.
Биты команд, автоматически инициализируемые:			
%QX(n+2).0	tCommand. bStop	BIT	Останов клиента.
%QX(n+2).1	tCommand. bRestart	BIT	Перезапуск клиента.
%QX(n+2).2	tCommand. bResetCounter	BIT	Перезапуск статистики 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.
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Зарезервировано.
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.
Статистика коммуникаций:			
%QW(n+4)	tStat. wTXRequests	WORD	Счетчик количества запросов, переданных клиентом (от 0 до 65535).
%QW(n+6)	tStat. wRXNormalResponses	WORD	Счетчик нормальных откликов, полученных клиентом (от 0 до 65535).

## 5. КОНФИГУРАЦИЯ

%QW(n+8)	tStat. wRXExceptionResponses	WORD	Счетчик откликов с кодом исключения (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalResponses	WORD	Счетчик недействительных откликов, полученных клиентом - неверный синтаксис, неверный CRC или недостаточное количество полученных байт (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 72: Диагностика протокола клиента MODBUS

Примечание:

Счетчики: Все счетчики диагностики клиента MODBUS TCP (MODBUS TCP Client) возвращаются к нулю при превышении предельного значения 65535.

### 5.4.5.1.2. Конфигурация устройств - Конфигурация через символьное преобразование

Конфигурация устройств, представленная на приведенном ниже рисунке, соответствует следующим параметрам:

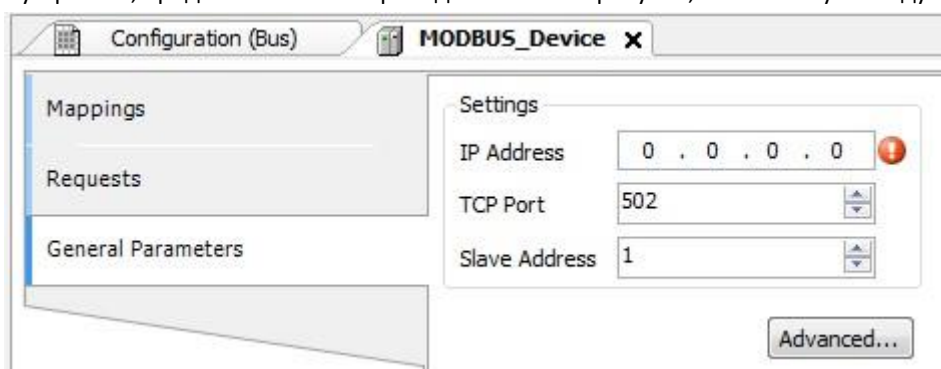


Рисунок 63: Настройки общих параметров устройства

Конфигурация	Описание	По умолчанию	Опции
IP-адрес	IP-адреса сервера	0.0.0.0	От 1.0.0.1 до 223.255.255.255
Порт TCP	Порт TCP	502	От 2 до 65534
Адрес ведомого устройства	Адрес ведомого устройства MODBUS	1	От 0 до 255

Таблица 73: Общие конфигурации клиента MODBUS

Примечания:

IP-адрес: IP-адрес устройства сервера Modbus (Modbus Server).

Порт TCP: если в один интерфейс сети Ethernet добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные порты TCP. Некоторые порты TCP, среди перечисленных выше,

## 5. КОНФИГУРАЦИЯ

зарезервированы и поэтому не могут быть использованы. К ним относятся: 80, 8080, 1217, 1740, 1741, 1742, 1743 и 11740.

Адрес ведомого устройства: согласно стандарту MODBUS, диапазон допустимых адресов для ведомых устройств составляет от 0 до 247, при этом адреса от 248 до 255 являются зарезервированными. Когда ведущее устройство посылает команду записи с адресом, установленным в ноль, оно выполняет широкоэвещательные запросы по сети.

Параметры в расширенных настройках устройства клиента MODBUS (MODBUS Client), определяемое по кнопке «Дополнительно...» (*Advanced...*) на вкладке «Общие параметры» (*General Parameters*), подразделяются на: «Максимальное количество одновременных запросов» (*Maximum Simultaneous Requests*), «Блокировка по превышению лимита времени связи» (*Communication Time-out*), «Режим блокировки по превышению лимита времени связи» (*Mode of Connection Time-out*) и «Время бездействия» (*Inactive Time*).

Конфигурация	Описание	По умолчанию	Опции
Максимальное количество одновременных запросов	Количество одновременных запросов, которые клиент может запросить у сервера	1	От 1 до 8
Блокировка по превышению лимита времени связи (мс)	Блокировка по превышению лимита времени на уровне приложения в мс	3000	От 10 до 65535
Режим	Определяет момент завершения соединения клиента с сервером	Соединение закрывается по истечении времени бездействия (с): 10 - 3600.	Соединение закрывается по истечении блокировки по превышению лимита времени. Соединение закрывается по окончании каждого обмена данными. Соединение закрывается по истечении времени бездействия (с): от 10 до 3600.
Время бездействия (с)	Время бездействия	10	3600

Таблица 74: Расширенные конфигурации клиента MODBUS

### Примечания:

Максимальное количество одновременных запросов: используется при высоком значении цикла сканирования. Данный параметр фиксируется на 1 (не редактируется), если конфигурируемый протокол – MODBUS RTU через TCP.

Блокировка по превышению лимита времени связи: блокировка по превышению лимита времени связи представляет собой время, в течение которого клиент будет ожидать ответа сервера на запрос. Для устройства клиента MODBUS (MODBUS Client) необходимо учитывать две переменные системы: время обработки запроса сервером и задержку отправки ответа, если она задана в сервере. Рекомендуется, чтобы блокировка по превышению лимита времени была равна или превышала удвоенную сумму данных параметров. Дополнительную информацию см. в разделе «Производительность связи» ([Communication Performance](#)).

Режим: определяет, когда соединение с сервером будет завершено клиентом. Ниже перечислены доступные варианты:

- Соединение закрывается блокировки по превышению лимита времени или соединение никогда не закрывается в обычных ситуациях: данные опции представляют собой аналогичную логику работы Клиента, закрывающего соединение ввиду отсутствия ответа на запрос со стороны Сервера до достижения блокировки по превышению лимита времени связи.
- Соединение закрывается по окончании каждого обмена данными: соединение закрывается клиентом после завершения каждого запроса.
- Соединение закрывается по истечении времени неактивности: соединение будет закрыто Клиентом, если оно достигнет времени неактивности без выполнения запроса к Серверу.

## 5. КОНФИГУРАЦИЯ

Время бездействия: время неактивности соединения.

### 5.4.5.1.3. Конфигурация преобразований - Конфигурация через символическое преобразование

Конфигурация отношений протокола MODBUS, показанная на рисунке, приведенном ниже, соответствует параметрам, описанным в таблице ниже:

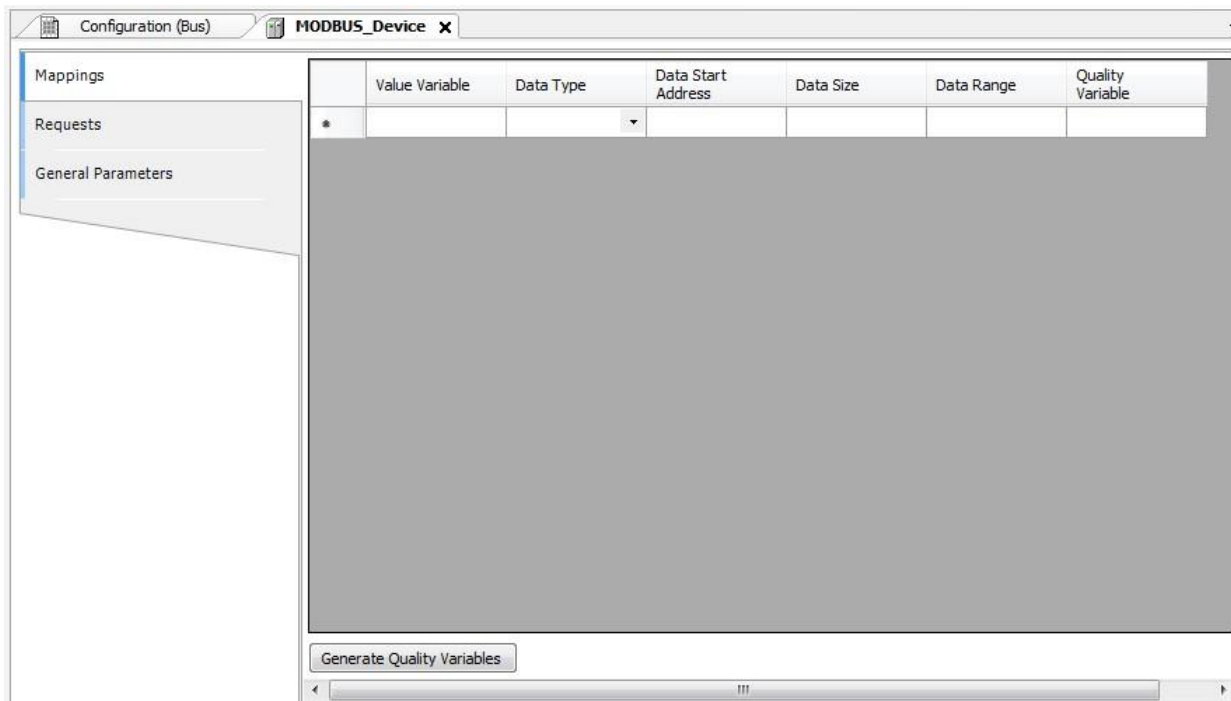


Рисунок 64: Тип данных протокола MODBUS

Конфигурация	Описание	По умолчанию	Опции
Переменная значения	Имя символической переменной	-	Имя переменной, объявленной в программе или списке глобальных переменных (GVL)
Тип данных	Тип данных MODBUS	-	Регистр флагов - Запись (1 бит) Регистр флагов - Чтение (1 бит) Регистр временного хранения - Запись (16 бит) Регистр временного хранения - Чтение (16 бит) Регистр временного хранения – Маска логической операции «И» (16 бит) Регистр временного хранения – Маска логической операции «ИЛИ» (16 бит) Регистр входа (16 бит)

## 5. КОНФИГУРАЦИЯ

			Состояние входа (1 бит)
Начальный адрес данных	Начальный адрес данных MODBUS	-	От 1 до 65536
Размер данных	Размер данных MODBUS	-	От 1 до 65536
Диапазон данных	Диапазон адресов сконфигурированных данных	-	-

Таблица 75: Настройки преобразований протокола MODBUS

Примечания:

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

Тип данных: данное поле используется для указания типа данных, используемого в отношении MODBUS.

Тип данных	Размер [биты]	Описание
Регистр флагов - Запись	1	Запись цифрового выхода.
Регистр флагов - Чтение	1	Считывание цифрового выхода.
Регистр временного хранения - Запись	16	Запись аналогового выхода.
Регистр временного хранения - Чтение	16	Считывание аналогового выхода.
Регистр временного хранения - маска «И»	16	Аналоговый выход, который может быть считан или записан с помощью маски «И».
Регистр временного хранения - маска «ИЛИ»	16	Аналоговый выход, который может быть считан или записан с помощью маски «ИЛИ».
Регистр входа	16	Аналоговый вход, который может быть только считан.
Состояние входа	1	Цифровой вход, который может быть только считан.

Таблица 76: Типы данных, поддерживаемые протоколом MODBUS

Начальный адрес данных: Начальный адрес данных преобразования MODBUS.

Размер данных: Значение размера данных определяет максимальный объем данных, к которому может получить доступ интерфейс MODBUS, начиная с начального адреса. Таким образом, для считывания непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном интерфейсе. Данное поле изменяется в зависимости от настроенного типа данных MODBUS.

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

### 5.4.5.1.4. Конфигурация запросов - Конфигурация через символьное преобразование

Конфигурация запросов MODBUS, представленная на рисунке, приведенном ниже, соответствует параметрам, описанным в таблице:

## 5. КОНФИГУРАЦИЯ

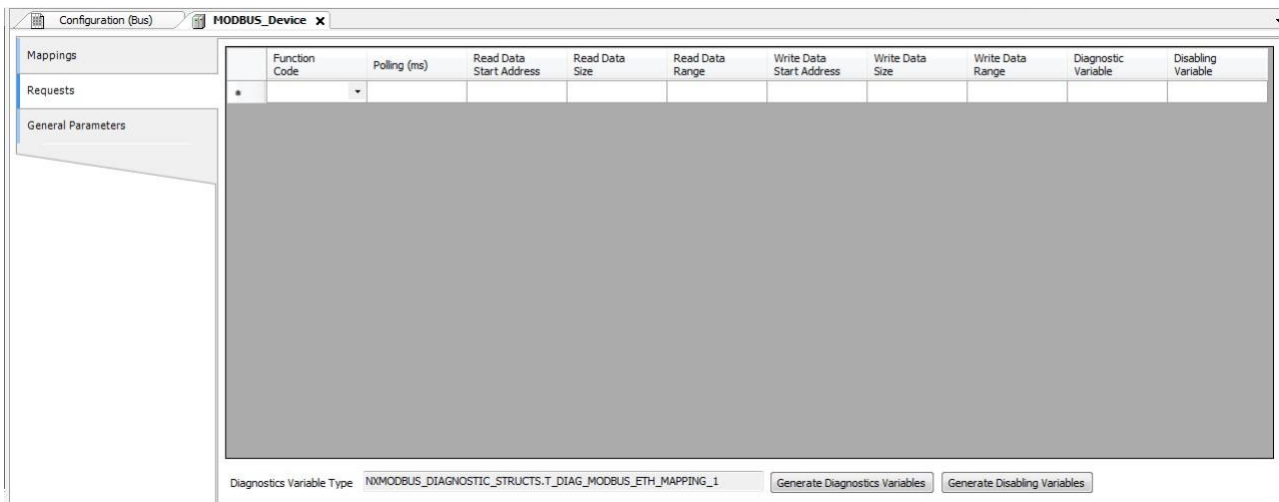


Рисунок 65: Экран запроса данных протокола MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Код функции	Тип функции MODBUS	-	01 – Считывание регистров флагов 02 – Считывание состояния входа 03 – Считывание регистров временного хранения 04 – Считывание регистров входов 05 – Запись одного регистра флагов 06 – Запись одного регистра 15 – Запись нескольких регистров флагов 16 – Запись нескольких регистров 22 – Регистр записи масок 23 – Чтение/запись нескольких регистров
Упорядоченный опрос (мс)	Период связи (мс)	100	от 0 до 3600000
Считывание начального адреса данных	Начальный адрес данных считывания MODBUS	-	от 1 до 65536
Считывание размера данных	Размер данных для считывания MODBUS	-	В зависимости от используемой функции
Считывание диапазона данных	Диапазон адресов считывания данных MODBUS	-	от 0 до 2147483646
Запись начального адреса данных	Начальный адрес данных записи MODBUS	-	от 1 до 65536
Запись размера данных	Размер данных записи MODBUS	-	В зависимости от используемой функции

## 5. КОНФИГУРАЦИЯ

Запись диапазона данных	Диапазон адресов записи данных MODBUS	-	от 0 до 2147483647
Диагностическая переменная	Имя диагностической переменной	-	Имя переменной, объявленной в программе или списке глобальных переменных (GVL)
Выключающая переменная	Переменная, используемая для отключения связи с MODBUS	-	Поле для символической переменной, используемой для индивидуального отключения сконфигурированных запросов MODBUS. Данная переменная должна относиться к типу BOOL. Переменная может быть простым элементом или элементом массива и может находиться в структурах.

Таблица 77: Конфигурация отношений протокола MODBUS

Примечания:

Уставка: количество заводских уставок параметров по умолчанию и значения для столбца «Опции» (Options) могут отличаться в зависимости от типа данных и функции MODBUS (FC).

Код функции: Доступны следующие функции MODBUS (FC):

Тип функции	Код		Описание
	DEC	HEX	
Доступ к переменным	1	0x01	Считывание регистров флагов (FC 01)
	2	0x02	Считывание состояния входа (FC 02)
	3	0x03	Считывание регистров временного хранения (FC 03)
	4	0x04	Считывание регистров входов (FC 04)
	5	0x05	Запись одного регистра флагов (FC 05)
	6	0x06	Запись одного регистра временного хранения (FC 06)
	15	0x0F	Запись нескольких регистров флагов (FC 15)
	16	0x10	Запись нескольких регистров временного хранения (FC 16)
	22	0x16	Маска записи регистра временного хранения (FC 22)
	23	0x17	Считывание/Запись нескольких регистров временного хранения (FC 23)

Таблица 78: Функции протокола MODBUS, поддерживаемые центральными процессорами (ЦП) Nexto

Упорядоченный опрос: данный параметр указывает на то, как часто должен выполняться обмен данными, установленный для данного запроса. По окончании обмена данными, время ожидания будет равно значению, заданному в поле «Упорядоченный опрос» (polling), после чего будет выполнен новый обмен данными.

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

Размер считываемых данных: минимальное значение размера считываемых данных равно 1, в то время как максимальное зависит от используемой функции MODBUS (FC), как показано ниже:

- Считывание регистров флагов (FC 01): 2000
- Считывание состояния входа (FC 02): 2000
- Считывание регистров временного хранения (FC 03): 125
- Считывание регистров входов (FC 04): 125
- Считывание/запись нескольких регистров временного хранения (FC 23): 121



## 5. КОНФИГУРАЦИЯ

Диапазон считываемых данных: данное поле показывает диапазон считываемых данных MODBUS, сконфигурированный для каждого запроса. Начальный адрес вместе с размером считываемых данных приводит к диапазону считываемых данных для каждого запроса.

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

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

- Запись одного регистра флагов (FC 05): 1
- Запись одного регистра (FC 06): 1
- Запись нескольких регистров флагов (FC 15): 1968
- Запись нескольких регистров (FC 16): 123
- Маскирование регистра записи (FC 22): 1
- Считывание/запись нескольких регистров (FC 23): 121

Диапазон данных записи: данное поле показывает диапазон данных записи MODBUS, сконфигурированный для каждого запроса. Начальный адрес вместе с размером считываемых данных приводит к диапазону данных записи для каждого запроса.

Диагностическая переменная: Диагностика запросов MODBUS, сконфигурированная по символьному преобразованию или по прямому представлению, хранится в переменных типа *T\_DIAG\_MODBUS\_RTU\_MAPPING\_1* для ведущих устройств и *T\_DIAG\_MODBUS\_ETH\_CLIENT\_1* для устройств клиента, а преобразование по прямому представлению - в 4-байтовых и 2-словных, которые описаны в таблице 57 (n - значение, заданное в поле «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area)).

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_MAPPING_1.*	Размер	Описание
Биты состояния связи:			
%QX(n).0	byStatus. bCommIdle	BIT	Неактивная связь (ожидание выполнения).
%QX(n).1	byStatus. bCommExecuting	BIT	Активная связь.
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_MAPPING_1.*	Размер	Описание
%QX(n).2	byStatus. bCommPostponed	BIT	Связь отложена, поскольку достигнуто максимальное количество одновременных запросов. Отложенные сообщения будут выполняться в той же последовательности, в которой они были заказаны, чтобы избежать неопределенности. Время, проведенное в этом состоянии, не учитывается для целей блокировки по превышению лимита времени. Биты bCommIdle и bCommExecuting ложны (false), если бит bCommPostponed истинен (true).
%QX(n).3	byStatus. bCommDisabled	BIT	Связь отключена. В этом состоянии бит bCommIdle перезапускается.
%QX(n).4	byStatus. bCommOk	BIT	Связь, завершенная ранее, была успешной.

## 5. КОНФИГУРАЦИЯ

%QX(n).5	byStatus. bCommError	BIT	В ранее завершенной связи возникла ошибка. Проверьте код ошибки.
%QX(n).6	byStatus. bCommAborted	BIT	Ранее завершенная связь была прервана из-за разрыва соединения.
%QX(n).7	byStatus. bDiag_7_reserved	BIT	Зарезервировано.
Код последней ошибки (включается, когда bCommError = TRUE («ИСТИНА»)):			
%QB(n+1)	eLastErrorCode	MASTER_ERROR _CODE (BYTE)	Информирует о возможной причине последней ошибки в части преобразования MODBUS. Более подробная информация приведена в таблице 80.
Последний код исключения, полученный ведущим устройством:			
%QB(n+2)	eLastExceptionCode	MODBUS_EXCEP- TION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*
Статистика коммуникаций:			
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.
%QW(n+4)	wCommCounter	WORD	Счетчик связи завершен, с ошибками или без. Пользователь может проверить, когда закончилась связь, по изменению этого счетчика. 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 79: Диагностика отношений клиента MODBUS

### Примечания:

Коды исключений: коды исключений, показанные в данном файле, представляют собой возвращаемые сервером значения. Определения кодов исключений 128, 129 и 255 действительны только для ведомых устройств компании Altus. Для ведомых других производителей данные коды исключений могут иметь другие значения.

Переменная отключения: поле для переменной, используемой для отключения запросов MODBUS, индивидуально сконфигурированных внутри запросов. Запрос отключен, если переменная, соответствующая запросу, равна 1, в противном случае запрос включен.

Код последней ошибки: Ниже приведены коды возможных ситуаций, приводящих к ошибке при обмене данными по протоколу MODBUS:

Код	Счетное значение	Описание
1	ERR_EXCEPTION	Ответ находится в коде исключения (см. eLastExceptionCode = Exception Code («Код исключения»)).
2	ERR_CRC	Ответ с недействительным CRC.

## 5. КОНФИГУРАЦИЯ

3	ERR_ADDRESS	Адрес MODBUS не найден. Адрес, ответивший на запрос, отличается от ожидаемого.
4	ERR_FUNCTION	Неверный код функции. Код функции ответа отличается от ожидаемого.
5	ERR_FRAME_DATA_COUNT	Объем данных в ответе оказался иным, чем ожидалось.
7	ERR_NOT_ECHO	Ответ не является эхоотображением (воспроизведением) запроса (FC 05 и 06).
8	ERR_REFERENCE_NUMBER	Неверный номер ссылки (FC 15 и 16).
9	ERR_INVALID_FRAME_SIZE	Ответ оказался короче, чем ожидалось.
20	ERR_CONNECTION	Ошибка при установлении соединения.
21	ERR_SEND	Ошибка на этапе передачи данных.
22	ERR_RECEIVE	Ошибка на этапе приема данных.
40	ERR_CONNECTION_TIMEOUT	Блокировка по превышению лимита времени на уровне приложения во время соединения.
41	ERR_SEND_TIMEOUT	Блокировка по превышению лимита времени на уровне приложения при передаче данных.
42	ERR_RECEIVE_TIMEOUT	Блокировка по превышению лимита времени на уровне приложения во время ожидания отклика.
43	ERR_CTS_OFF_TIMEOUT	Блокировка по превышению лимита времени при ожидании CTS = false («ЛОЖЬ») при передаче.
44	ERR_CTS_ON_TIMEOUT	Блокировка по превышению лимита времени при ожидании CTS = TRUE («ИСТИНА») при передаче.
128	NO_ERROR	С момента запуска ошибок нет.

Таблица 80: Коды ошибок отношений MODBUS

**ВНИМАНИЕ:**

В отличие от других задач приложения, при достижении метки при отладке MainTask выполнение задачи экземпляра клиента MODBUS Ethernet (MODBUS Ethernet Client) или любой другой задачи MODBUS прекращается в тот момент, когда она пытается произвести запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока MainTask не запущена.

### 5.4.5.2. Конфигурация клиента MODBUS Ethernet (MODBUS Ethernet Client) через прямое представление (%Q)

Для конфигурирования данного протокола с использованием прямого представления (%Q) необходимо выполнить следующие действия:

- Конфигурирование общих параметров протокола MODBUS, таких как: время связи и переменные прямого представления (%Q) для получения диагностики.
- Добавление и конфигурирование устройств путем задания адреса, переменных прямого представления (%Q) для отключения отношений, блокировок по превышению лимита времени связи и т.д.
- Добавление и конфигурирование отношений MODBUS с указанием типа данных и функции MODBUS, блокировок по превышению лимита времени, переменных прямого представления (%Q) для получения диагностики отношения и других для приема/записи данных, количества передаваемых данных и упорядоченного опроса отношения.

Описание каждой конфигурации приведено ниже в данном разделе.

#### 5.4.5.2.1. Общие параметры клиента протокола MODBUS - конфигурация для прямого представления (%Q)

Общие параметры, расположенные на главном экране конфигурации протокола MODBUS (рисунок, приведенный ниже), определяются следующим образом:

## 5. КОНФИГУРАЦИЯ

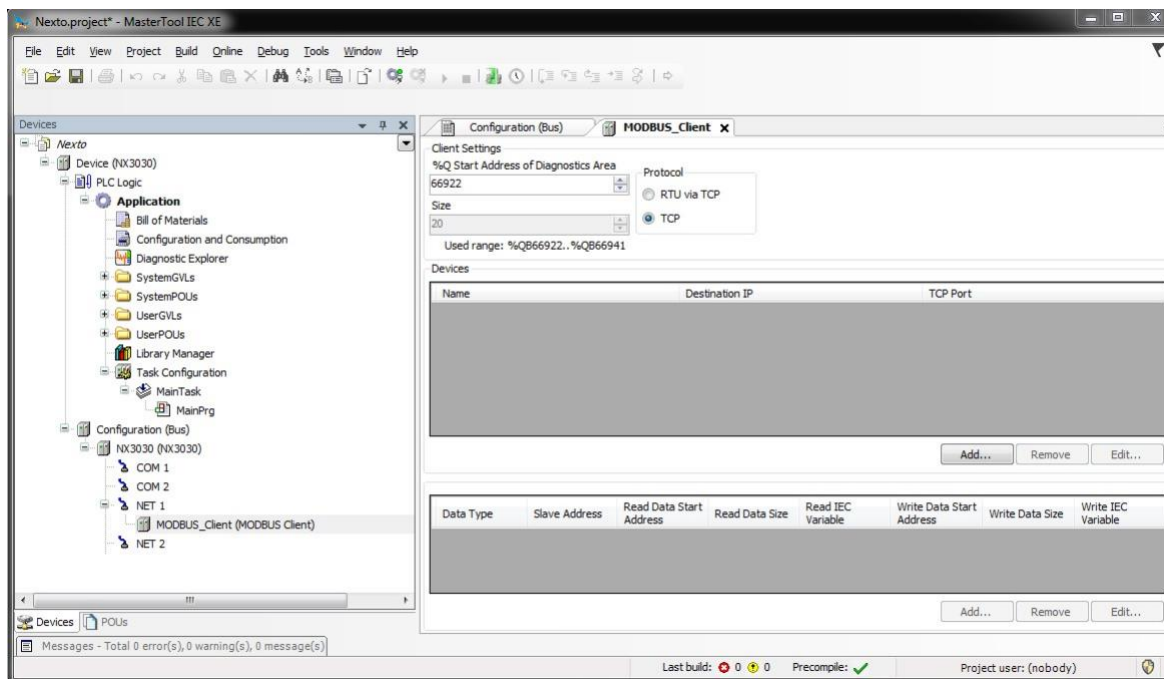


Рисунок 66: Экран настройки клиента MODBUS

Выбор протокола и переменных прямого представления (%Q) для диагностики:

Настройка	Описание	Значение по умолчанию	Опции
Начальный адрес %Q области диагностики	Начальный адрес диагностических переменных	-	от 0 до 2147483628
Размер	Размер диагностики	20	Выключено для редактирования
Протокол	Выбор протокола	TCP	RTU через TCP TCP

Таблица 81: Настройки клиента MODBUS

Примечания:

Начальный адрес %Q области диагностики: данное поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого центрального процессора, который можно найти в разделе «Память».

Значение по умолчанию: значение по умолчанию для поля «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area) не может быть определено, поскольку создание экземпляра протокола может быть произведено в любой момент в рамках разработки приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), пока неиспользуемое.

Диагностика и команды протокола MODBUS описаны в таблице 72.

### 5.4.5.2.2. Конфигурирование устройства - Конфигурация через прямое представление (%Q)

Конфигурация устройств, представленная на приведенном ниже рисунке, включает в себя следующие параметры:

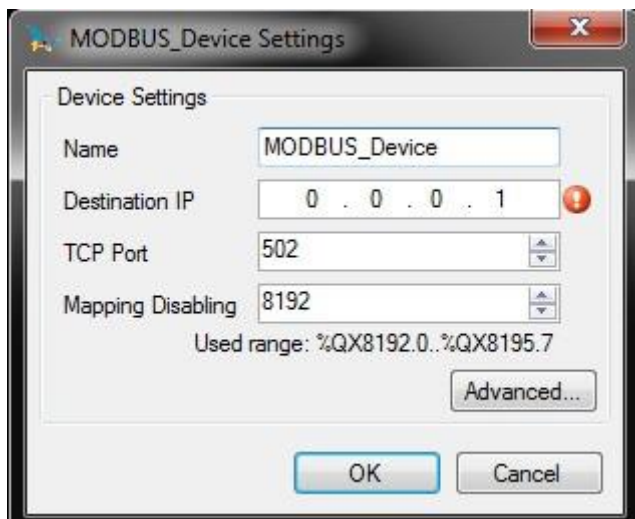


Рисунок 67: Конфигурирование клиента MODBUS

Конфигурация	Описание	Заводские настройки по умолчанию	Опции
Имя	Наименование экземпляра	MODBUS_Device	Идентификатор, в соответствии с МЭК 61131-3
IP-адрес назначения	IP-адрес сервера	0. 0. 0. 1	от 1.0.0.1 до 223.255.255.255
Порт TCP	Порт TCP	502	от 2 до 65534
Выключение преобразования	Начальный адрес, используемый для отключения связи MODBUS	-	Любой адрес области %Q, ограниченный моделью центрального процессора (ЦП)

Таблица 82: Конфигурация клиентских устройств

**Примечания:**

Наименование экземпляра: данное поле является идентификатором устройства, проверка которого осуществляется в соответствии с МЭК 61131-3 (IEC 61131-3), т.е. не допускает пробелов, специальных символов и начинающихся с числовых символов. Оно ограничено 24 символами.

Порт TCP: если в один интерфейс сети Ethernet добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные порты TCP. Некоторые порты TCP, среди перечисленных выше, зарезервированы и поэтому не могут быть использованы. К ним относятся: 80, 8080, 1217, 1740, 1741, 1742, 1743 и 11740.

Выключение преобразования: состоит из 32 битов и используется для отключения по отдельности 32 отношений MODBUS, сконфигурированных в пространстве «Преобразований устройств» (*Device Mappings*). Отношение отключено, если соответствующий бит равен 1, в противном случае преобразование включено. Данное поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого центрального процессора (ЦП), который можно найти в разделе «Память».

Значение по умолчанию: заводское значение для поля «Выключение преобразования» (*Mapping Disabling*) не может быть установлено, поскольку создание экземпляра протокола может быть произведено в любой момент в рамках разработки приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), пока неиспользуемое.

## 5. КОНФИГУРАЦИЯ

Блокировка по превышению лимита времени связи: настройки, присутствующие при нажатии кнопки «Дополнительно...» (Advanced...) на TCP-соединении, описаны в примечаниях к разделу «Конфигурация устройств - Конфигурация через символьное преобразование».

### 5.4.5.2.3. Конфигурация преобразований - Конфигурация через прямое представление (%Q)

Настройки отношений MODBUS, представленные на приведенных ниже рисунках, соответствуют параметрам, описанным в таблице:

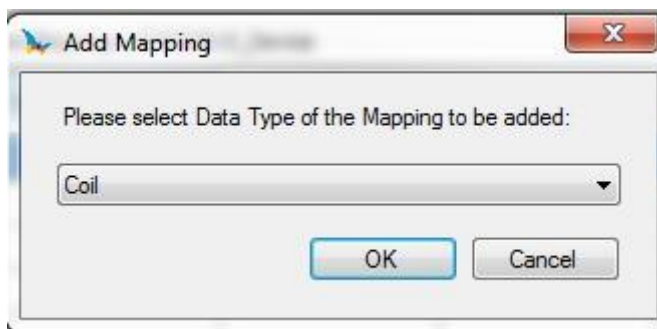


Рисунок 68: Тип данных протокола MODBUS

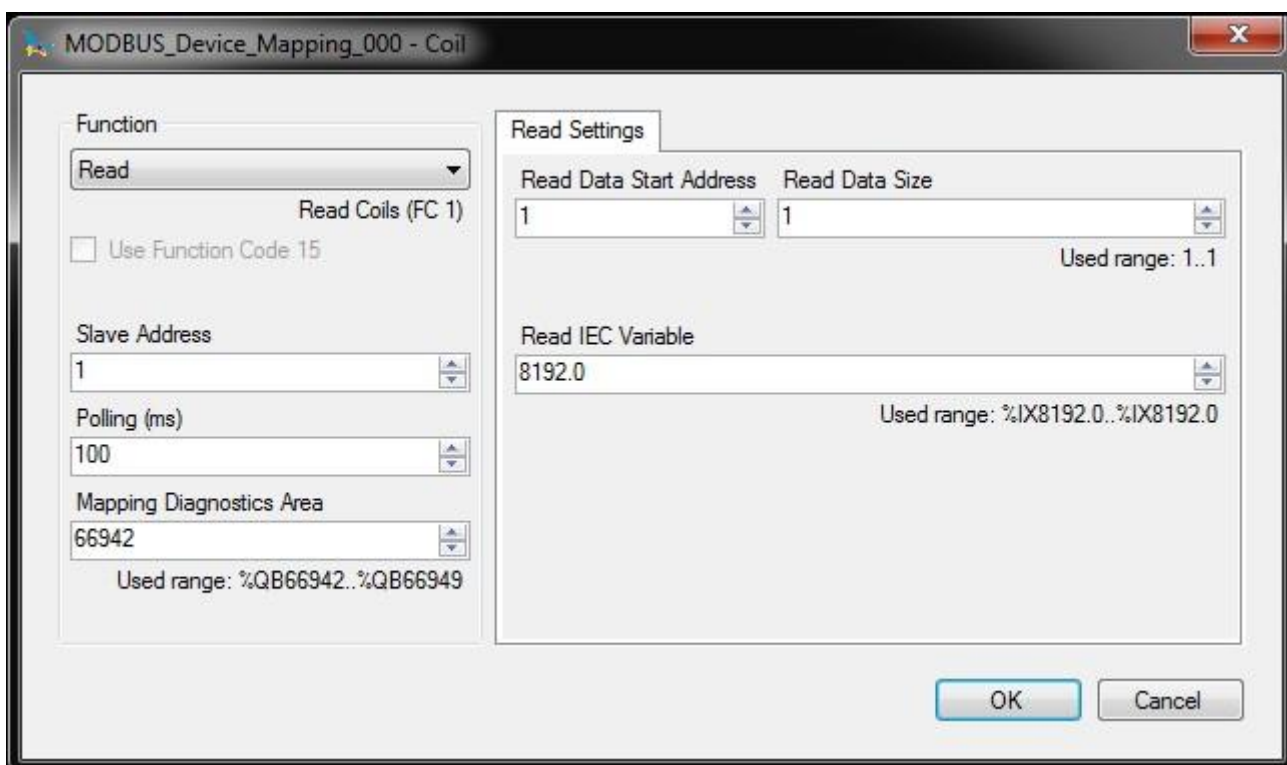


Рисунок 69: Функция протокола MODBUS

В приведенной ниже таблице количество заводских установок по умолчанию и значения для столбца «Опции» (Options) могут отличаться в зависимости от типа данных и функции MODBUS (FC).

Конфигурация	Описание	Значение по умолчанию	Опции
Функция	Адрес ведомого устройства MODBUS	Считывание	Считывание Запись Считывание/запись

## 5. КОНФИГУРАЦИЯ

			Маска записи
Адрес ведомого устройства	Период связи (мс)	1	от 0 до 255
Упорядоченный опрос (мс)	Начальный адрес диагностики интерфейса MODBUS	100	от 0 до 3600000
Область диагностики преобразования	Начальный адрес считывания данных MODBUS	-	от 0 до 2147483640
Конфигурация	Описание	Значение по умолчанию	Опции
Считывание начального адреса данных	Начальный адрес считывания данных MODBUS	1	от 1 до 65536
Считывание размера данных	Количество данных считывания MODBUS	-	В зависимости от используемой функции
Считывание переменной IEC	Начальный адрес считывания переменных (%I)	-	от 0 до 2147483647
Запись начального адреса данных	Начальный адрес данных записи MODBUS	1	от 1 до 65536
Запись размера данных	Число данных записи MODBUS	-	В зависимости от используемой функции
Запись переменной IEC	Начальный адрес переменных записи (%Q)	-	от 0 до 2147483647
Маскирование записи переменных IEC	Начальный адрес переменных для записи маски (%Q)	-	от 0 до 2147483644

Таблица 83: Преобразование устройства

### Примечания:

Таблица преобразования устройств: количество настроек и значений, описанных в столбце «Опции» (Options), может меняться в зависимости от типа данных и функции MODBUS.

Адрес ведомого устройства: как правило, адрес 0 используется, когда сервер представляет собой шлюз MODBUS RTU или MODBUS RTU через TCP, и он же транслирует запрос всем сетевым устройствам. При использовании адреса 0 клиент ожидает ответа, и его использование служит только для записи команд. Помимо этого, в соответствии со стандартом MODBUS, диапазон допустимых адресов для ведомых устройств составляет от 0 до 247, а адреса от 248 до 255 являются зарезервированными.

Упорядоченный опрос: данный параметр указывает на то, как часто должен выполняться обмен данными, установленный для данного отношения. По окончании обмена данными, время ожидания будет равно значению, заданному значению упорядоченного опроса, после чего как можно скорее будет выполнен новый обмен данными.

Область диагностики преобразований: данное поле ограничено размером адресуемой памяти выходных переменных (%Q) на центральном процессоре (ЦП), который можно найти в разделе «Память». Конфигурируемая диагностика отношений MODBUS описана в таблице 57.

Размер данных для считывания и записи: подробности о размере данных, поддерживаемых каждой функцией, описаны в примечаниях раздела «Конфигурация запросов - Настройки символьного преобразования».

### ВНИМАНИЕ:

При обращении к памяти данных связи между устройствами с различным порядком следования байтов многобайтного объекта в памяти или при передаче (эндианизм) ((Little-Endian (порядок байтов от младшего к старшему) и Big-Endian (порядок байтов от старшего

## 5. КОНФИГУРАЦИЯ

к младшему)) может произойти инверсия данных при считывании/записи. В этом случае пользователь должен скорректировать данные в приложении.

Считывание переменной IEC: если тип данных MODBUS – *Coil* («*Регистр флагов*») либо *Input Status* («*Состояние входов*») (бит), то начальный адрес переменных считывания IEC будет иметь, например, формат *%IX10.1*. Тем не менее, же тип данных MODBUS – *Holding Register* («*Регистр временного хранения*») или *Input Register* («*Регистр входов*») (16 бит), то начальный адрес переменных считывания IEC будет иметь формат *%IW*. Данное поле ограничено размером памяти адресуемых входных переменных (*%I*) в центральном процессоре (ЦП), который можно найти в разделе «*Память*».

Запись переменной IEC: если тип данных MODBUS – *Coil* («*Регистр флагов*») (бит), то начальный адрес переменных записи IEC будет иметь, например, формат *%QX10.1*. Если же тип данных MODBUS – *Holding Register* («*Регистр временного хранения*») (16 бит), то начальный адрес записи переменных IEC будет иметь формат *%QW*. Данное поле ограничено размером адресуемой памяти выходных переменных (*%Q*) в центральном процессоре (ЦП), который можно найти в разделе «*Память*».

Маскировка записи переменных IEC: функция «*Маскирование регистра записи*» (*Mask Write Register*) (FC 22) использует логику между уже записанным значением и двумя словами, которые конфигурируются в данном поле, используя *%QW(0)* для маски AND («*логическая операция И*») и *%QW(2)* для маски OR («*логическая операция ИЛИ*»); что позволяет пользователю работать с этим словом. Данное поле ограничено размером адресуемой памяти выходных переменных (*%Q*) каждого центрального процессора (ЦП), который можно найти в разделе «*Память*».

Значение по умолчанию: заводское значение по умолчанию для полей «*Область диагностики преобразований*» (*Mapping Diagnostics Area*), «*Считывание переменной IEC*» (*Read IEC Variable*), «*Запись переменной IEC*» (*Write IEC Variable*) и «*Маскировка записи переменных IEC*» (*Mask Write IEC Variables*) установить нельзя, так как создание отношения может быть выполнено в любой момент при разработке приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (*%Q*), которое пока не используется. Для полей «*Размер считываемых данных/данных записи*» (*Read/Write Data Size*) нельзя установить заводское значение, поскольку они будут изменяться в зависимости от выбранного типа данных MODBUS.

### ВНИМАНИЕ:

В отличие от других задач приложения, при достижении метки при отладке MainTask выполнение задачи экземпляра клиента MODBUS Ethernet или любой другой задачи MODBUS прекращается в тот момент, когда она пытается произвести запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока MainTask не запущена.

### 5.4.5.3. Запуск отношения клиента MODBUS (MODBUS Client) в ациклической форме

Для запуска отношения клиента MODBUS (MODBUS Client) в ациклической форме предлагается следующий метод, который может быть реализован простым способом в прикладной программе пользователя:

- Определите максимальное время упорядоченного опроса отношений;
- Держите отношение отключенным при обычных условиях;
- Включите отношение в нужный момент выполнения;
- Дождитесь подтверждения окончания выполнения отношения и в этот момент снова отключите его.

### 5.4.6. Сервер MODBUS Ethernet (MODBUS Ethernet Server)

Данный протокол доступен для всех центральных процессоров (ЦП) серии Nexto на его каналах сети Ethernet. При выборе данной опции в MasterTool IEC XE центральный процессор становится коммуникационным сервером MODBUS, что позволяет устанавливать связь с устройствами клиента MODBUS. Данный протокол доступен только в том случае, если центральный процессор находится в режиме исполнения (режим «*Выполнение*» (*Run*)).

Существует два способа конфигурирования данного протокола. Первый - через *прямое представление* (*%Q*), при котором переменные определяются своим адресом. Второй - через *символьное преобразование*, при котором переменные определяются по имени.

Процедура вставки экземпляра протокола подробно описана в Руководстве пользователя MasterTool IEC XE – MU299609.



## 5. КОНФИГУРАЦИЯ

### 5.4.6.1. Конфигурация протокола Ethernet сервера MODBUS (MODBUS Server) для символьного преобразования

Для конфигурирования данного протокола с использованием *символьных преобразований* необходимо выполнить следующие действия:

- Сконфигурируйте общие параметры протокола сервера MODBUS, такие как: порт TCP, выбор протокола, IP-фильтры для считывания и записи (доступны посредством кнопки «Конфигурация фильтров») и время обмена данными (доступны посредством кнопки «Расширенные конфигурации сервера»).
- Добавление и конфигурирование преобразований MODBUS с указанием имени переменной, типа данных, начального адреса данных и размера данных.

Описание каждой конфигурации приведено далее в данном разделе.

#### 5.4.6.1.1. Общие параметры протокола сервера MODBUS (MODBUS Server) - Конфигурация через символьное преобразование

Общие параметры, представленные на начальном экране протокола MODBUS (рисунок, приведенный ниже), определяются следующим образом.

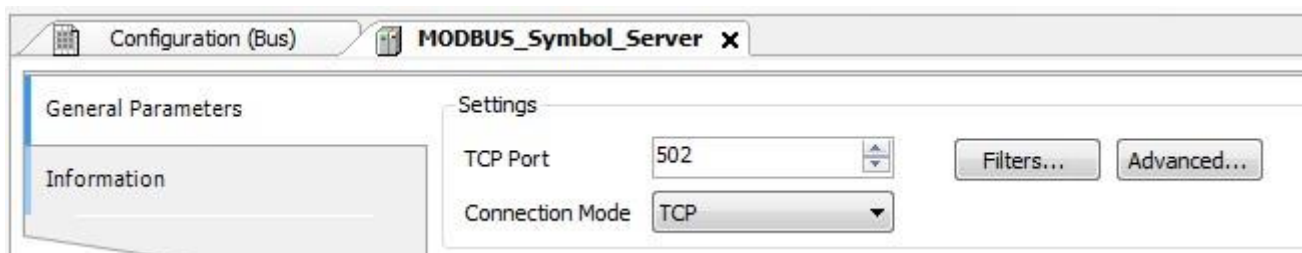


Рисунок 70: Экран конфигурации общих параметров сервера MODBUS

Конфигурация	Описание	По умолчанию	Опции
Порт TCP	Порт TCP	502	от 2 до 65534
Режим соединения	Выбор протокола	TCP	RTU через TCP TCP

Таблица 84: Общие конфигурации сервера MODBUS

#### Примечания:

Порт TCP: если в один интерфейс сети Ethernet добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные порты TCP. Некоторые порты TCP, среди перечисленных выше, зарезервированы и, таким образом, не могут быть использованы. К ним относятся: 80, 8080, 1217, 1740, 1741, 1742, 1743 и 11740.

Настройки, присутствующие на кнопке «*Фильтры...*» (*Filters...*), описанные в таблице ниже, относятся к фильтрам коммуникаций TCP:

Конфигурация	Описание	Значение по умолчанию	Опции
IP-адрес записи фильтра	Задаёт диапазон IP-адресов с доступом на запись в переменные, объявленные в интерфейсе MODBUS.	0.0.0.0	от 0.0.0.0 до 255.255.255.255
Запись маски фильтра	Указание маски подсети в сочетании с параметром IP-фильтра для записи.	0.0.0.0	от 0.0.0.0 до 255.255.255.255

## 5. КОНФИГУРАЦИЯ

Считывание фильтра IP-адреса	Указание диапазона IP-адресов с доступом на чтение в переменных, объявленных в интерфейсе MODBUS.	0.0.0.0	от 0.0.0.0 до 255.255.255.255
Считывание маски фильтра	Указание маски подсети подсети в сочетании с параметром IP-фильтра для чтения.	0.0.0.0	от 0.0.0.0 до 255.255.255.255

Таблица 85: IP-фильтры

Примечание:

Фильтры: фильтры используются для установления диапазона IP-адресов, которые имеют права доступ на запись или считывание отношений MODBUS, будучи индивидуально сконфигурированными. Критерии разрешения выполняются с помощью логической операции «И» (AND) между маской фильтра записи и IP-адресом клиента. Если результат совпадает с IP-адресом фильтра записи, то клиент имеет право на запись. Например, если IP-адрес фильтра записи = 192.168.15.0, а маска фильтра записи = 255.255.255.0, то право на запись имеют только клиенты с IP-адресом = 192.168.15.x. Аналогичная процедура применяется в параметрах «Считывание фильтра» (*Read Filter*) для определения прав на считывание.

Время связи по протоколу сервера MODBUS (MODBUS server), определяемое по кнопке «Дополнительно...» (*Advanced...*) экрана конфигурации, подразделяется на «Цикл задач» (*Task Cycle*) и «Блокировка по превышению лимита времени неактивности соединения» (*Connection Inactivity Time-out*).

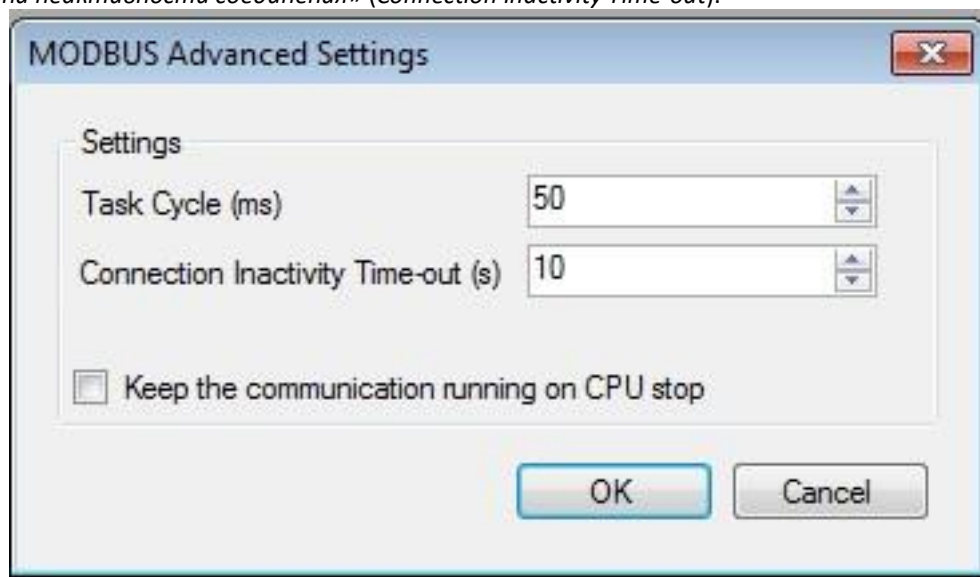


Рисунок 71 Экран конфигурации дополнительных настроек сервера MODBUS

Конфигурация	Описание	По умолчанию	Опции
Цикл задач (мс)	Время выполнения экземпляра в рамках цикла, без учета собственного времени выполнения	50	от 5 до 100
Блокировка по превышению лимита времени неактивности соединения (с)	Максимальное время простоя между клиентом и сервером до закрытия соединения сервером	10	от 10 до 3600

## 5. КОНФИГУРАЦИЯ

Поддерживайте связь при останове центрального процессора (ЦП)	Разрешить ведомому устройству MODBUS Symbol (MODBUS Symbol Slave) работать, пока центральный процессор (ЦП) находится в состоянии STOP («Останов») или после точки контрольного прерывания	Не отмечено	Отмечено или Не отмечено
---	--	-------------	--------------------------

Таблица 86: Расширенные конфигурации сервера MODBUS

### Примечания:

Цикл задач: пользователь должен быть осторожен при изменении данного параметра, поскольку он напрямую влияет на время отклика, объем данных для сканирования и, главным образом, на баланс ресурсов центрального процессора (ЦП) между коммуникациями и другими задачами.

Блокировка по превышению лимита времени неактивности соединения: данный параметр был создан для того, чтобы избежать достижения максимального количества TCP-соединений, представляя, что неактивные соединения остаются открытыми ввиду самых разных проблем. Он показывает, как долго соединение (клиентское или серверное) может оставаться открытым без использования (без обмена коммуникационными сообщениями). Если указанное время не достигнуто, соединение закрывается, освобождая вход в таблицу соединений.

### 5.4.6.1.2. Диагностика сервера MODBUS (MODBUS Server) - Конфигурация через символьное преобразование

Диагностика и команды протокола сервера MODBUS (MODBUS server), сконфигурированные либо по символическому преобразованию, либо по прямому представлению, хранятся в переменных типа `T_DIAG_MODBUS_ETH_SERVER_1`, а преобразование по прямому представлению - в 4 байтах и 8 словах, которые описаны в таблице, приведенной ниже (где «n» - сконфигурированное значение в поле «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area)):

Переменная прямого представления	Диагностическая переменная <code>T_DIAG_MODBUS_ETH_SERVER_1.*</code>	Размер	Описание
Диагностические биты:			
%QX(n).0	tDiag. bRunning	BIT	Сервер работает.
%QX(n).1	tDiag. bNotRunning	BIT	Сервер не запущен (см. бит bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	Бит bNotRunning был включен, поскольку работа сервера была прервана пользователем через задающий бит.
%QX(n).3	tDiag. bConfigFailure	BIT	Прекращенная диагностика.
%QX(n).4	tDiag. bRXFailure	BIT	Прекращенная диагностика.
%QX(n).5	tDiag. bTXFailure	BIT	Прекращенная диагностика.
%QX(n).6	tDiag.	BIT	Прекращенная диагностика.

## 5. КОНФИГУРАЦИЯ

	bModuleFailure		
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Зарезервировано.
%QB(n+1)	byDiag_1_reserved	BYTE	Зарезервировано.
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS _ETH_SERVER_1.*	Размер	Описание
Биты команд, перезапускаемые автоматически:			
%QX(n+2).0	tCommand. bStop	BIT	Останов сервера.
%QX(n+2).1	tCommand. bRestart	BIT	Перезапуск сервера.
%QX(n+2).2	tCommand. bResetCounter	BIT	Статистика диагностики перезапуска (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.
Статистика коммуникаций:			
%QW(n+4)	tStat. wActiveConnections	WORD	Количество установленных соединений между клиентом и сервером (от 0 до 64).
%QW(n+6)	tStat. wTimeoutClosedConnections	WORD	Счетчик соединений между клиентом и сервером, прерванных после блокировки по превышению лимита времени бездействия (от 0 до 65535).
%QW(n+8)	tStat. wClientClosedConnections	WORD	Счетчик соединений прерван по запросу клиента (от 0 до 65535).
%QW(n+10)	tStat. wRXFrames	WORD	Счетчик кадров сети Ethernet, полученных сервером. Кадр сети Ethernet может содержать более одного запроса (от 0 до 65535).
%QW(n+12)	tStat. wRXRequests	WORD	Запросы, полученные счетчиком сервера и получивших нормальный отклик (от 0 до 65535).

## 5. КОНФИГУРАЦИЯ

%QW(n+14)	tStat. wTXExceptionResponses	WORD	Запросы, полученные счетчиком сервера от ответ на которые был получен в форме кодов исключений (от 0 до 65535). Коды исключений перечислены ниже: 1: код функции (FC) является законным, но не поддерживается. 2: связь не найдена в этих данных MODBUS. 3: недопустимое значение для адреса. 128: ведущее устройство/клиент не имеет права на считывание или запись. 129: связь MODBUS отключена.
%QW(n+16)	tStat. wRXIllegalRequests	WORD	Счетчик недействительных запросов (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Зарезервировано.

Таблица 87: Диагностика сервера MODBUS

Примечание:

Счетчики: Все счетчики диагностики сервера MODBUS Ethernet (MODBUS Ethernet Server) возвращаются к нулю при превышении предельного значения 65535.

### 5.4.6.1.3. Конфигурация преобразований – Конфигурация через символьное преобразование

Конфигурация отношений протокола MODBUS, показанная на рисунке, приведенном ниже, соответствует параметрам, описанным в таблице ниже:

Mappings

	Value Variable	Data Type	Data Start Address	Absolute Data Start Address	Data Size	Data Range
*						

Рисунок 72: Экран преобразования данных сервера MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Переменная значения	Имя символической переменной	-	Имя переменной, объявленной в программе или списке глобальных переменных (GVL)
Тип данных	Тип данных MODBUS	-	Регистр флагов Состояние входа Регистр временного хранения Регистр входа
Начальный адрес данных	Начальный адрес данных MODBUS	-	от 1 до 65536
Абсолютный начальный адрес данных	Начальный адрес абсолютных данных	-	-

## 5. КОНФИГУРАЦИЯ

	Modbus в качестве его типа		
Размер данных	Размер данных MODBUS	-	от 1 до 65536
Диапазон данных	Сконфигурированный адрес диапазона данных	-	-

Таблица 88: Конфигурация преобразований MODBUS Ethernet

Примечания:

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

Тип данных: данное поле используется для указания типа данных, используемого 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 временного хранения с адресом 5 имеет абсолютный адрес 400005. Данное поле доступно только для чтения и служит для конфигурирования клиента/ведущего устройства MODBUS, который будет взаимодействовать с данным устройством. Значения зависят от базового адреса (смещения) каждого типа данных и допустимого адреса MODBUS для каждого типа данных.

Размер данных: значение «Размер данных» (Data Size) задает максимальный объем данных, к которому может обращаться отношение MODBUS с начального адреса. Таким образом, для считывания непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном отношении. Данное поле изменяется в зависимости от конфигурируемого типа данных MODBUS.

Диапазон данных: поле, доступное только для чтения, сообщает о диапазоне адресов, который используется данным преобразованием. Он образуется суммой полей «Начальный адрес данных» (Data Start Address) и «Размер данных» (Data Size). Не может быть перекрытия диапазона с другими преобразованиями того же типа данных.

**ВНИМАНИЕ:**

В отличие от других задач приложения, при достижении метки при отладке MainTask выполнение задачи экземпляра сервера MODBUS Ethernet или любой другой задачи MODBUS прекращается в тот момент, когда она пытается произвести запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока MainTask не запущена.

### 5.4.6.2. Конфигурирование протокола Ethernet сервера MODBUS (MODBUS Server) через прямое представление (%Q)

Для конфигурирования данного протокола с использованием *прямого представления (%Q)* пользователю необходимо выполнить следующие шаги:

- Сконфигурируйте общие параметры протокола сервера MODBUS Server (MODBUS Server), такие как: время связи, адрес и переменные прямого представления (%Q) для приема отношения диагностики и управления.
- Добавьте и сконфигурируйте отношения MODBUS, указав тип данных MODBUS, переменные прямого представления (%Q) для приема/записи данных, а также количество передаваемых данных.

Описание каждой конфигурации приведено ниже в данном разделе.

#### 5.4.6.2.1. Общие параметры протокола сервера MODBUS (MODBUS Server) - конфигурирование через прямое представление (%Q)

Общие параметры, расположенные на главном экране конфигурации протокола MODBUS (рисунок, приведенный ниже), определяются следующим образом:

## 5. КОНФИГУРАЦИЯ

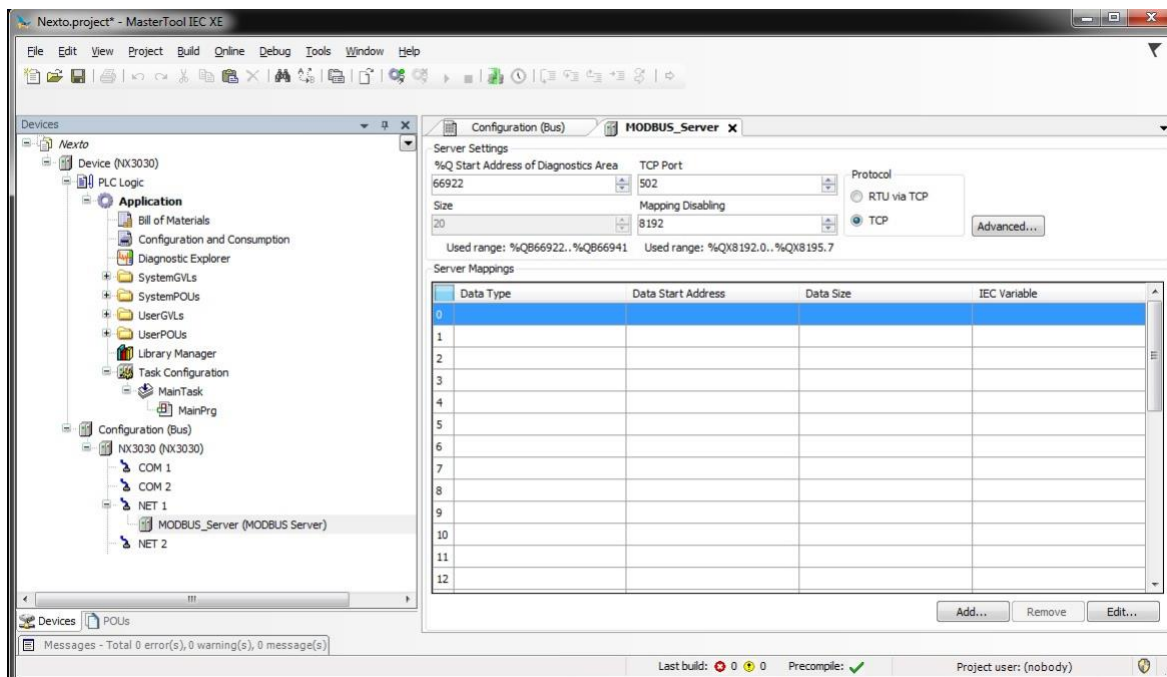


Рисунок 73: Экран настройки сервера MODBUS

Порт TCP, протокол и переменные прямого представления (%Q) для отношений управления и диагностики:

Конфигурация	Описание	Значение по умолчанию	Опции
Начальный адрес области диагностики %Q	Начальный адрес переменных диагностики	-	от 0 до 2147483628
Размер	Размер диагностики	20	Выключено для редактирования
Порт TCP	Порт TCP	502	от 2 до 65534
Выключение преобразования	Начальный адрес, используемый для выключения отношений MODBUS	-	от 0 до 2147483644
Протокол	Выбор протокола	TCP	RTU через TCP TCP

Таблица 89: Настройки для управления отношениями и диагностикой

Примечания:

Начальный адрес %Q области диагностики: данное поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого центрального процессора, который можно найти в разделе «Память».

Порт TCP: если в один интерфейс сети Ethernet добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные порты TCP. Некоторые порты TCP, среди перечисленных выше, зарезервированы и поэтому не могут быть использованы. К ним относятся: 80, 8080, 1217, 1740, 1741, 1742, 1743 и 11740.

Выключение преобразования: состоит из 32 битов и используется для отключения по отдельности 32 отношений MODBUS, сконфигурированных в пространстве «Преобразований сервера» (*Server Mappings*). Отношение отключено, если соответствующий бит равен 1, в противном случае преобразование включено. Данное поле ограничено размером

## 5. КОНФИГУРАЦИЯ

адресуемой памяти выходных переменных (%Q) каждого центрального процессора (ЦП), который можно найти в разделе «Память».

Значение по умолчанию: для полей «Начальный адрес %Q области диагностики» (%Q Start Address of Diagnostics Area) и «Выключение преобразования» (Mapping Disabling) не может быть установлено заводское значение по умолчанию, поскольку создание экземпляра протокола может быть проведено в любой момент разработки приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), которое еще не используется.

Время связи по протоколу сервера MODBUS (MODBUS Server), указанное на кнопке «Дополнительно...» (Advanced...) экрана конфигурации, подразделяется на «Цикл задач» (Task Cycle) (мс) и «Блокировка по превышению лимита времени неактивности соединения» (Connection Inactivity Time-out) (с). Более подробная информация приведена в разделе «Общие параметры протокола сервера MODBUS (MODBUS Server)- Конфигурация через символическое преобразование».

Диагностика и команды протокола MODBUS описаны в таблице 87.

### 5.4.6.2.2. Конфигурация преобразований - Конфигурация через прямое представление (%Q)

Настройки отношений MODBUS, представленные на приведенных ниже рисунках, соответствуют параметрам, описанным в таблице ниже:

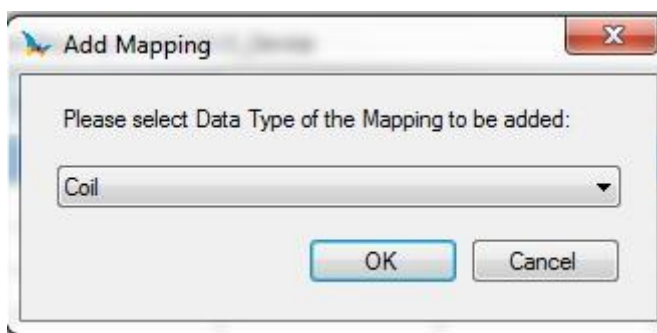


Рисунок 74: Тип данных MODBUS

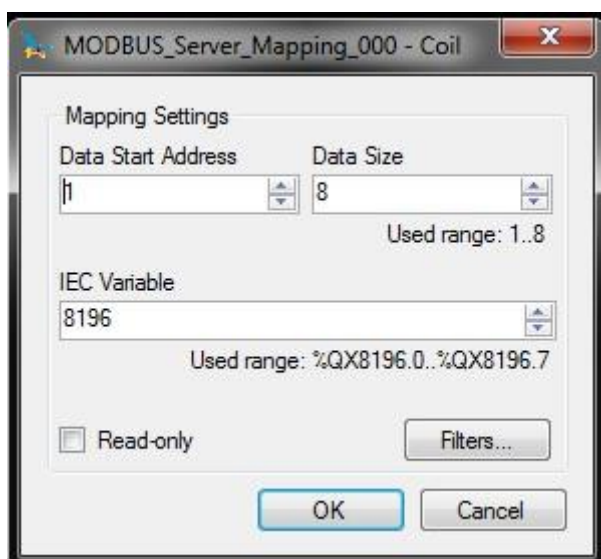


Рисунок 75: Функция сервера MODBUS

Конфигурация	Описание	По умолчанию	Опции
--------------	----------	--------------	-------



## 5. КОНФИГУРАЦИЯ

Тип данных	Тип данных MODBUS	Регистр флагов	Регистр флагов (1 бит) Регистр временного хранения (16 битов) Состояние входа (1 бит) Регистр входа (16 битов)
Адрес начала данных	Адрес начальных данных MODBUS	1	от 1 до 65536
Размер данных	Количество данных MODBUS	8	От 1 до 65536 (регистр временного хранения и регистр входа) От 8 до 65536 (состояние регистра флагов и входа)
Переменная IEC	Начальный адрес переменных (%Q)	-	от 0 до 2147483647
Только чтение	Разрешить только считывание	Выключено	Включено или выключено

Таблица 90: Преобразования сервера

### Примечания:

Опции: значения, записанные в столбце «Опции» (*Options*), могут отличаться в зависимости от конфигурации данных протокола MODBUS.

Размер данных: значение «Размер данных» (*Data Size*) задает максимальный объем данных, к которому может обратиться отношение MODBUS с начального адреса. Таким образом, для считывания непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном отношении. Данное поле изменяется в зависимости от установленного типа данных MODBUS, т.е. при выборе *Coil* («Регистр флагов») либо *Input Status* («Состояние входов») поле «Размер данных» (*Data Size*) должно быть числом, кратным 8. Важно также следить за тем, чтобы максимальное значение не превышало размер адресуемой выходной памяти, а приписываемые значения не совпадали с уже использованными в процессе применения.

### ВНИМАНИЕ:

При обращении к памяти данных связи между устройствами с различным порядком следования байтов многобайтного объекта в памяти или при передаче (эндианизм) (*Little-Endian* (порядок байтов от младшего к старшему) и *Big-Endian* (порядок байтов от старшего к младшему)) может произойти инверсия данных при считывании/записи. В этом случае пользователь должен скорректировать данные в приложении.

Переменная IEC: если тип данных MODBUS – *Coil* («Регистр флагов») или *Input Status* («Состояние входов») (бит), то начальный адрес переменных IEC будет в формате *%QX10.1*. Если же тип данных MODBUS – *Holding Register* («Регистр временного хранения») или *Input Register* («Регистр входов») (16 бит), то начальный адрес переменных IEC будет в формате *%QW*. Данное поле ограничено объемом памяти адресуемых выходных переменных (%Q) от каждого центрального процессора, который можно посмотреть в разделе «Память».

Только чтение: если данная опция включена, она позволяет ведущему устройству связи только считывать данные переменных. Выполнение записи не разрешено. Данная опция действительна только для функций записи.

По умолчанию: для поля «Переменная IEC» (*IEC Variable*) нельзя задать значение по умолчанию, поскольку создание экземпляра протокола может быть произведено в любой момент разработки приложения, что заставит программное обеспечение MasterTool IEC XE самостоятельно выделить значение из еще не используемого диапазона выходных переменных прямого представления (%Q). Для поля «Размер данных» (*Data Size*) нельзя определить значение по умолчанию, поскольку оно будет изменяться в зависимости от выбранного типа данных MODBUS.

Настройки, присутствующие на кнопке «Фильтры...» (*Filters...*), описанные в таблице ниже, относятся к фильтрам коммуникаций TCP:

## 5. КОНФИГУРАЦИЯ

Конфигурация	Описание	Значение по умолчанию	Опции
IP-адрес записи фильтра	Задаёт диапазон IP-адресов с доступом на запись в переменные, объявленные в интерфейсе MODBUS.	0.0.0.0	от 0.0.0.0 до 255.255.255.255
Запись маски фильтра	Указание маски подсети в сочетании с параметром IP-фильтра для записи.	0.0.0.0	от 0.0.0.0 до 255.255.255.255
IP-адрес считывания фильтра	Указание диапазона IP-адресов с доступом на чтение в переменных, объявленных в интерфейсе MODBUS.	0.0.0.0	от 0.0.0.0 до 255.255.255.255
Конфигурация	Описание	Значение по умолчанию	Опции
Считывание маски фильтра	Указание маски подсети в сочетании с параметром IP-фильтра для чтения.	0.0.0.0	от 0.0.0.0 до 255.255.255.255

Таблица 91: IP-фильтры

Примечание:

Фильтры: фильтры используются для установления диапазона IP-адресов, которые имеют права доступ на запись или считывание отношений MODBUS, будучи индивидуально сконфигурированными. Критерии разрешения выполняются с помощью логической операции «И» (AND) между маской фильтра записи и IP-адресом клиента. Если результат совпадает с IP-адресом фильтра записи, то клиент имеет право на запись. Например, если IP-адрес фильтра записи = 192.168.15.0, а маска фильтра записи = 255.255.255.0, то право на запись имеют только клиенты с IP-адресом = 192.168.15.x. Аналогичная процедура применяется в параметрах «Считывание фильтра» (*Read Filter*) для определения прав на считывание.

В ранее определенных отношениях максимальный размер данных MODBUS может составлять 65536 (максимальное значение, заданное в поле «Размер данных» (*Data Size*)). Тем не менее запрос, поступающий в сервера MODBUS Ethernet (MODBUS Ethernet Server), должен быть обращен к подгруппе данного отображения, и данная группа должна иметь, максимум, размер данных, зависящий от кода функции, который определен ниже:

- Считывание регистров флагов (FC 1): 2000
- Считывание состояния входа (FC 2): 2000
- Считывание регистров временного хранения (FC 3): 125
- Считывание регистров входов (FC 4): 125
- Запись одного регистра флагов (FC 5): 1
- Запись одного регистра временного хранения (FC 6): 1
- Форсирование нескольких регистров флагов (FC 15): 1968
- Запись регистров временного хранения (FC 16): 123
- Маскирование регистра записи (FC 22): 1
- Считывание/запись регистров временного хранения (FC 23):
  - Считывание: 121
  - Запись: 121

ВНИМАНИЕ:

## 5. КОНФИГУРАЦИЯ

В отличие от других задач приложения, при достижении метки очистки MainTask выполнение задачи экземпляра сервера MODBUS Ethernet или любой другой задачи MODBUS прекращается в тот момент, когда она пытается произвести запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока MainTask не запущена.

### 5.4.7. Сервер OPC DA (OPC DA Server)

Связь с центральными процессорами (ЦП) серии Nexto можно осуществлять посредством технологии OPC DA (*Доступ к данным коммуникации открытой платформы*). Данная открытая коммуникационная платформа была разработана как стандарт в области промышленных коммуникаций. Будучи основанной на архитектуре клиент/сервер, она предоставляет ряд преимуществ при разработке проектов и взаимодействии с системами автоматизации.

Очень распространенной аналогией для описания технологии OPC DA является аналогия с принтером. При правильном подключении компьютеру требуется драйвер для взаимодействия с оборудованием. Аналогичным образом OPC помогает обеспечить интерфейс между системой контроля и данными в области памяти на ПЛК.

При разработке проекта настройка связи и обмена информацией между системами предельно проста при использовании технологии OPC DA. При использовании других драйверов, основанных на адресах, необходимо создать таблицы, связывающие метки из системы наблюдения с переменными программируемого (логического) контроллера. При изменении областей данных в ходе проекта необходимо перераспределить переменные и создать новые таблицы, содержащие отношения между информацией на ПЛК и автоматизированной системой связи и управления, установленной на объектах (SCADA).



Рисунок 76: Архитектура OPC DA

На рисунке, приведенном выше, показана архитектура связи системы SCADA (автоматизированная система управления технологическим процессом) и ПЛК в проектах автоматизации. Все роли, присутствующие в коммуникации, явно выражены на данном рисунке независимо от оборудования, на котором она осуществляется, поскольку они могут выполняться как на одном и том же оборудовании, так и на разных. Описание каждой из ролей данной архитектуры приведено в таблице ниже.

Роль	Описание
Уровень программируемых (логических) контроллеров и полевых устройств	В полевых устройствах и ПЛК хранится информация о состоянии работы и управлении установкой. Система SCADA (автоматизированная система управления технологическим процессом) получает доступ к информации на данных устройствах и сохраняет ее на сервере SCADA (автоматизированная система управления технологическим процессом), чтобы клиенты SCADA (автоматизированная система управления технологическим процессом) могли обращаться к ней во время работы установки.
Сеть сбора данных	Сеть сбора данных представляет собой место, куда поступают запросы на получение данных, собранных полевыми устройствами, чтобы запросить данные, собранные полевыми устройствами.
Шлюз для связи с ПЛК	Шлюз обеспечивает связь между сервером OPC DA (OPC DA Server) и ПЛК серии Nexto. Шлюз всегда должен находиться в той же подсети, что и ПЛК, как описано в главе «Настройки связи» Руководства пользователя MasterTool IEC XE - MU299609.
Модуль сервера OPC	Сервер OPC DA (OPC DA Server) представляет собой модуль, отвечающий за прием запросов OPC DA и их преобразование в обмен данными с полевыми устройствами.
Роль	Описание
Клиент OPC модуля устройства	Модуль устройства клиента OPC отвечает за запросы к серверу OPC DA (OPC DA Server) по протоколу OPC DA. Собранные данные сохраняются в базе данных сервера SCADA (автоматизированная система управления технологическим процессом).
Уровень сервера SCADA (автоматизированная система управления технологическим процессом)	Сервер SCADA отвечает за подключение к различным коммуникационным устройствам и хранение собранных ими данных в базе данных, чтобы к ним могли обращаться клиенты SCADA (автоматизированная система управления технологическим процессом).
Сеть контроля	Сеть наблюдения (автоматизированная система управления технологическим процессом) представляет собой сеть, через которую клиенты SCADA (автоматизированная система управления технологическим процессом) подключаются к серверам SCADA (автоматизированная система управления технологическим процессом). В топологии, где нет нескольких Клиентов или Сервер и Клиент установлены на одном оборудовании, такой сети не существует.

<p>Уровень клиента SCADA (автоматизированная система управления технологическим процессом)</p>	<p>Клиенты SCADA (автоматизированная система управления технологическим процессом) отвечают за направление запросов к серверам SCADA (автоматизированная система управления технологическим процессом) необходимых данных для отображения на экране, на котором происходит работа установки. Через них можно выполнять считывания и записи по данным, хранящимся в базе данных сервера SCADA (автоматизированная система управления технологическим процессом).</p>
--	---

Таблица 92: Описание ролей в архитектуре сервера OPC DA

Связь между метками (тегами) в системе контроля и данными процесса в переменных контроллера абсолютно прозрачна. Это означает, что при изменении областей данных в ходе развития проекта нет необходимости переделывать связи между информацией на ПЛК и SCADA (автоматизированная система управления технологическим процессом): достаточно использовать новую переменную, предоставленную ПЛК, в системах, запрашивающих эти данные.

Использование OPC обеспечивает большую производительность и возможность взаимодействия с системами SCADA (автоматизированная система управления технологическим процессом). Это способствует сокращению времени разработки приложений и расходов на их техническое обслуживание. Оно даже позволяет вводить новые данные в коммуникацию в упрощенной форме, а также с большей гибкостью и совместимостью между системами автоматизации, благодаря тому, что это открытый стандарт.

Установка сервера OPC DA (OPC DA Server) производится вместе с установкой MasterTool IEC XE, а его настройки выполняются внутри инструмента. Следует отметить, что OPC доступен только через локальный интерфейс сети Ethernet центральных процессоров (ЦП) Nexto. Модули расширения сети Ethernet не поддерживают данную функциональность.

#### 5.4.7.1. Создание проекта для связи с OPC DA

В отличие от связи с такими драйверами, как MODBUS и PROFIBUS DP, для настройки связи с OPC необходимо лишь правильно задать узел и указать, какие переменные будут использоваться в связи. Существует два способа указать, какие переменные проекта будут доступны в сервере OPC DA (OPC DA Server). В обоих случаях необходимо добавить в приложение объект «Символьная конфигурация» (*Symbol Configuration*), если такового нет. Чтобы добавить его, щелкните правой кнопкой мыши на объекте «Приложение» (*Application*) и выберите опцию.

##### ВНИМАНИЕ:

Переменные, указанные в объектах *IoConfig\_Globals*, *IoConfig\_Application\_Mappings* и *IoConfig\_Global\_Mappings*, используются внутри системы для управления входом/выходом и не должны использоваться пользователем.

##### ВНИМАНИЕ:

В дополнение к переменным, объявленным в структурной единице программы (POU) языка последовательностных функциональных схем (SFC), показаны также некоторые неявно созданные переменные. Для каждого создаваемого шага создается переменная типа *IecSfc.SFCStepType*, в которой можно отслеживать состояния шага, а именно: активен он или нет, а также время, в течение которого он активен, как это предусмотрено стандартом МЭК 61131-1 (IEC 61131-1). Для каждого перехода создается переменная типа *BOOL*, которая определяет, является ли переход истинным или ложным. Данные переменные отображаются в объекте «Символьная Конфигурация» (*Symbol Configuration*), к которому может быть предоставлен доступ к клиенту OPC.

## 5. КОНФИГУРАЦИЯ

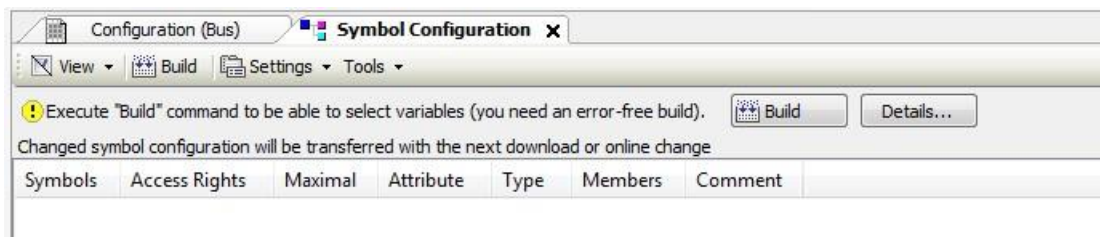


Рисунок 77: Объект «Символьная конфигурация» (Symbol Configuration)

В приведенной ниже таблице ниже представлены описания полей экрана объекта «Символьная конфигурация» (Symbol Configuration).

Поле	Описание
Символы	Идентификатор переменной, который будет предоставлен серверу OPC DA (OPC DA Server).
Права доступа	Указывает возможный уровень прав доступа в объявленном символе. Если данный столбец не используется, то он остается пустым, а уровень прав доступа - максимальным. В противном случае уровень права доступа может быть изменен щелчком мыши над этим полем. Возможны следующие варианты:  Только чтение Только запись Считывание и запись
Максимальный	Указывает максимальный уровень прав доступа, который можно назначить переменной. Символы имеют те же значения, что и в «Правах доступа». Изменить его невозможно, и на это указывает наличие либо отсутствие атрибута 'symbol' («символ»).
Атрибут	Указывает, используется ли атрибут 'symbol' («символ») при объявлении переменной. Если атрибут не используется, то данный столбец остается пустым. Для случаев, когда атрибут используется, логика работы будет следующей:  Атрибут 'symbol' := столбец показывает 'read' Атрибут 'symbol' := столбец показывает 'write' Атрибут 'symbol' := столбец показывает 'readwrite'
Тип	Тип данных объявленной переменной.
Члены	Если типом данных является структура, то в этом столбце включается кнопка. Щелчок на кнопке позволяет выбрать, какие элементы данной структуры будут предоставлены серверу OPC DA (OPC DA Server).
Комментарий	Комментарий к переменной, вставляемый в структурную единицу программы (POU) либо список глобальных переменных (GVL), где переменная была объявлена. Чтобы комментарий к переменной отображался здесь, он должен быть введен на одну строку раньше переменной в редакторе в текстовом режиме или в столбце комментариев в табличном режиме.

Таблица 93: Описание полей экрана объекта «Символьная конфигурация» (Symbol Configuration)

## 5. КОНФИГУРАЦИЯ

При изменении настроек проекта, например, добавлении или удалении переменных, необходимо выполнить команду «Компоновка» (*Build*), чтобы обновить список переменных. Данную команду необходимо выполнять до тех пор, пока не исчезнет сообщение, показанное на рисунке 77. После этого все имеющиеся в проекте переменные, независимо от того, объявлены ли они на структурных единицах программы (POU), в списках глобальных переменных (GVL) или в диагностике, будут показаны здесь и могут быть выбраны. Выбранные переменные будут доступны на сервере OPC DA (OPC DA Server) для доступа клиентов.

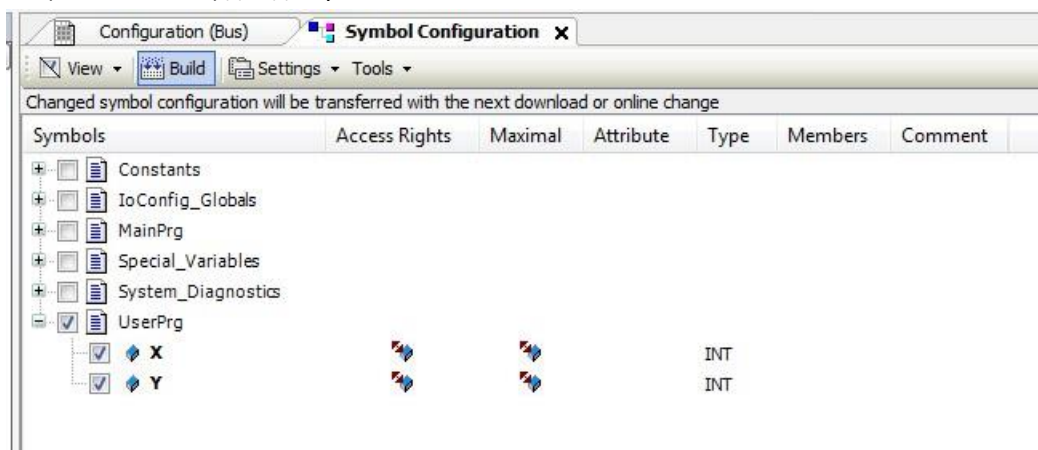


Рисунок 78: Выбор переменных в символьной конфигурации

После выполнения данной процедуры проект должен быть загружен в ПЛК, чтобы переменные были доступны для связи с сервером OPC DA (OPC DA Server). Если открыть экран конфигурации символов объекта и изменить любую из выбранных переменных, структурные единицы программы (POU) или списки глобальных переменных (GVL), то ее имя будет выделено красным цветом. Это может произойти, например, при удалении переменной или изменении значения атрибута.

Также можно установить, какие переменные будут доступны на сервере OPC DA (OPC DA Server), с помощью атрибута, вставляемого непосредственно в структурную единицу программы (POU) или списки глобальных переменных (GVL), где объявлены переменные. При наличии атрибута 'symbol' («символ») в объявлении переменной, а это может быть как перед определением имени структурной единицы программы (POU) или списка глобальных переменных (GVL), так и для каждой переменной в отдельности, эти переменные отправляются непосредственно в объект «Символьная конфигурация» (*Symbol Configuration*), с символом в столбце «Атрибут» (*Attribute*). В этом случае необходимо перед загрузкой проекта в центральный процессор выполнить команду «Компоновка» (*Build*) из объекта «Символьная конфигурация» (*Symbol Configuration*).

Для использования атрибута допустимы следующие синтаксисы:

- атрибут 'symbol' := 'none' – если значение атрибута равно 'none' («нет»), то переменные не будут доступны для сервера OPC DA (OPC DA Server) и не будут отображаться на экране объекта «Символьная конфигурация» (*Symbol Configuration*).
- атрибут 'symbol' := 'read' - если значение атрибута равно 'read' («считывание»), то переменные будут доступны серверу OPC DA (OPC DA Server) с правом доступа только на чтение.
- атрибут 'symbol' := 'write' - если значение атрибута равно 'write' («запись»), то переменные будут доступны серверу OPC DA (OPC DA Server) с правом доступа только на запись.
- атрибут 'symbol' := 'readwrite' – если значение атрибута равно 'readwrite' («чтение и запись»), то переменные будут доступны серверу OPC DA (OPC DA Server) с правом доступа на чтение и запись.

В следующем примере объявления переменных параметры переменных A и B позволяют серверу OPC DA (OPC DA Server) обращаться к ним с правами чтения и записи. Тем не менее, доступ к переменной C невозможен, а к переменной D можно обращаться с правами только на чтение.

```
{attribute 'symbol' := 'readwrite'}  
ПРОГРАММА UserPrg  
VAR
```

## 5. КОНФИГУРАЦИЯ

```
A: INT;  
B: INT;  
{атрибут 'symbol' := 'none'}  
C: INT;  
{атрибут 'symbol' := 'read'}  
D :INT;  
END_VAR
```

При определении переменной с типом, отличным от базовых типов, использование атрибута должно осуществляться внутри объявления данного испытываемого устройства (DUT), а не только в контексте, в котором создается переменная. Например, в случае, когда экземпляр испытываемого устройства (DUT) находится внутри структурной единицы программы (POU) или списка глобальных переменных (GVL), имеющего атрибут, он не будет влиять на логику работы элементов данного экземпляра испытываемого устройства (DUT). Необходимо будет применить тот же уровень прав доступа к объявлению испытываемого устройства (DUT).

### ВНИМАНИЕ:

Конфигурации символов, которые будут предоставляться сервером OPC DA, хранятся внутри проекта ПЛК. При изменении данных конфигураций необходимо загрузить приложение на ПЛК, чтобы можно было получить доступ к данным переменным.

### ВНИМАНИЕ:

При удалении переменной из проекта и ее загрузке в ПЛК, сняв флажок с объекта «Символьная конфигурация» (*Symbol Configuration*), переменная больше не может быть считана с помощью клиента OPC. Если переменная снова будет добавлена в проект под тем же именем и с тем же контекстом и вставлена на объект «Символьная конфигурация» (*Symbol Configuration*), то необходимо будет перезагрузить клиент OPC для обновления адресной ссылки переменной, которая будет создана в другой области памяти ПЛК.

### 5.4.7.2. Конфигурирование ПЛК на сервере OPC DA (OPC DA Server)

Конфигурирование ПЛК производится внутри MasterTool IEC XE с помощью опции, доступной в *диалоговом режиме («онлайн»)*. Необходимо запустить MasterTool IEC XE от имени администратора.

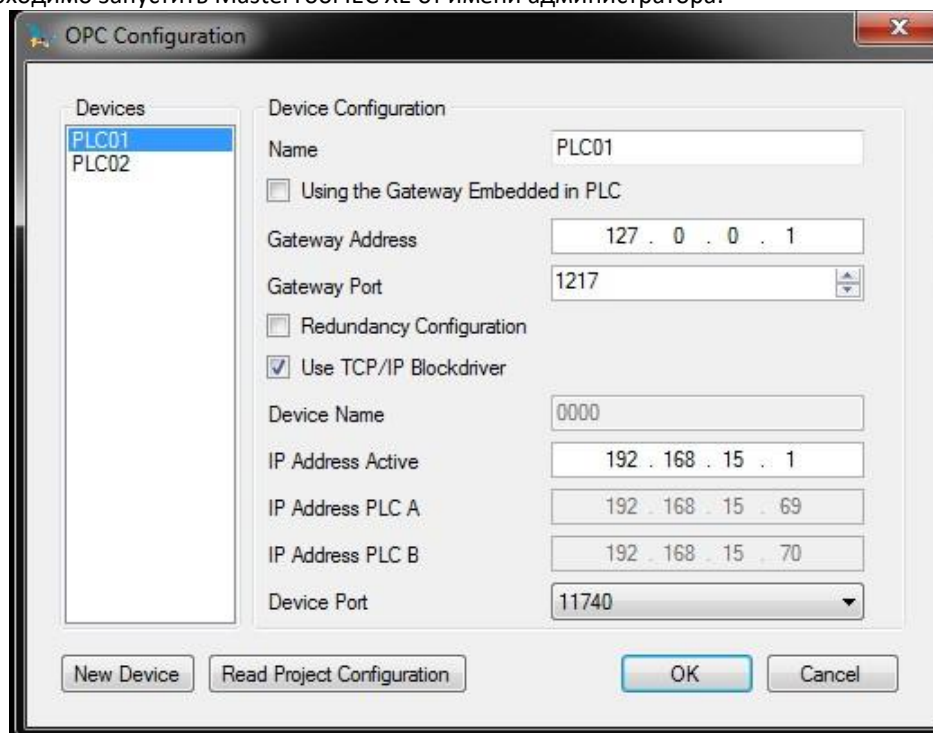




Рисунок 79: Настройки сервера OPC DA

Конфигурация шлюза также задается в шлюзе, используемом для связи между MasterTool IEC XE и ПЛК и описанном в разделе «Настройки связи» (Communication Settings), приведенном в руководстве пользователя MasterTool IEC XE – MU299609. Если используется конфигурация *localhost* («локальный хост»), то адрес шлюза (Gateway Address) должен быть заполнен с использованием 127.0.0.1. Такая конфигурация необходима, поскольку сервер OPC DA (OPC DA Server) использует тот же коммуникационный шлюз и тот же протокол, который используется для связи между ПЛК и MasterTool IEC XE.

Существует опция «Использование шлюза, встроенного в ПЛК» (Using the Gateway Embedded in PLC), которая может быть выбрана, когда необходимо использовать шлюз, находящийся в самом ПЛК. Данная опция может быть использована для оптимизации связи, поскольку позволяет избежать избыточного трафика через определенную станцию, когда к одному ПЛК подключено несколько станций с клиентом OPC (OPC Client).

Для конфигурирования ПЛК возможны два типа конфигурации, в зависимости от установки флажка «Использовать драйвер блокировки (Blockdriver) TCP/IP» (Use TCP/IP Blockdriver). Если опция не была выбрана, то поле «Имя устройства» (Device Name) должно быть заполнено именем ПЛК. Данное имя отображается в ПЛК, выбранном в качестве активного на экране «Настройки связи» (Communication Settings).

Другой вариант - использовать IP-адрес (IP Address) интерфейсов сети Ethernet. В этом поле должен быть указан тот же адрес, что и на экранах конфигурации. Помимо этого, при использовании данного способа номер порта должен быть установлен на 11740. Подтверждение сохранит конфигурацию сервера OPC DA (OPC DA Server).

Конфигурация устройства	Описание	Настройка по умолчанию	Возможности
Имя	Описание ПЛК в конфигурационном файле сервера OPC DA (OPC DA Server). Данное поле может иметь любое имя, но в организационных целях рекомендуется использовать имя проекта, загруженного в ПЛК.	'PLC01'	Данное поле относится к типу STRING («Строка») и принимает алфавитно-цифровые (буквы и цифры) символы и символ «_». Не допускается инициировать STRING («Строка») с цифрами или с «_». Допускается до 49 символов.
Адрес шлюза	IP-адрес компьютера, на котором установлен сервер OPC DA (OPC DA Server), для случаев, когда все ПЛК находятся в одной подсети. Если какой-либо ПЛК находится в другой подсети, то необходимо указать шлюз, используемый в данной подсети.	127.0.0.1	от 0.0.0.0 до 255.255.255.255
Порт шлюза	Порт TCP для соединения со шлюзом.	1217	от 2 до 65534
Имя устройства	Это имя ПЛК, отображаемое в «Настройках связи» на вкладке «Устройство» (Device). Имя - это STRING («Строка») перед шестнадцатеричным значением, которая находится между [ ]. Включается только в том случае, если не установлен флажок «Использовать TCP/IP Blockdriver».	'0000'	Это поле является STRING («Строка») и принимает любые символы, как это сделано при настройке имени ПЛК на вкладке «Настройки связи» устройства. Допускается до 49 символов.

## 5. КОНФИГУРАЦИЯ

IP-адрес активен	IP-адрес ПЛК. Включается только при установленном флажке «Использовать TCP/IP Blockdriver». Используется только в том случае, если настройка не является избыточной.	192.168.15.1	от 0.0.0.0 до 255.255.255.255
IP-адрес ПЛК А	IP-адрес ПЛК А. Включается только при резервировании конфигурации. Это основной адрес ПЛК, с которым сервер будет связываться при отсутствии сбоя.	192.168.15.69	от 0.0.0.0 до 255.255.255.255
IP-адрес ПЛК В	IP-адрес ПЛК В. Включается только при резервировании конфигурации. Это вторичный адрес ПЛК, с которым сервер будет связываться в случае сбоя.	192.168.15.70	от 0.0.0.0 до 255.255.255.255
Порт устройства	Порт TCP. Включается только при установленном флажке «Использовать TCP/IP Blockdriver».	11740	11740 или 11739

Таблица 94: Параметры конфигурации каждого ПЛК для сервера OPC DA

При необходимости конфигурирования нового ПЛК на сервере OPC DA (OPC DA Server) достаточно нажать кнопку «Новое устройство» (*New Device*), и конфигурация будет создана. При обращении к экрану настройки отображается список всех ПЛК, уже сконфигурированных на сервере OPC DA («OPC DA Server»). Существующие конфигурации можно редактировать, выбрав ПЛК в списке «Устройства» (*Device*) и отредактировав его параметры. Параметры ПЛК, которые больше не используются, могут быть удалены. Максимальное количество ПЛК, сконфигурированных на сервере OPC DA (OPC DA Server), составляет 16.

Если используемая архитектура автоматизации предусматривает, что сервер OPC DA (OPC DA Server) должен быть запущен на компьютере, который не осуществляет связь с ПЛК через MasterTool IEC XE, то инструмент должен быть установлен на этом компьютере, чтобы обеспечить конфигурирование сервера OPC DA (OPC DA Server) таким же образом, как это делается в других ситуациях.

**ВНИМАНИЕ:**

Для сохранения конфигурации сервера OPC DA необходимо запустить MasterTool IEC XE с правами администратора на операционной системе. В зависимости от версии ОС данную привилегию необходимо создать в момент запуска программы: щелкнуть правой кнопкой мыши на значке MasterTool IEC XE и выбрать пункт «Выполнение» (*Run*) от имени администратора (*Administrator*).

**ВНИМАНИЕ:**

Настройки ПЛК на сервере OPC DA не сохраняются в проекте, созданном в MasterTool IEC XE. По этой причине ее можно выполнять как в открытом, так и в закрытом проекте. Настройки хранятся в конфигурационном файле, в котором установлен сервер OPC DA. При изменении настроек не требуется загружать приложение на ПЛК, но в зависимости от клиента OPC может потребоваться повторное подключение к серверу или загрузка настроек для корректного обновления данных.

### 5.4.7.2.1. Импорт конфигурации проекта

С помощью кнопки «Считывание конфигурации проекта» (*Read Project Configuration*), как показано на рисунке 79, можно присвоить конфигурацию открытого проекта редактируемой конфигурации ПЛК. Для корректной работы данной опции должен быть открыт проект и установлен *активный путь* (*Active Path*), как описано в разделе «Настройки связи» (*Communication Settings*), присутствующем в руководстве пользователя MasterTool IEC XE – MU299609. Если какое-либо из этих условий не выполняется, то будет выдано сообщение об ошибке, и данные не будут изменены.

## 5. КОНФИГУРАЦИЯ

При выполнении указанных условий в настройке ПЛК вводятся параметры открытого проекта. Информация об *IP-адресе (IP Address)* и *порте шлюза (Gateway Port)* настраивается в соответствии с описанием, приведенным в разделе «*Настройка связи (Communication Settings)*» в соответствии с *активным путем (Active Path)*. Тем не менее, настройки IP-адреса считываются из настроек интерфейса сети Ethernet NET 1. Порт для подключения к ПЛК в этом случае всегда назначается 11740.

### 5.4.7.3. Переменные качества и состояния связи OPC DA

Для каждого ПЛК, созданного в сервере OPC DA (OPC DA Server), формируются переменные состояния с именами *\_CommState* и *\_CommStateOK*. Переменная *\_CommState* указывает на состояние связи между OPC и ПЛК. Данное состояние может интерпретироваться клиентами OPC в соответствии с приведенной ниже таблицей.

Состояние	Значение	Описание
STATE_TERMINATE	-1	Если данная связь между сервером OPC DA (OPC DA Server) и клиентом OPC (OPC Client) прерывается, то будет возвращено данное значение. Если одновременно подключено несколько клиентов OPC, то возврат произойдет при отключении последнего из них. Помимо того, что это состояние находится в переменной, его значение не может быть визуализировано, поскольку оно изменяется только при отсутствии связи с клиентом.
STATE_PLC_NOT_CONNECTED	0	ПЛК, сконфигурированный в сервере OPC DA (OPC DA Server), не подключен. Это может произойти, если конфигурация неверна (неверный IP-адрес ПЛК и/или шлюза) или ПЛК в данный момент недоступен.
STATE_PLC_CONNECTED	1	ПЛК, сконфигурированный в сервере OPC DA Server, подключен. Это переходное состояние во время соединения.
STATE_NO_SYMBOLS	2	В ПЛК, сконфигурированном в сервере OPC DA (OPC DA Server), отсутствуют символы (переменные). Это может произойти, если нет символов или на ПЛК не загружен проект.
Состояние	Значение	Описание
STATE_SYMBOLS_LOADED	3	Завершился процесс считывания символов (переменных) из ПЛК, сконфигурированного в сервере OPC DA (OPC DA Server). Это переходное состояние во время соединения.
STATE_RUNNING	4	После считывания символов (переменных) сервера OPC DA (OPC DA Server) выполняет периодическое обновление значений доступных символов в каждом сконфигурированном ПЛК.
STATE_DISCONNECT	5	Произошел разрыв связи с ПЛК, сконфигурированным в сервере OPC DA (OPC DA Server).
STATE_NO_CONFIGURATION	6	Если конфигурация OPC (хранящаяся в файле OPCServer.ini) имеет неправильный синтаксис, то значение переменной будет таким. Как правило, такая логика работы не наблюдается, поскольку MasterTool IEC XE поддерживает данную конфигурацию в действительном состоянии.

Таблица 95: Описание состояний связи между сервером OPC DA и ПЛК

*\_CommStateOK* представляет собой переменную типа Bool (булева), которая показывает, работает ли связь между сервером OPC DA (OPC DA Server) и ПЛК. Если значение равно TRUE («ИСТИНА»), это означает, что связь работает правильно. Если значение FALSE («ЛОЖЬ»), то по какой-то причине установить связь с ПЛК невозможно.

Помимо контроля состояния связи, клиент OPC (OPC Client) может получить информацию о качестве связи. Биты качества образуют байт. Они разделены на три группы битов: *Quality* («Качество»), *Substatus* («Подстатус») и *Limit* («Предел»). Биты распределяются следующим образом: *QQSSSSL*, где *QQ* - биты качества, *SSSS* - биты подстатуса и *LL* - биты лимита. При этом биты *QQ* являются старшими в байте, а биты *LL* - младшими.

QQ	Значения битов	Определение	Описание
----	----------------	-------------	----------

0	00SSSSL	Плохо	Значение read («считывание») не может быть использовано из-за проблем с соединением. Можно проследить за значением <i>_CommState</i> и произвести диагностику проблемы.
1	01SSSSL	Неопределенность	Качество не может быть определено и может быть представлено в поле «Подстатус» (Substatus).
2	10SSSSL	Не применимо	Данное значение зарезервировано и не используется стандартом OPC.
3	11SSSSL	Хорошо	Качество хорошее, и считанное значение может быть использовано.

Таблица 96: Описание значения качества OPC

В таблице 96 представлены возможные значения качества. Сервер OPC DA (OPC DA Server) возвращает только значения «Хорошее качество» (*Good Quality*) и «Плохое качество» (*Bad Quality*). Клиент OPC (OPC Client) может поддерживать качество как «Неопределенное» (*Uncertain*) по причине сбоев, при которых он не может установить соединение с сервером. В случае мониторинга 8 битов качества непосредственно с сервера OPC DA (OPC DA Server) поля «Подстатус» (*Substatus*) и «Предел» (*Limit*) должны быть нулевыми, и «Хорошее качество» (*Good Quality*) будет представлено как значение 192, а «Плохое качество» (*Bad Quality*) - как значение 0.

#### 5.4.7.4. Пределы связи с сервером OPC DA

Для связи с сервером OPC DA (OPC DA Server) существуют некоторые ограничения, которые должны соблюдаться для корректной работы приложения. Для каждого ПЛК не может быть более 20000 переменных, сконфигурированных как доступные на сервере OPC DA (OPC DA Server). Таким образом, 20000 переменных – это максимальный предел для связи с одним ПЛК.

Помимо этого, при конфигурировании переменных, доступных на сервере OPC DA (OPC DA Server), количество объявленных переменных в каждой структурной единице программы (POU) или списке глобальных переменных (GVL) не может превышать предельного значения в 5000. В таблице, приведенной ниже, представлены ограничения на конфигурацию сервера OPC DA (OPC DA Server).

Максимальное количество переменных, взаимодействующих с одним ПЛК	20000
Максимальное количество переменных, объявленных в одной структурной единице программы (POU) или списке глобальных переменных (GVL)	5000
Максимальное количество ПЛК на сервере OPC DA (OPC DA Server)	16
Максимальное количество одновременных подключений сервера OPC DA (OPC DA Server) в одном ПЛК	8

Таблица 97: Ограничения связи с сервером OPC DA

#### ВНИМАНИЕ:

Максимальное количество одновременных подключений сервера OPC DA в одном ПЛК разделяется с подключениями, осуществляемыми с помощью MasterTool IEC XE. Иными словами, сумма подключений сервера OPC DA и MasterTool IEC XE не должна превышать максимального количества, определенного в таблице 97.

Для связи между сервером OPC DA (OPC DA Server) и ПЛК используется тот же протокол, что и для связи MasterTool IEC XE с ПЛК. Данный протокол доступен только для интерфейсов сети Ethernet центральных процессоров (ЦП) серии Nexto, с модулями расширения сети Ethernet подобный обмен данными невозможен.

При установлении связи между сервером OPC DA (OPC DA Server) и ПЛК эти два элемента инициируют серию операций, направленных на решение адресов каждой из объявленных переменных, оптимизируя связь в режиме считывания данных. Помимо этого, на данном этапе для оптимизации связи определяются коммуникационные группы,

## 5. КОНФИГУРАЦИЯ

используемые некоторыми клиентами. Данный начальный процесс требует определенного времени и зависит от количества отображаемых переменных.

Примерное время данного начального этапа при 5000 переменных составляет около 2 минут. В случаях, когда используется большее количество переменных, это время может увеличиться до 4 минут, в зависимости от типа данных и настроек сети.

### 5.4.7.5. Доступ к данным через клиент OPC DA (OPC DA Client)

После конфигурирования сервера OPC DA (OPC DA Server) доступ к имеющимся данным всех ПЛК можно получить через клиент OPC (OPC Client). В конфигурации клиента OPC необходимо выбрать имя сервера OPC DA (OPC DA Server). В данном случае это имя *CoDeSys.OPC.DA*.

На приведенном ниже рисунке показан выбор сервера в клиентском драйвере программного обеспечения SCADA (автоматизированная система управления технологическим процессом) BluePlant.

#### ВНИМАНИЕ:

Так же, как и в MasterTool IEC XE, для корректной работы клиента OPC выполнение некоторых инструментов должно выполняться с правами администратора в операционной системе. В зависимости от версии ОС данная привилегия должна быть активизирована в момент выполнения программы. Для этого щелкните правой кнопкой мыши на значке MasterTool IEC XE и выберите пункт «Выполнение» (Run) от имени администратора (Administrator).

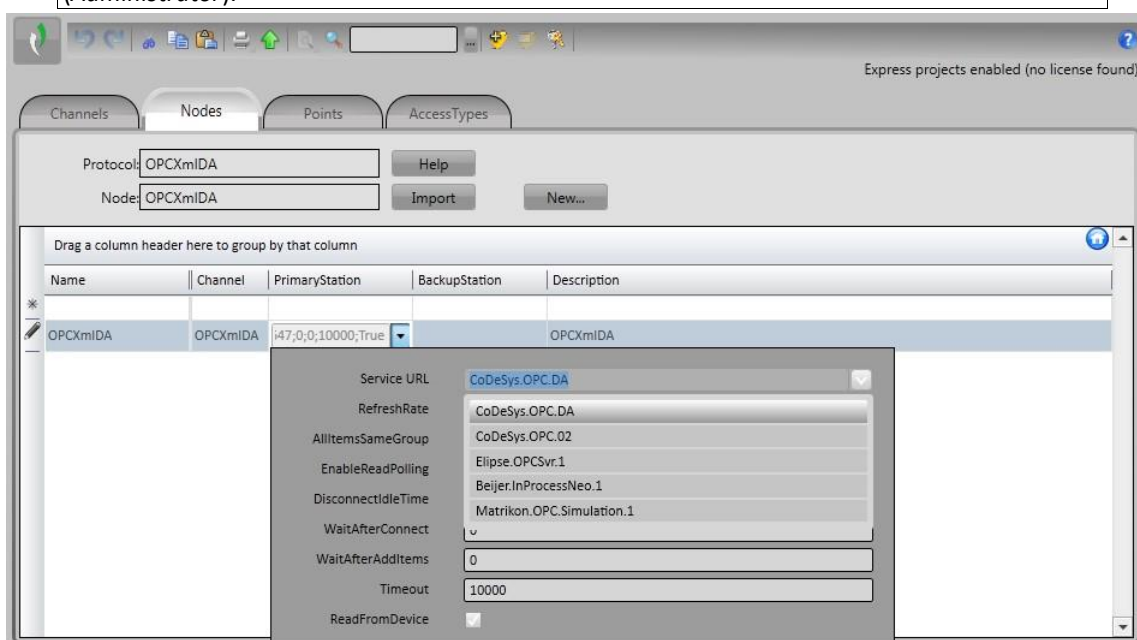


Рисунок 80: Выбор сервера OPC DA в конфигурации клиента

В случаях, когда сервер расположен удаленно, может потребоваться добавить сетевой путь или IP-адрес компьютера, на котором установлен сервер. В этих случаях возможны два варианта настройки. Первый - непосредственная настройка, при этом необходимо включить службу COM/DCOM Windows Service. Тем не менее, более простым способом является использование инструмента-проходчика (Tunneller), который абстрагирует настройки COM/DCOM и обеспечивает более безопасное взаимодействие между клиентом и сервером. Более подробная информация об этом типе инструмента приведена в статье *NAP151 - Tunneller OPC*.

После того как клиент соединился с сервером, можно использовать команды импорта тегов (TAG). Данные команды обращаются к информации, объявленной в ПЛК, и возвращают список со всеми имеющимися в нем символами.

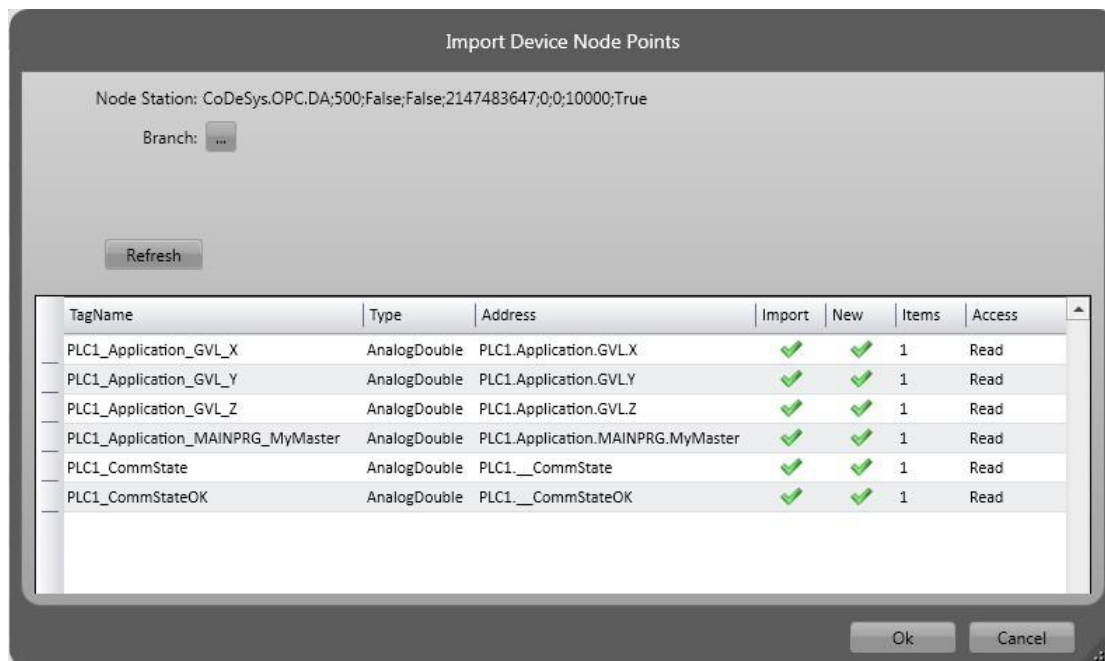


Рисунок 81: Список символов, к которому обращается клиент OPC

Список выбранных переменных будет включен в список коммуникаций клиента и может быть использован, например, в экране системы SCADA (автоматизированная система управления технологическим процессом).

**ВНИМАНИЕ:**

Для проведения тестов связи OPC можно использовать режим моделирования программного обеспечения MasterTool IEC XE. Информация по его настройке представлена в разделе «Тест связи OPC с помощью моделирующего устройства (симулятора)» (*Testing an OPC Communication using the Simulator*) руководства пользователя MasterTool IEC XE – MU299609.

#### 5.4.8. Сервер OPC UA (OPC UA Server)

Протокол OPC UA представляет собой эволюцию семейства OPC. Независимо от платформы он призван стать новым стандартом, используемым в промышленных коммуникациях.

Будучи основанным на архитектуре «клиент/сервер», протокол OPC UA предоставляет многочисленные преимущества при разработке проектов и средств связи с системами автоматизации.

При разработке проекта конфигурирование связи и обмен информацией между системами чрезвычайно просты с использованием технологии OPC UA. При использовании других адресных драйверов необходимо создавать таблицы для связи тегов систем наблюдения и переменных программируемого (логического) контроллера. При изменении областей данных в процессе разработки проекта необходимо заново создавать преобразования и новые таблицы с взаимосвязями между информацией ПЛК и системы SCADA (автоматизированная система управления технологическим процессом).

5. КОНФИГУРАЦИЯ

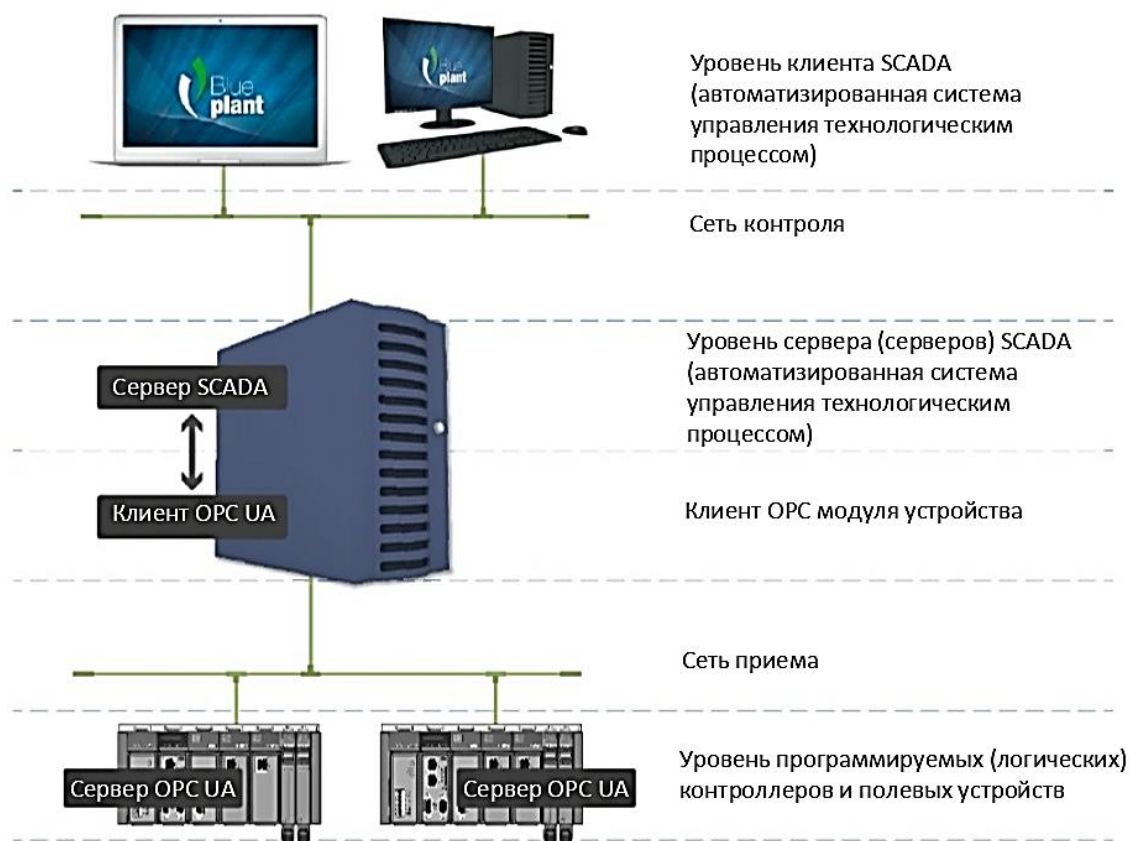


Рисунок 82: Архитектура OPC UA

На приведенном выше рисунке представлена типовая архитектура связи системы SCADA (автоматизированная система управления технологическим процессом) и ПЛК при проектировании систем автоматизации. На данном рисунке явным образом представлены все роли, присутствующие в коммуникации, независимо от того, где они выполняются, - на одном и том же оборудовании или на разном. Каждая из ролей данной архитектуры описана в таблице ниже.

Роль	Описание
Уровень программируемых (логических) контроллеров и полевых устройств	В полевых устройствах и ПЛК хранится информация о состоянии работы и управлении установкой. Система SCADA (автоматизированная система управления технологическим процессом) получает доступ к информации на данных устройствах и сохраняет ее на сервере SCADA (автоматизированная система управления технологическим процессом), чтобы клиенты SCADA (автоматизированная система управления технологическим процессом) могли обращаться к ней во время работы установки.
Модули сервера OPC UA	Сервер OPC UA (OPC UA Server) представляет собой внутренний модуль ПЛК, отвечающий за прием запросов OPC UA и их трансляцию для связи с полевыми устройствами.

## 5. КОНФИГУРАЦИЯ

Сеть сбора данных	Сеть сбора данных представляет собой сеть, по которой передаются сообщения OPC UA для запроса данных, собираемых с ПЛК и полевых устройств.
Модуль устройства клиента OPC	Модуль клиента OPC UA (OPC UA Client), входящий в состав сервера SCADA (автоматизированная система управления технологическим процессом), отвечает за выполнение запросов к серверу OPC UA по протоколу OPC UA. Собранные им данные хранятся в базе данных сервера SCADA (автоматизированная система управления технологическим процессом).
Уровень сервера SCADA (автоматизированная система управления технологическим процессом)	Сервер SCADA (автоматизированная система управления технологическим процессом) отвечает за подключение к различным коммуникационным устройствам и хранение собранных ими данных в базе данных, чтобы к ним могли обращаться клиенты SCADA (автоматизированная система управления технологическим процессом).
Сеть контроля	Сеть контроля представляет собой сеть, по которой клиенты SCADA (автоматизированная система управления технологическим процессом) подключаются к серверам SCADA (автоматизированная система управления технологическим процессом), часто с использованием проприетарного протокола системы SCADA (автоматизированная система управления технологическим процессом). В топологии, в которой не используется несколько Клиентов или Сервер и Клиент установлены в одном оборудовании, такая сеть отсутствует, и в этом случае данное оборудование должно напрямую использовать протокол OPC UA для связи с ПЛК.
Уровень клиента SCADA (автоматизированная система управления технологическим процессом)	Клиенты SCADA (автоматизированная система управления технологическим процессом) отвечают за запрос к серверам SCADA (автоматизированная система управления технологическим процессом) необходимых данных для отображения на экране, на котором происходит работа установки. Через них можно выполнять считывания и записи по данным, хранящимся в базе данных сервера SCADA (автоматизированная система управления технологическим процессом).

Таблица 98: Описание ролей в архитектуре сервера OPC UA

При использовании протокола OPC UA связь между метками (тегами) систем диспетчеризации и данными процесса в переменных контроллера полностью прозрачна. Это означает, что при изменении областей данных в ходе разработки проекта нет необходимости заново устанавливать связи между информацией ПЛК и SCADA (автоматизированная система управления технологическим процессом). Достаточно просто использовать новую переменную, предоставленную ПЛК, в системах, запрашивающих эти данные.

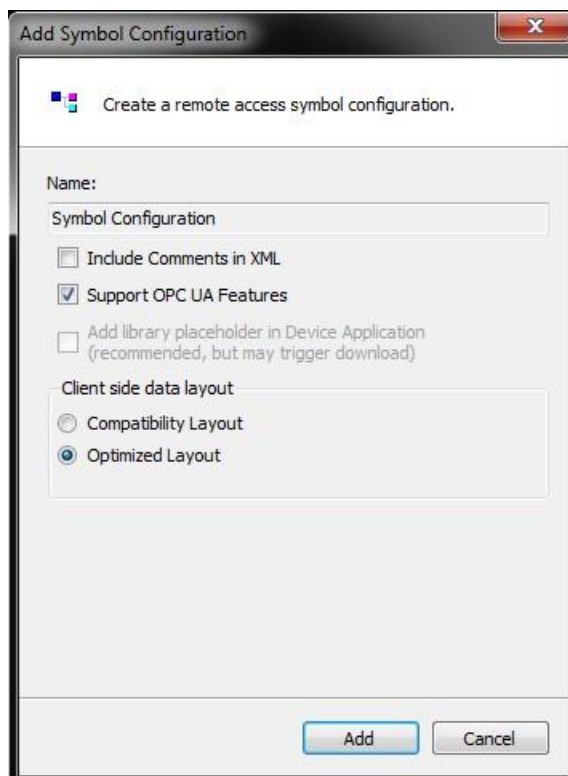
Использование OPC UA обеспечивает более высокую производительность и связь с системами SCADA (автоматизированная система управления технологическим процессом). Это способствует сокращению времени разработки приложений и расходов на их техническое обслуживание. Он также позволяет вводить новые данные в коммуникацию упрощенным способом с большей гибкостью и функциональной совместимостью между системами автоматизации, поскольку является открытым стандартом.

Следует отметить, что OPC UA доступен только на локальных интерфейсах сети Ethernet центральных процессоров (ЦП) Nexto. Модули расширения сети Ethernet не поддерживают данную функциональность.



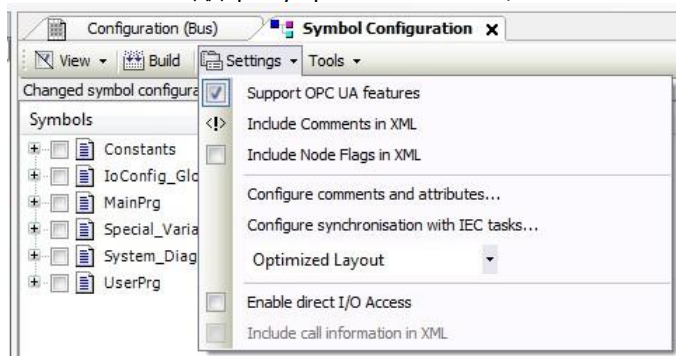
## 5.4.8.1. Создание проекта для связи с OPC UA

Шаги по созданию проекта с OPC UA очень похожи на шаги, описанные в разделе «Создание проекта для связи с OPC DA». Как и в случае с протоколом OPC DA, конфигурация протокола OPC UA основана на конфигурации объекта «Символьная конфигурация» (*Symbol Configuration*). Чтобы включить OPC UA, достаточно включить опцию «Поддержка функций OPC UA» (*Support OPC UA Features*) в конфигурации, как показано на рисунке, приведенном ниже.

Рисунок 83: Объект «Символьная конфигурация» (*Symbol Configuration*)**ВНИМАНИЕ:**

При включении поддержки протокола OPC UA поддержка протокола OPC DA остается включенной. Вы можете одновременно включить связь OPC UA и OPC DA, чтобы сообщать о переменных, настроенных в объекте «Символьная конфигурация» (*Symbol Configuration*) либо через атрибуты.

Другой способ для получения доступа к этой конфигурации, если уже создан проект с объектом «Символьная конфигурация» (*Symbol Configuration*), заключается в обращении к меню «Настройки» (*Settings*) вкладки конфигурации объекта «Символьная конфигурация» (*Symbol Configuration*). Просто выберите опцию «Поддержка функций OPC UA» (*Support OPC UA Features*), чтобы включить поддержку протокола OPC UA, как показано на рисунке, приведенном ниже.



### Рисунок 84: Включение OPC UA в объекте «Символьная конфигурация» (Symbol Configuration)

После выполнения данной процедуры проект может быть загружен в ПЛК, и выбранные переменные будут доступны для связи с сервером OPC UA (OPC UA Server).

#### 5.4.8.2. Типы поддерживаемых переменных

В данном разделе определяются типы переменных, поддерживающих связь по протоколу OPC UA, если они объявлены в списках глобальных переменных (GVL) или структурных единицах программы (POU), а также выбраны в объекте «Символьная конфигурация» (*Symbol Configuration*) (см. предыдущий раздел).

Поддерживаются следующие типы простых переменных:

- BOOL
- SINT
- USINT / BYTE
- INT
- UINT / WORD
- DINT
- UDINT / DWORD
- LINT
- ULINT / LWORD
- REAL
- LREAL
- STRING
- TIME
- LTIME

Можно также использовать структурированные типы (STRUCT или функциональные блоки), созданные на основе предыдущих простых типов.

Наконец, можно создавать массивы из простых или структурированных типов.

#### 5.4.8.3. Ограничение количества подключенных клиентов на сервере OPC UA (OPC UA Server)

Максимальное количество одновременно подключенных клиентов OPC UA в ПЛК - 8 (восемь).

#### 5.4.8.4. Ограничение переменных связи на сервере OPC UA (OPC UA Server)

Максимальное количество переменных, конфигурируемых в ПЛК для связи через OPC UA, составляет 20000. Тем не менее, при конфигурировании переменных, которые будут доступны для сервера OPC UA (OPC UA Server), количество переменных, объявленных в каждой структурной единице программы (POU) или списке глобальных переменных (GVL), не должно превышать предела в 5000, при этом для использования 20000 переменных требуется не менее четырех структурных единиц программы (POU) или списков глобальных переменных (GVL).

Необходимо учитывать следующие замечания по подсчету переменных:

- Каждая простая переменная считается одной переменной;
- Каждое поле в структуре имеет одну переменную;
- Каждая позиция массива имеет одну переменную, за исключением массивов структур, где каждая позиция считается как количество полей в структуре.




При установлении связи между сервером OPC UA (OPC UA) и ПЛК эти два элемента иницируют серию операций, направленных на решение адреса каждой объявленной переменной, оптимизируя связь в режиме чтения данных. Помимо этого, на данном этапе также решаются вопросы классификации групп связи, используемых некоторыми клиентами для целей оптимизации связи. Данный начальный процесс занимает некоторое время и зависит от количества преобразованных переменных.

#### 5.4.8.5. Настройки шифрования



При желании пользователь может настроить шифрование для связи OPC UA с использованием профиля *Basic256SHA256* для обеспечения безопасного соединения (кибербезопасности).

## 5. КОНФИГУРАЦИЯ

Чтобы настроить шифрование на сервере OPC UA, необходимо создать для него сертификат, выполнив следующие действия в устройстве программирования (программаторе) MasterTool:

1. Определите активный путь для связи с контроллером (вход в систему не требуется);
2. В меню «Вид» (View) выберите пункт «Экран безопасности» (Security Screen);
3. Перейдите на вкладку «Устройства» (Devices) в левой части данного экрана;
4. Щелкните по значку , чтобы выполнить обновление;
5. Щелкните по значку «Устройство» (Device), под которым откроется несколько сертификатов («Собственные сертификаты» (Own Certificates), «Надежные сертификаты» (Trusted Certificates), «Ненадежные сертификаты» (Untrusted Certificates), «Карантинные сертификаты» (Quarantined Certificates));
6. Щелкните по значку  для генерации сертификата и выберите следующие параметры:
  - Длина ключа (Key length) (бит): 3072
  - Срок действия (Validity period) (дней): 365 (при желании может быть изменен)
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 (OPC UA server) и генерации сертификата с профилем Basic256Sha256 (подробную информацию см. в руководстве по конкретному клиенту OPC UA);
10. Вернувшись в MasterTool, щелкните по значку  экрана безопасности (Security Screen), чтобы выполнить обновление;
11. На экране безопасности (Security Screen) выберите папку «Карантинный сертификат» (Quarantined Certificates) в разделе «Устройство» (Device). На правой панели должен появиться сертификат, запрашиваемый клиентом OPC UA;
12. Перетащите данный сертификат в папку «Надежные сертификаты» (Trusted Certificates);
13. Выполните настройки в клиенте OPC UA (подробнее см. руководство по конкретному клиенту OPC UA).

Для удаления шифрования, ранее настроенного на контроллере, необходимо выполнить следующие действия:

1. Определите активный путь для связи с контроллером (вход в систему не требуется);
2. В меню «Вид» (View) выберите пункт «Экран безопасности» (Security Screen);
3. Перейдите на вкладку «Устройства» (Devices) в левой части данного экрана;
4. Щелкните по значку , чтобы выполнить обновление;
5. Щелкните по значку «Устройство» (Device), под которым откроется несколько сертификатов («Собственные сертификаты» (Own Certificates), «Надежные сертификаты» (Trusted Certificates), «Ненадежные сертификаты» (Untrusted Certificates), «Карантинные сертификаты» (Quarantined Certificates));
6. Щелкните по папке «Собственные сертификаты» (Own Certificates) и на правой панели выберите сертификат (сервер OPC UA (OPC UA Server));
7. Щелкните по значку  чтобы удалить данный проект и сертификат драйвера;
8. Выполните возврат контроллера в исходное состояние (выключите и включите).

### 5.4.8.6. Основные параметры связи, настраиваемые в клиенте OPC UA (OPC UA Client)

Некоторые параметры связи OPC UA настраиваются на клиенте OPC UA и согласовываются с сервером OPC UA в момент установления соединения между ними. В следующих подразделах описаны основные параметры связи OPC UA, их значение и порядок выбора соответствующих значений.

В клиенте OPC UA можно группировать переменные сервера в различные подписки (subscriptions). Каждая подписка представляет собой набор переменных, которые сообщаются в одном коммуникационном пакете (PublishResponse), передаваемом от сервера клиенту. Выбор переменных, составляющих каждую подписку, производится в клиенте OPC UA.

### ВНИМАНИЕ:

Группировка переменных в несколько *подписок* интересна для оптимизации вычислительной мощности и потребления пропускной способности канала связи локальной сети Ethernet. Подобные аспекты оптимизации более подробно анализируются в указаниях по применению NAR165, где предлагаются некоторые правила составления *подписок*. В данных указаниях по применению также более подробно рассматриваются некоторые понятия, связанные с протоколом OPC UA.

Некоторые из описанных ниже параметров связи должны быть определены для сервера в целом, другие - для каждой *подписки*, третьи - для каждой переменной, входящей в *подписку*.

#### 5.4.8.6.1. URL-адрес конечной точки

Данный параметр задает IP-адрес и порт TCP сервера, например:

`opc.tcp://192.168.17.2:4840`

В данном примере IP-адрес контроллера равен 192.168.17.2.

Значение порта TCP всегда должно быть равно 4840.

#### 5.4.8.6.2. Интервал публикации (мс) и Интервал выборки (мс)

Для каждой *подписки* должен быть задан параметр «Интервал публикации» (*Publishing Interval*) (единицы измерения: миллисекунды).

Параметр «Интервал выборки» (*Sampling Interval*) должен быть задан для каждой переменной (единицы измерения: миллисекунды). Тем не менее, во многих клиентах OPC UA параметр «Интервал выборки» (*Sampling Interval*) может быть задан для *подписки*, причем он может быть одинаковым для всех переменных, объединенных в *подписку*.

В коммуникационном пакете «Опубликовать отклик» (*Publish Response*) клиенту сообщаются только те переменные *подписки*, значения которых были изменены. Параметр «Опубликовать интервал» (*Publishing Interval*) определяет минимальный интервал между последовательными пакетами «Опубликовать отклик» (*Publish Response*) одной и той же *подписки*, чтобы ограничить потребление обработки и пропускную способность канала связи сети Ethernet.

Чтобы выяснить, какие переменные *подписки* были изменены и должны быть сообщены клиенту в следующем пакете «Опубликовать отклик» (*Publish Response*), сервер должен выполнить сравнения, и такие сравнения (*выборки*) выполняются наравне с *Интервалом выборки* (*Sampling Interval*). Рекомендуется, чтобы значение *Интервала выборки* (*Sampling Interval*) колебалось в пределах от 50% до 100% от значения *Интервала публикации* (*Publishing Interval*), поскольку при сравнении, выполняемом в каждом *Интервале выборки* (*Sampling Interval*), возникает относительно большое потребление обработки данных.

Можно сказать, что сумма между *Интервалом публикации* (*Publishing Interval*) и *Интервалом выборки* (*Sampling Interval*) представляет собой максимальную задержку между изменением значения на сервере и передачей пакета «Опубликовать отклик» (*Publish Response*), сообщающего об этом изменении. Половина этой суммы - средняя задержка между изменением значения на сервере и передачей пакета «Опубликовать отклик» (*Publish Response*), сообщающего об этом изменении.

#### 5.4.8.6.3. Отсчет времени существования и Отсчет времени активного соединения

Эти два параметра должны быть настроены для каждой *подписки*.

Назначение этих двух параметров - создать механизм деактивации *подписки* по инициативе сервера в случае, если он длительное время не получает клиентские пакеты связи *PublishRequest* («Опубликовать запрос») для данной *подписки*. Пакеты *PublishRequest* («Опубликовать запрос») должны быть получены сервером для того, чтобы он мог транслировать пакеты «Опубликовать отклик» (*Publish Response*), содержащие изменившие свои значения переменные *подписки*.

Если сервер не получает пакеты *PublishRequest* («Опубликовать запрос») в течение времени, превышающего значение «Отсчет времени существования» (*Lifetime Count*), умноженное на *Интервал публикации* (*Publishing Interval*), то сервер деактивирует *подписку*, которая в будущем должна быть вновь создана клиентом при желании.

---

## 5. КОНФИГУРАЦИЯ

В ситуациях, когда переменные *подписки* не меняются, может долгое время не передаваться *PublishResponses* («Опубликовать отклики») и, соответственно, *PublishRequests* («Опубликовать запросы»), что приведет к нежелательной деактивации *подписки*. Чтобы этого не произошло, был создан параметр «Отсчет времени активного соединения» (*Keep-Alive Count*). Если в течение времени, равного значению «Отсчет времени активного соединения» (*Keep-Alive Count*), умноженному на *Интервал публикации* (*Publishing Interval*), не происходит изменений данных *подписки*, то сервер посылает небольшой пустой пакет «Опубликовать отклик» (*Publish Response*), указывающий на то, что переменная не изменилась. Данный пустой пакет «Опубликовать отклик» (*Publish Response*) разрешает клиенту немедленно отправить следующий пакет *PublishRequest* («Опубликовать запрос»).

Значение «Отсчет времени активного соединения» (*Keep-Alive Count*) должно быть меньше значения «Отсчет времени существования» (*Lifetime Count*) для предотвращения нежелательной деактивации *подписки*. Рекомендуется, чтобы значение «Отсчет времени существования» (*Lifetime Count*) было как минимум в 3 раза больше, чем «Отсчет времени активного соединения» (*Keep-Alive Count*).

### 5.4.8.6.4. Размер очереди и Сброс самого старого значения

Данные параметры должны поддерживаться со следующими фиксированными значениями, которые, как правило, являются значениями по умолчанию на клиентах:

- Размер очереди (*Queue Size*): 1
- Сброс самого старого значения (*Discard Oldest*): включить

Согласно стандарту OPC UA, можно определить данные параметры для каждой переменной. Тем не менее, многие клиенты позволяют определить общие значения для всех переменных, сконфигурированных в *подписке*.

*Размер очереди* (*Queue Size*) следует оставить со значением 1, поскольку в данной реализации сервера OPC UA (OPC UA server) нет поддержки событий, в связи с чем нет необходимости в определении очереди. Увеличение значения *Размера очереди* (*Queue Size*) может привести к увеличению пропускной способности канала связи и обработки центральным процессором (ЦП), поэтому этого следует избегать.

*Сброс самого старого значения* (*Discard Oldest*) должен поддерживаться со значением «включить» (*enable*), чтобы пакет «Опубликовать отклик» (*Publish Response*) всегда сообщал о последнем изменении значения, обнаруженном для каждой переменной.

### 5.4.8.6.5. Тип фильтра и Тип зоны нечувствительности

Данные параметры должны поддерживаться со следующими фиксированными значениями, которые, как правило, являются значениями по умолчанию на клиентах:

- *Тип фильтра* (*Filter Type*): *DataChangeFilter*
- *Тип зоны нечувствительности* (*Deadband Type*): *нет*

Согласно стандарту OPC UA, можно определить данные параметры для каждой переменной. Тем не менее, многие клиенты позволяют определить общие значения для всех переменных, сконфигурированных в *подписке*.

Параметр «*Тип фильтра*» (*Filter Type*) должен иметь значение *DataChangeFilter*, указывающее на то, что изменение значения переменных должно вызывать его передачу в пакете «Опубликовать отклик» (*Publish Response*).

Параметр «*Тип зоны нечувствительности*» (*Deadband Type*) должен иметь значение «*none*» («*нет*»), поскольку для аналоговых переменных не существует реализации *зоны нечувствительности* (*deadband*). Таким образом, любое изменение аналоговой переменной, каким бы минимальным оно ни было, приводит к ее передаче в пакете «Опубликовать отклик» (*Publish Response*).

Для снижения вычислительной мощности и пропускной способности сети Ethernet можно самостоятельно развернуть *зоны нечувствительности* следующим образом:

- Не включайте аналоговую переменную в *подписку*;
- Вместо этого включите в *подписку* вспомогательную переменную, связанную с аналоговой переменной;
- Скопируйте аналоговую переменную во вспомогательную переменную только при экстраполяции управляемой пользователем *зоны нечувствительности*.

---

## 5. КОНФИГУРАЦИЯ

### 5.4.8.6. Параметры «PublishingEnabled», «MaxNotificationsPerPublish» и «Priority» («Приоритет»)

Предлагается сохранить следующие значения параметров, которые, как правило, используются в клиентах по умолчанию:

- *Publishing* Включено: *true* («истина»)
- *MaxNotificationsPerPublish*: 0
- *Priority* («Приоритет»): 0

Данные параметры должны быть настроены для каждой подписки.

Параметр *PublishingEnable* должен иметь значение «*true*» («истина»), чтобы переменные подписки сообщались в случае изменения значения.

Параметр *MaxNotificationsPerPublish* указывает на то, сколько переменных, изменивших значение, может быть включено в один пакет Publish Response. Специальное значение «0» указывает на отсутствие ограничений, и рекомендуется использовать это значение для того, чтобы все изменившиеся переменные были сообщены в одном пакете «Опубликовать отклик» (*Publish Response*).

Параметр *Priority* («Приоритет») указывает на относительный приоритет данной подписки по сравнению с другими. Если в какой-то момент времени серверу необходимо отправить несколько пакетов «Опубликовать отклик» (*Publish Response*) с разными подписками, то приоритет будет отдан той из них, которая имеет наибольшее значение приоритета. Если все подписки имеют одинаковый приоритет, то пакеты «Опубликовать отклик» (*Publish Response*) будут передаваться в фиксированной последовательности.

### 5.4.8.7. Доступ к данным через клиент OPC UA (OPC UA Client)

После конфигурирования сервера OPC UA (OPC UA Server) доступ к данным, имеющимся во всех ПЛК, можно получить через клиент OPC UA. При конфигурировании клиента OPC UA необходимо выбрать адрес нужного сервера OPC UA (OPC UA Server). В данном случае это адрес *opc.tcp://ip-address-of-device:4840*. На рисунке, приведенном ниже, показан выбор сервера в драйвере клиентского программного обеспечения SCADA (автоматизированная система управления технологическим процессом) BluePlant.

**ВНИМАНИЕ:**

Как и в случае MasterTool IEC XE, для корректной работы клиента OPC UA некоторые инструменты должны быть запущены с правами администратора операционной системы. В зависимости от версии операционной системы это право должно быть авторизовано при запуске программы. Для этого щелкните правой кнопкой мыши на исполняемом файле инструмента и выберите опцию «Выполнение» (*Run*) от имени администратора (*Administrator*).

## 5. КОНФИГУРАЦИЯ

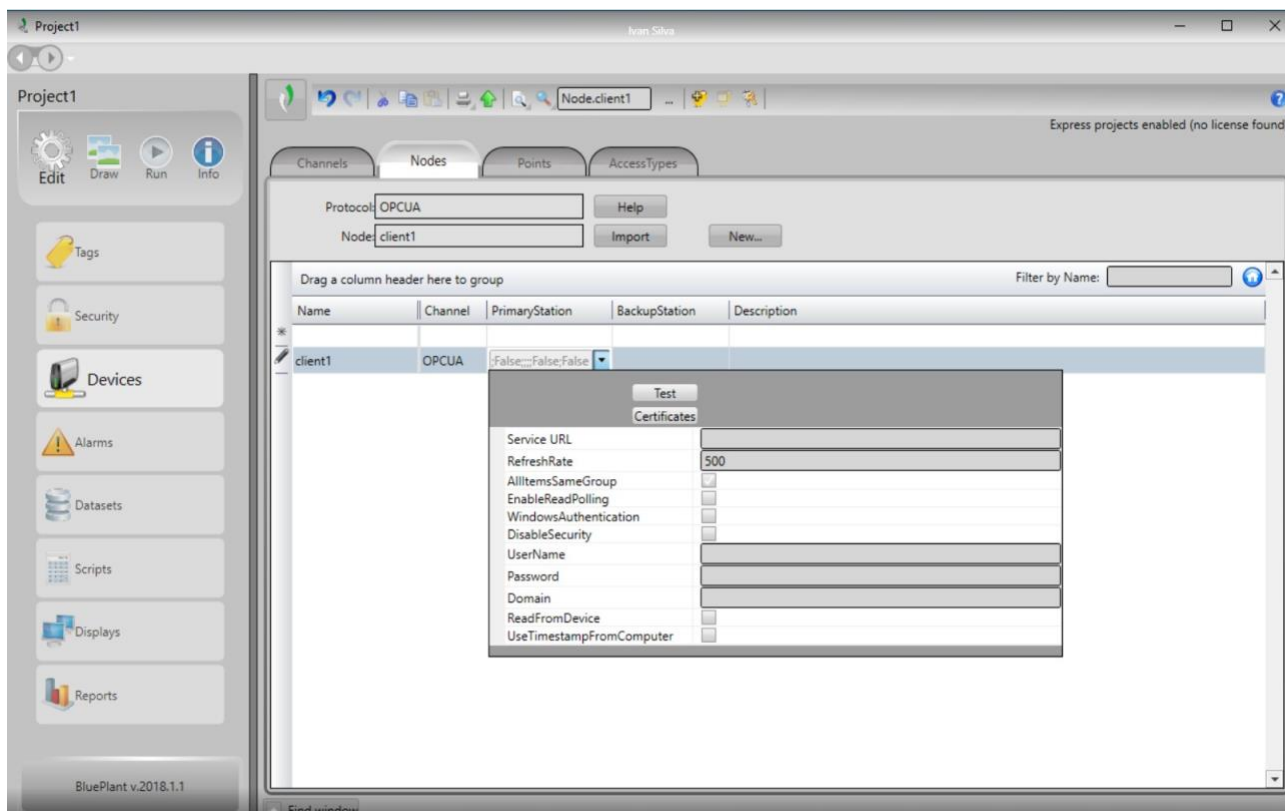


Рисунок 85: Выбор сервера OPC UA в конфигурации клиента

После того как клиент подключится к серверу, можно использовать команды импорта тега (TAG). Данные команды запрашивают информацию, объявленную в ПЛК, и возвращают список со всеми символами, доступными в ПЛК.

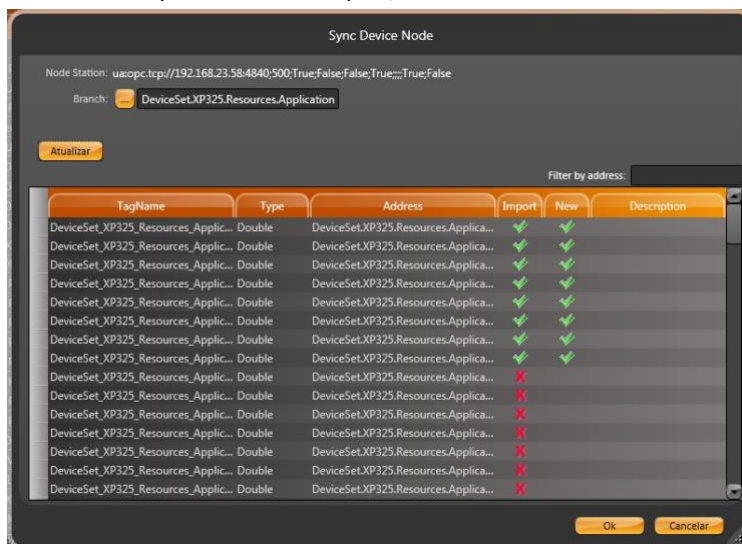


Рисунок 86: Список символов, просматриваемых в OPC UA

Список выбранных переменных будет включен в список коммуникаций клиента и может быть использован, например, на экранах системы SCADA (автоматизированная система управления технологическим процессом).

### 5.4.9. Протокол EtherNet/IP

EtherNet/IP представляет собой протокол с архитектурой «ведущее устройство-ведомое устройство», состоящий из сканера EtherNet/IP (ведущее устройство) и одного или нескольких адаптеров EtherNet/IP (ведомое устройство). Редактор EtherNet/IP предоставляет диалоговые окна для настройки параметров и преобразования входов/выходов с переменными.

Протокол Ethernet/IP основан на протоколе CIP (*Стандартный Промышленный Протокол*), который имеет два основных назначения: Передача управляющих данных, связанных с устройствами ввода/вывода, и передача другой информации, относящейся к управляемой системе, например, параметров конфигурации и диагностики. Первое осуществляется с помощью неявных сообщений, а второе - с помощью явных.

Их система поддержки выполнения программ может выступать как в роли сканера, так и в роли адаптера. Интерфейс NET каждого центрального процессора (ЦП) поддерживает только один экземпляр EtherNet/IP, и он не может ссылаться на модуль расширения сети Ethernet.

Каждое устройство, имеющее один или два сканера EtherNet/IP, поддерживает до 32 адаптеров под ним/ними. Число в 32 адаптера является максимальным. Если сканеров два, то в сумме они не могут иметь значение больше данного предела. А каждый из этих адаптеров не может иметь более 500 входов/выходов (узлов).

Один экземпляр адаптера EtherNet/IP поддерживает до 64 модулей ввода или вывода, ограниченных 505 байтами на входе и 509 байтами на выходе. Данные модули могут иметь значение BYTE, WORD, SINT, INT, DINT, DWORD или REAL. Диапазон устройства MainTask, работающего в качестве адаптера, должен быть меньше или равен интервалу запроса пакета (RPI).

**ВНИМАНИЕ:**

EtherNet/IP не может использоваться совместно с объединением сетевых карт локальной сети Ethernet либо с резервированием полукластеров.

**ВНИМАНИЕ:**

EtherNet/IP требует циклической MainTask, а значит, не может использоваться ни с одной конфигурацией, в которой MainTask задается как произвольная. Например, профиль проекта Basic («Базовый») не поддерживает EtherNet/IP.

**ВНИМАНИЕ:**

Во избежание проблем со связью, сканер EtherNet/IP может иметь адаптеры, настроенные только в пределах одной подсети.

#### 5.4.9.1. Протокол EtherNet/IP

Для добавления сканера или адаптера EtherNet/IP необходимо добавить интерфейс EtherNet/IP в интерфейс NET. Это можно сделать с помощью команды «*Добавить устройство*» (*Add Device*). Под этим интерфейсом EtherNet/IP можно добавить *сканер* или *адаптер*.



## 5. КОНФИГУРАЦИЯ

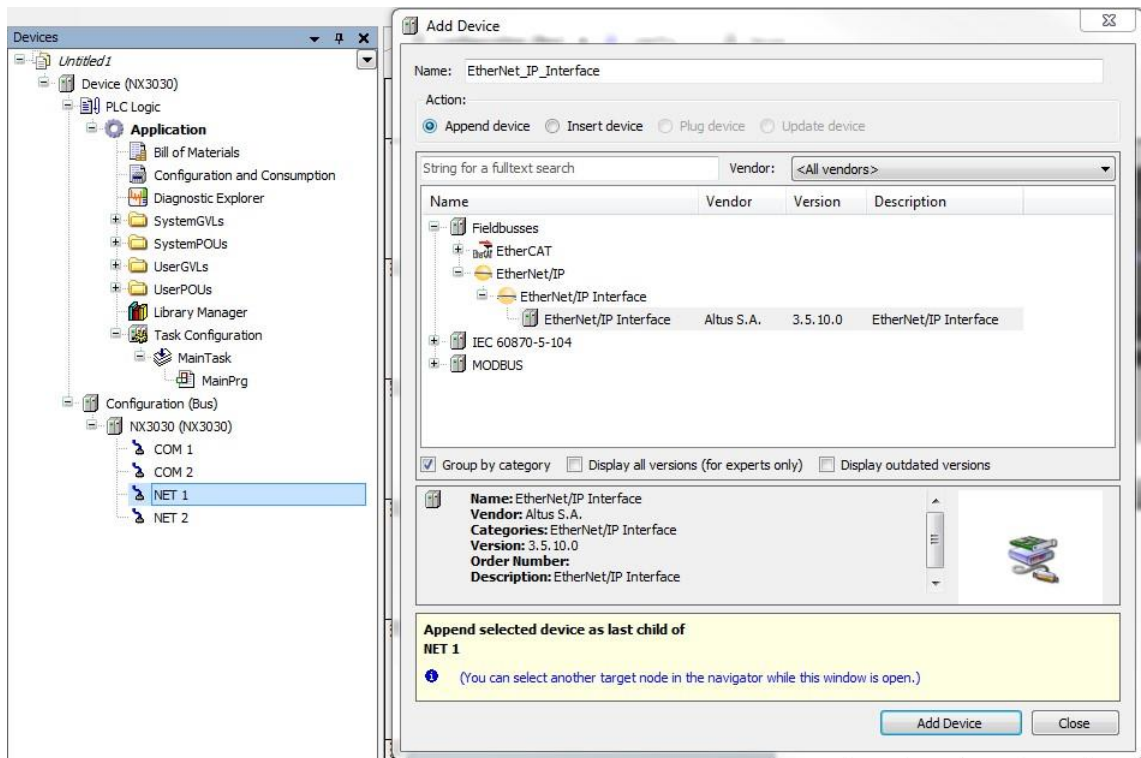


Рисунок 87: Добавление интерфейса EtherNet/IP

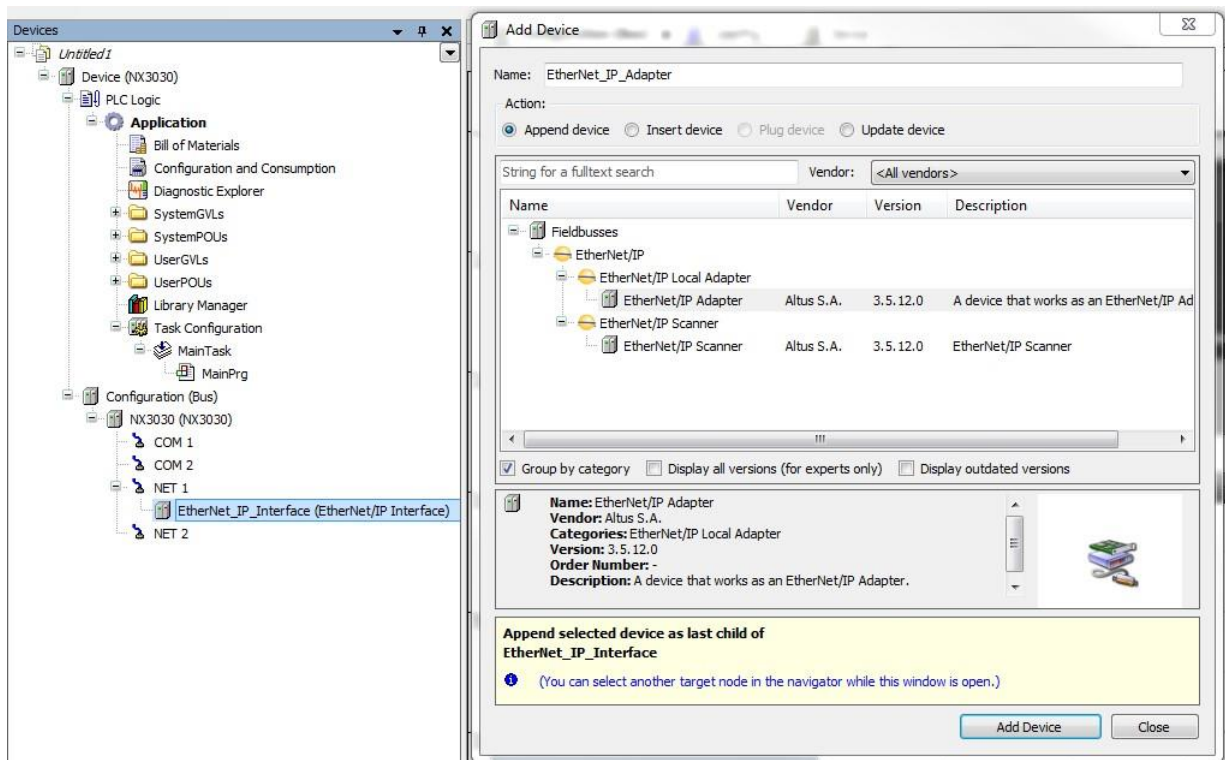


Рисунок 88: Добавление адаптера или сканера EtherNet/IP

### 5.4.9.2. Конфигурация сканера EtherNet/IP

## 5. КОНФИГУРАЦИЯ

Сканеру требуется как минимум один Адаптер, с которым он будет осуществлять обмен данными. Новые Адаптеры могут быть установлены в MasterTool с помощью файлов EDS и DCF. Параметры конфигурации могут отличаться в зависимости от файла описания устройства добавляемого Адаптера.

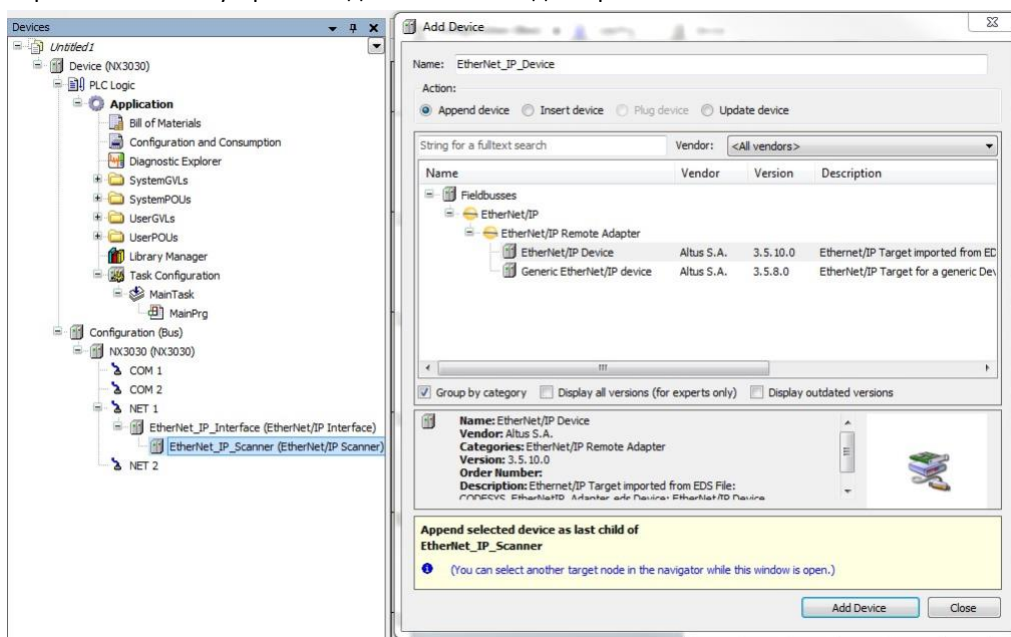


Рисунок 89: Добавление адаптера EtherNet/IP под сканером

### 5.4.9.2.1. Общее

После открытия адаптера, заявленного под сканером, можно произвести его необходимую настройку. Первая вкладка – «Общее» (General), на ней можно сконфигурировать IP-адрес (IP address) и параметры Электронного ввода ключей (Electronic Keying). Данные параметры должны быть отмечены либо с них должны быть сняты отметки, если используемый адаптер установлен на MasterTool. В противном случае, если используемый адаптер имеет тип «Обобщенный» (Generic), то поля «Проверить тип устройства» (Check device type), «Проверить идентификатор поставщика» (Check vendor ID), «Проверить код изделия» (Check product code), «Проверить основную редакцию» (Check major revision) и «Проверить ревизию с несущественными изменениями» (Check minor revision) должны быть заполнены корректной информацией, а флажки установлены на необходимое время. Проверка может быть переключена с режима проверки совместимости (Compatibility Check) на режим проверки строгого тождества (Strict Identity Check).

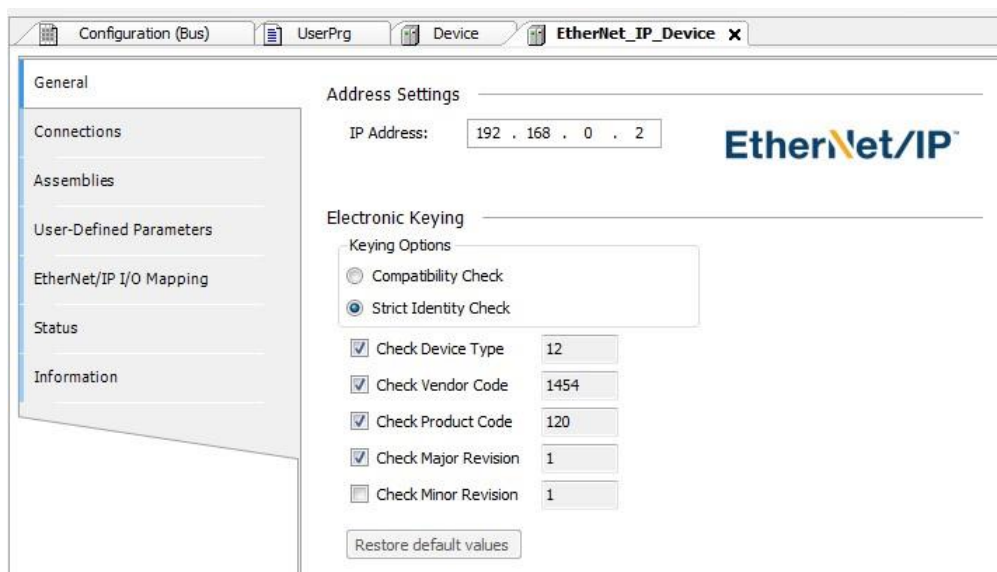


Рисунок 90: Общая вкладка EtherNet/IP

## 5.4.9.2.2. Соединения

В верхней области вкладки «Соединения» (*Connections*) отображается список всех сконфигурированных соединений. Если в файле EDS присутствует соединение «Обладатель исключительных прав» (*Exclusive Owner*), то оно автоматически вставляется при добавлении адаптера. Конфигурационные данные для данных соединений можно изменить в нижней части представления.

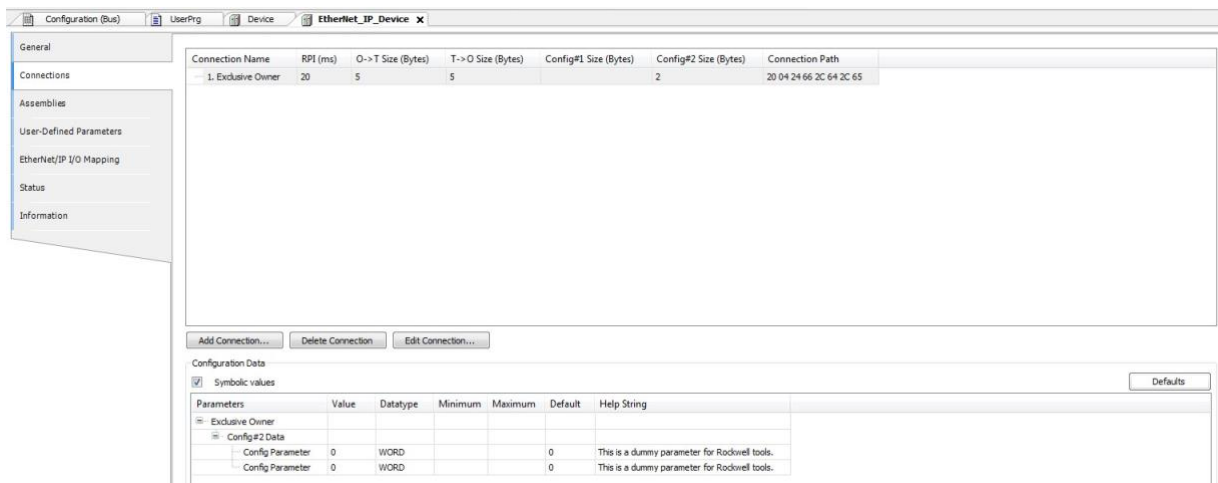


Рисунок 91: Вкладка подключения EtherNet/IP

Примечания:

Для подключения двух или более сканеров EtherNet/IP к одному и тому же удаленному адаптеру:

1. Только один из сканеров может установить соединение «Обладатель исключительных прав» (*Exclusive Owner*).
2. Для сканеров должно быть настроено одинаковое значение *интервала запроса пакета (мс)*.

Конфигурационные данные задаются в файле EDS. Эти данные передаются удаленному адаптеру при открытии соединения.

Конфигурация	Описание	Значение по умолчанию	Опции
Интервал запроса пакета (мс)	Интервал между пакетами запросов: интервал обмена входными и выходными данными.	10 мс	Кратный интервалу MainTask
O -> размер T (Байты)	Размер данных производителя от сканера к адаптеру (Originator -> Target (Отправитель -> Цель))	Зависит от адаптера EDS	0 - 400
T -> размер O (Байты)	Размер потребительских данных от адаптера к сканеру (T -> O)	Зависит от адаптера EDS	0 - 400
Размер конфигурации #1 (Байты)	Размер данных конфигурации 1.	Зависит от адаптера EDS	-
Размер конфигурации #2 (Байты)	Размер данных конфигурации 2.	Зависит от адаптера EDS	-

## 5. КОНФИГУРАЦИЯ

Путь соединения	Адреса объектов конфигурации - входных объектов - выходных объектов.	Зависит от адаптера EDS	-
-----------------	--	-------------------------	---

Таблица 99: Параметры подключения EtherNet/IP

Для добавления новых соединений предусмотрена кнопка «Добавить соединение...» (Add Connection..), которая открывает окно «Новое соединение» (New connection). В данном окне можно сконфигурировать новый тип соединения из predeterminedных в EDS Адаптера либо общее соединение с нуля.

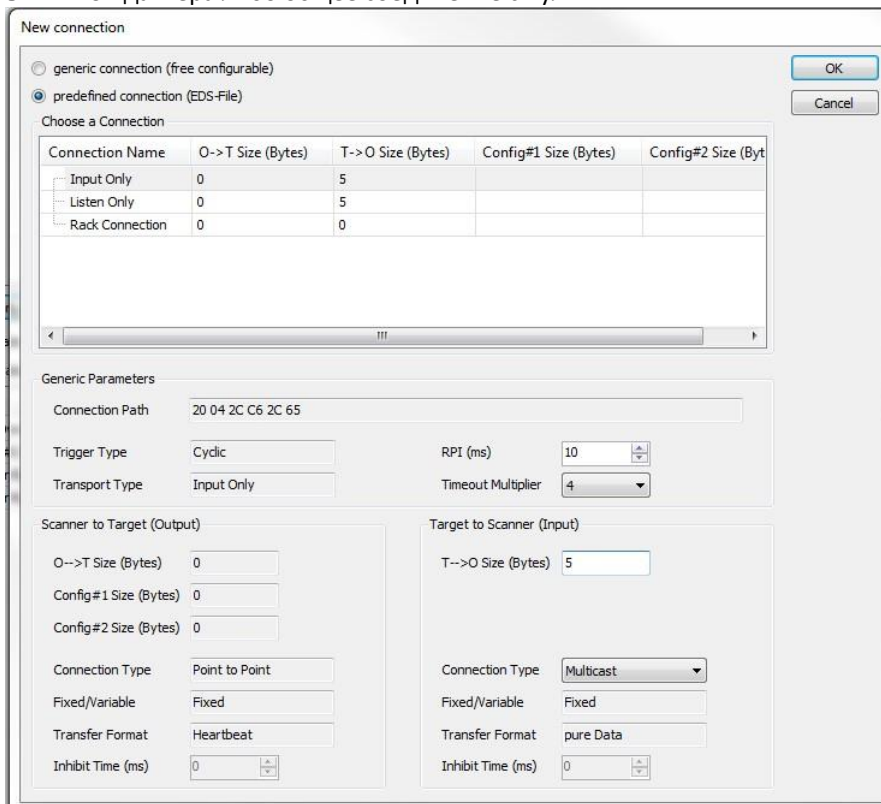


Рисунок 92: Окно нового подключения EtherNet/IP

### 5.4.9.2.3. Узлы

В верхней области вкладки «Узлы» (Assemblies) отображается список всех сконфигурированных соединений. При выборе соединения соответствующие входы и выходы отображаются в нижней области вкладки.

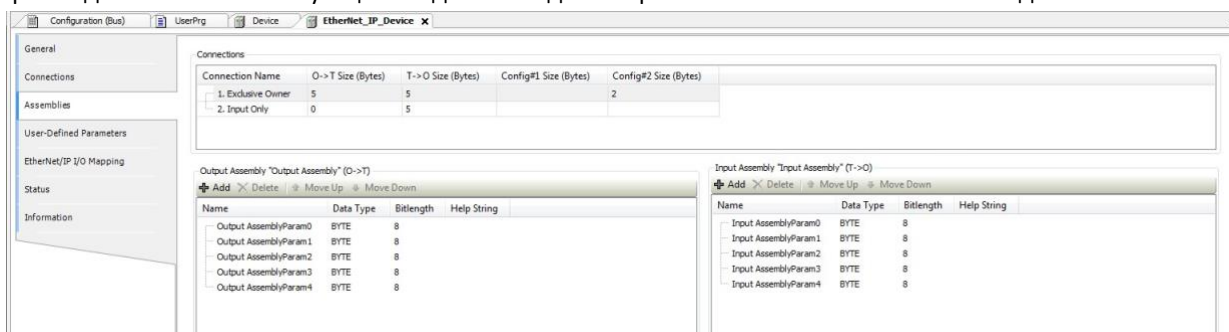


Рисунок 93: Узлы EtherNet/IP

## 5. КОНФИГУРАЦИЯ

Выходной узел и входной узел:

Конфигурация	Описание
Добавить	Открывается диалоговое окно «Добавить вход/выход» (“Add Input/Output”)
Удалить	Удаляет все выбранные входы/выходы.
Переместить вверх	Перемещает выбранный вход/выход в пределах списка.
Переместить вниз	Порядок в списке определяет порядок в отображении входов/выходов.
Имя	Данные значения могут быть изменены двойным щелчком мыши в текстовом поле.
Строка справки	
Длина бита	Данное значение не подлежит редактированию.

Таблица 100: Вкладка узлов EtherNet/IP

Диалоговое окно «Добавить вход/выход» (Add Input/Output):

Конфигурация	Описание
Имя	Название вставляемого входа/выхода.
Тип данных	Тип вставляемого входа/выхода. Данный тип также определяет длину его бита.

Таблица 101: Окно «Добавить вход/выход» EtherNet/IP

### 5.4.9.2.4. Преобразование входов/выходов протокола EtherNet/IP

На вкладке «Преобразование входов/выходов» (I/O Mapping) в столбце «Переменная» (Variable) отображается имя автоматически сгенерированного экземпляра адаптера в разделе «Объекты IEC» (IEC Objects). Таким образом, данный экземпляр может быть доступен приложению. Здесь переменные проекта преобразуются с входами и выходами адаптера. Опция «Всегда обновлять переменные» (Always update variables) должна быть установлена по умолчанию в «Включить 1» (Enable 1).

### 5.4.9.3. Конфигурация адаптера EtherNet/IP

Для работы адаптера EtherNet/IP требуются модули Ethernet/IP. Модули обеспечивают преобразование входов/выходов, которыми может манипулировать пользовательское приложение через адреса %I или %Q в соответствии со своей конфигурацией (INPUT BYTE («ВХОДНОЙ БАЙТ»), OUTPUT BYTE («ВЫХОДНОЙ БАЙТ») и т.д.).

## 5. КОНФИГУРАЦИЯ

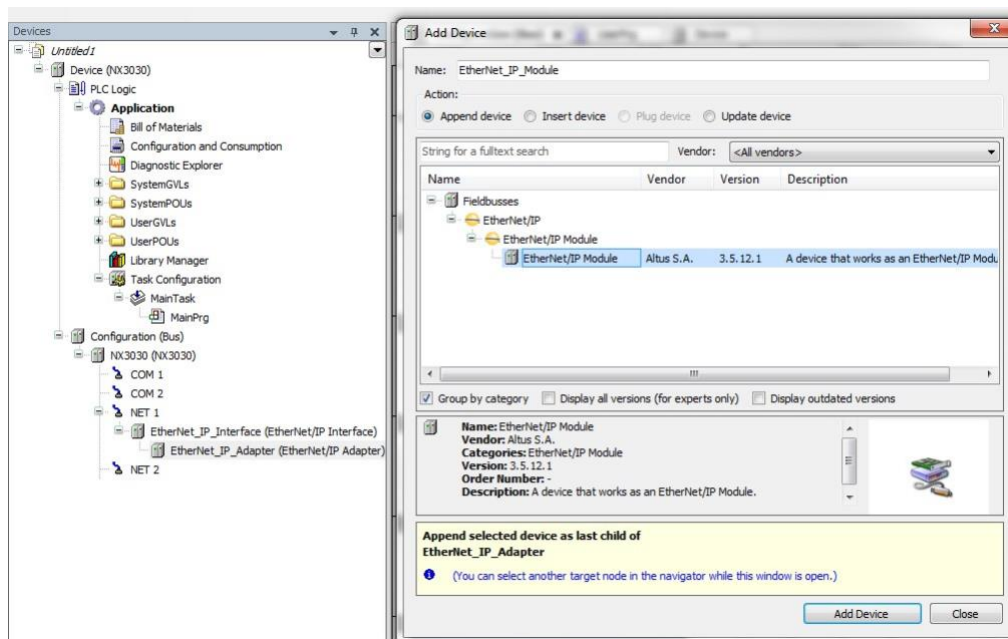


Рисунок 94: Добавление модуля EtherNet/IP под адаптер

### 5.4.9.3.1. Типы модулей

В адаптер можно добавить 18 различных модулей. Девять выходов и девять входов. Они имеют типы BYTE, WORD, DWORD, REAL, SINT, INT, DINT и BIG. Данные типы могут быть выбраны на вкладке «Общее» (General) модуля.

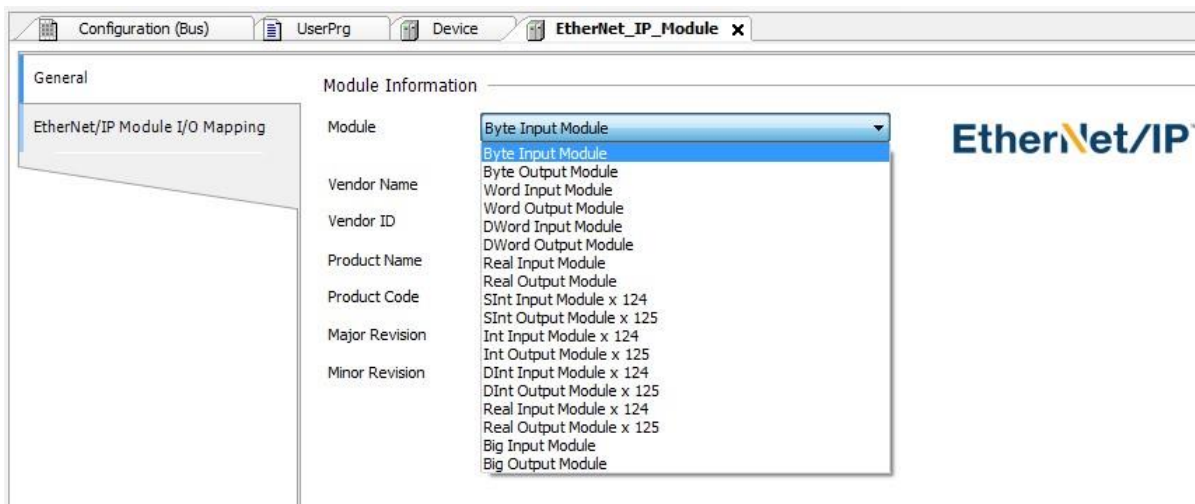


Рисунок 95: Тип модуля EtherNet/IP

### 5.4.9.3.2. Преобразование входов/выходов модуля EtherNet/IP

В столбце «Переменная» (Variable) отображается имя автоматически сгенерированного экземпляра модуля в разделе «Объекты IEC» (IEC Objects). Таким образом, данный экземпляр может быть доступен приложению. Опция «Всегда обновлять переменные» (Always update variables) должна быть оставлена по умолчанию в «Включить 1» (Enable 1).

## 5.5. Производительность связи

### 5.5.1. Сервер MODBUS

Устройства MODBUS, конфигурируемые в центральном процессоре (ЦП) Nexto, работают в фоновом режиме, с приоритетом ниже пользовательского приложения и циклически. Таким образом, их производительность меняется в зависимости от оставшегося времени, принимая во внимание разницу между интервалом и временем работы приложения. Например, устройство MODBUS в приложении, запускаемом каждые 100 мс, при времени выполнения 50 мс будет иметь меньшую производительность, чем приложение, запускаемое с интервалом от 50 до 200 мс. Это происходит потому, что в последнем случае у центрального процессора (ЦП) будет больше времени между каждым циклом MainTask для выполнения задач с более низким приоритетом.

Необходимо также учитывать количество циклов, которое требуется устройству, ведомому устройству либо серверу для отклика на запрос. Для обработки и передачи отклика ведомому устройству MODBUS RTU потребуется несколько циклов (время цикла задачи MODBUS), в то время как задача сервера MODBUS Ethernet (MODBUS Ethernet Server) занимает всего один цикл. Но это минимальное время между получением запроса и откликом. В случае если отправка запроса осуществляется сразу же после выполнения задания, то время цикла MODBUS может быть равно времени цикла, помноженному на 2 или 3, для ведомого устройства MODBUS и времени цикла, помноженного на от 1 до 2, для сервера MODBUS.

В данном случае: Максимальное время отклика = 3 \* (время цикла) + (время выполнения задач) + (время межкадровых символьных операндов) + (задержка отправки).

Например, для задачи сервера MODBUS Ethernet (MODBUS Ethernet Server) с циклом 50 мс приложение, выполняющееся в течение 60 мс каждые 100 мс, способно выполнить только один цикл между каждым циклом приложения. С другой стороны, при использовании того же приложения, выполняющегося в течение 60 мс, но с интервалом 500 мс, MODBUS работает лучше, поскольку пока приложение не запущено, оно будет выполняться каждые 50 мс и только на каждый цикл MainTask у него будет уходить больше времени. Для этих случаев наихудшая производительность будет равна сумме времени выполнения пользовательского приложения и времени цикла задачи MODBUS.

Для ведущего и клиентского устройств принцип работы точно такой же, но с учетом времени выполнения упорядоченного опроса отношения MODBUS, а не времени цикла задачи MODBUS. Для этих случаев наихудшее выполнение отношения будет происходить после времени выполнения упорядоченного опроса, наряду с временем выполнения пользовательского приложения.

Важно подчеркнуть, что количество работающих устройств MODBUS также изменяет его производительность. В пользовательском приложении с временем выполнения 60 мс и интервалом 100 мс центральному процессору (ЦП) остается 40 мс для выполнения всех задач с более низким приоритетом. Поэтому производительность центрального процессора с одним сервером MODBUS Ethernet (MODBUS Ethernet Server) будет выше, чем у центрального процессора, использующего четыре таких устройства.

#### 5.5.1.1. Локальные интерфейсы центрального процессора (ЦП)

В случае устройства сервера MODBUS Ethernet (MODBUS Ethernet Server) можно утверждать, что оно способно отвечать на  $x$  количество запросов в секунду. Либо, другими словами, сервер способен передавать  $n$  байт в секунду, в зависимости от размера каждого запроса. Чем меньше время цикла задачи сервера MODBUS, тем больше влияние количества соединений на скорость отклика. Тем не менее, для времени цикла, составляющего менее 20 мс, это влияние нелинейно, и, в порядке ознакомления, следует ознакомиться с приведенной ниже таблицей.

В таблице ниже показано количество запросов, на которые способен ответить сервер MODBUS, подключенный к локальному интерфейсу сети Ethernet, в зависимости от времени цикла, настроенного для задачи MODBUS, и количества активных соединений:

Число активных соединений	Ответы на заявки в секунду при цикле задач MODBUS на уровне 5 мс	Ответы на заявки в секунду при цикле задач MODBUS на уровне 10 мс	Ответы на заявки в секунду при цикле задач MODBUS на уровне 20 мс
1 соединение	185	99	50
2 соединения	367	197	100
4 соединения	760	395	200

## 5. КОНФИГУРАЦИЯ

7 соединений	1354	695	350
10 соединений	1933	976	500

Таблица 102: Скорость передачи данных сервера MODBUS по локальному интерфейсу

**ВНИМАНИЕ:** Приведенные в данном разделе характеристики связи являются лишь примерами, использующими центральный процессор (ЦП) с одним устройством сервера MODBUS TCP (MODBUS TCP Server), без какой-либо логики, выполняемой в приложении, которая могла бы задержать связь. Таким образом, данные показатели следует рассматривать как максимальные.

Для времени цикла, равного или превышающего 20 мс, увеличение скорости отклика происходит линейно и может быть рассчитано в соответствии с уравнением:

$$N = C \times (1 / T)$$

Где:

N - среднее количество откликов в секунду; C - количество активных соединений; T - интервал задачи MODBUS в секундах.

В качестве примера сервера MODBUS, имеющего только одно активное соединение и время цикла 50 мс, получаем:

$$C = 1; T = 0,05 \text{ с};$$

$$N = 1 \times (1 / (0,05))$$

$$N = 20$$

То есть в данной конфигурации сервер MODBUS отвечает в среднем на 20 запросов в секунду.

Если полученное значение умножить на количество байт в каждой заявке, то получим скорость передачи n байт в секунду.

### 5.5.2. Сервер OPC DA (OPC DA Server)

Производительность связи с сервером OPC DA (OPC DA Server) подвергалась проверке путем создания структурных единиц программы (POU) с 1 000 переменных INT в каждой. Все сценарии были испытаны с профилем *Single* («Одиночный») и интервалом MainTask, равным 100 мс. Связь была включена с помощью атрибута 'symbol' := 'readwrite', чтобы данные были доступны серверу OPC DA (OPC DA Server). Измерения проводились при отключенном от центрального процессора MasterTool, а длительность MainTask составляла 5%, 50% и 80% от сконфигурированного Интервала, как показано в таблице ниже.

На стороне клиента OPC использовался драйвер системы SCADA (автоматизированная система управления технологическим процессом). Сконфигурированное время обновления составляло 50 мс. Результаты работы в этих условиях приведены в таблице ниже.

Общее количество переменных в проекте ПЛК	Время обновления переменных на клиенте OPC DA		
	5% занятости ЦП	50% занятости ЦП	80% занятости ЦП
1000	600 мс	800 мс	1400 мс
2000	800 мс	900 мс	2800 мс
5000	1000 мс	2000 мс	6500 мс
10000	2000 мс	4000 мс	13700 мс
Общее количество переменных в проекте ПЛК	Время обновления переменных на клиенте OPC DA		
	5% занятости ЦП	50% занятости ЦП	80% занятости ЦП
15000	3200 мс	6400 мс	20000 мс
20000	4000 мс	8100 мс	25000 мс

Таблица 103: Скорость передачи данных сервера OPC DA



## 5. КОНФИГУРАЦИЯ

### 5.5.3. Сервер OPC UA

Для выполнения испытаний производительности связи с сервером OPC UA (performance tests) были созданы проекты для ПЛК с объявлением переменных типа INT.

Все сценарии испытаний имеют следующие общие характеристики:

- Проекты с *машинным профилем (Machine Profile)* и интервалом MainTask, настроенным на 100 мс;
- Все переменные типа INT модифицируются через каждые 100 мс. В протоколе OPC UA клиентам OPC UA сообщается только о тех переменных, которые изменяются. В приложениях, где часто изменяется небольшое количество переменных, производительность связи по протоколу OPC UA значительно повышается;
- Длительность MainTask установлена на потребление 10%, 30% и 60% от настроенного интервала в 100 мс. Если длительность MainTask крайне велика, то для связи OPC UA остается мало вычислительных мощностей, что может существенно снизить ее производительность;
- Параметр «Интервал публикации» (*Publishing Interval*) был настроен на клиенте на 1000 мс, что означает, что переменные, изменяющие свое значение, должны сообщаться клиенту каждые 1000 мс. Очень низкое значение данного параметра может существенно снизить производительность связи OPC UA;
- Параметр «Интервал выборки» (*Sampling Interval*) настроен на клиенте в 500 мс, что означает, что контроллер (сервер) будет определять изменения значений переменных каждые 500 мс, чтобы знать, какие из них следует сообщать клиенту. Очень низкое значение данного параметра может существенно снизить производительность обмена данными по протоколу OPC UA. Рекомендуется, чтобы оно составляло не менее половины «Интервала публикации» (*Publishing Interval*), а как максимум – было равно «Интервалу публикации» (*Publishing Interval*);
- Шифрование не было сконфигурировано. Если оно сконфигурировано, то производительность связи OPC UA несколько снизится;
- Был подключен только один клиент OPC UA. Подключение нескольких клиентов может значительно снизить производительность связи OPC UA.

В таблице ниже приведен интервал между сообщениями (значение которого должно составлять 1000 мс в соответствии с сконфигурированным *Интервалом публикации (Publishing Interval)*):

Общее количество переменных INT в проекте ПЛК	Время обновления переменной в клиенте OPC UA (OPC UA Client)		
	10% занятости ЦП	30% занятости ЦП	60% занятости ЦП
1000	1010 мс	1014 мс	1040 мс
2000	1013 мс	1017 мс	1472 мс
5000	1321 мс	1405 мс	2351 мс

Таблица 104: Скорость передачи данных сервера OPC UA

**ВНИМАНИЕ:**

В указаниях по применению NAR165 более подробно анализируется производительность связи OPC UA, в том числе рассматривается потребление пропускной способности канала связи локальной сети Ethernet. Помимо этого, в данных указаниях по применению рассматриваются концепции работы протокола OPC UA.

## 5.6. Эксплуатационные характеристики системы

В тех случаях, когда в приложении имеется только одна пользовательская задача MainTask, отвечающая за выполнение одного блока программирования типа «Программа» (Program), называемого MainPrg (как в профиле Single («Одиночный»)), ПЛК затрачивает определенное время на обработку задачи. Это время мы называем *временем выполнения (Execution Time)*.

Среднее время выполнения задачи в приложении можно узнать с помощью MasterTool IEC XE в элементе «Устройство» (*Device*) *дерева устройств (Devices Tree)* следующим образом:

Логика ПЛК (PLC Logic)-> Приложение (Application)-> Конфигурация задачи на вкладке «Монитор», столбец «Среднее время цикла» (*Task Configuration in the Monitor tab, Average Cycle Time column*).

## 5. КОНФИГУРАЦИЯ

Пользователь должен обращать внимание на *время цикла*, чтобы оно не превышало 80% от интервала, заданного в пользовательской задаче MainTask. Например, в приложении, где интервал составляет 100 мс, подходящее *время цикла* - до 80 мс. Это связано с тем, что центральному процессору (ЦП) необходимо время для выполнения других задач, таких как обработка данных от каналов связи, обработка дисплея и карты памяти, и данные задачи имеют место быть в диапазоне (оставшиеся 20% *времени цикла*).

### ВНИМАНИЕ:

При очень высоком значении времени цикла (как правило, более 300 мс), даже при соблюдении значения 80%, возможно уменьшение времени отклика дисплея и кнопки диагностики. В случае, если значение 80% не соблюдается, а время выполнения пользовательской задачи приближается к значению интервала, установленного для основной задачи, либо превышает его, дисплей и кнопка диагностики не могут реагировать, поскольку их приоритет в работе системы ниже, чем у пользовательских задач. В случае если в центральный процессор (ЦП) загружено приложение с ошибками, может потребоваться его перезапуск без загрузки данного приложения, как показано в разделе «Системный журнал».

### ВНИМАНИЕ:

Системные журналы центрального процессора (ЦП) серии Nexto, начиная с версии микропрограммы (прошивки) 1.4.0.33, теперь перезагружаются в случае сброса центрального процессора либо перезагрузки *системы поддержки выполнения программ*, то есть при наступлении одного из данных условий можно просмотреть старые журналы.

### 5.6.1. Время сканирования входов/выходов

Для проекта, в котором используются модули цифрового ввода/вывода, будучи вставленными в шину и объявленными в проекте, время выполнения MainTask будет увеличиваться в зависимости от количества модулей. В таблице, приведенной ниже, показано среднее время, которое добавляется к MainTask:

Объявленные модули на шине	Добавленное время во время цикла (MainTask) (мкс)
5	300
10	700
20	1000

Таблица 105: Время сканирования входов/выходов

В проектах, использующих удаленные входы/выходы, например, с помощью модуля ведущего устройства PROFIBUS-DP (PROFIBUS-DP Master) NX5001, для получения информации о производительности и влиянии модуля на выполнение пользовательских задач необходимо обратиться к руководству соответствующего модуля.

## 5.7. Часы реального времени (RTC)

Центральные процессоры (ЦП) серии Nexto имеют внутренние часы, которые можно использовать с помощью библиотеки *NextoStandard.lib*. Данная библиотека автоматически загружается при создании нового проекта (для выполнения процедуры вставки библиотеки см. раздел «Библиотеки»). На рисунке, приведенном ниже, показано, как включить данные блоки в проект:

## 5. КОНФИГУРАЦИЯ

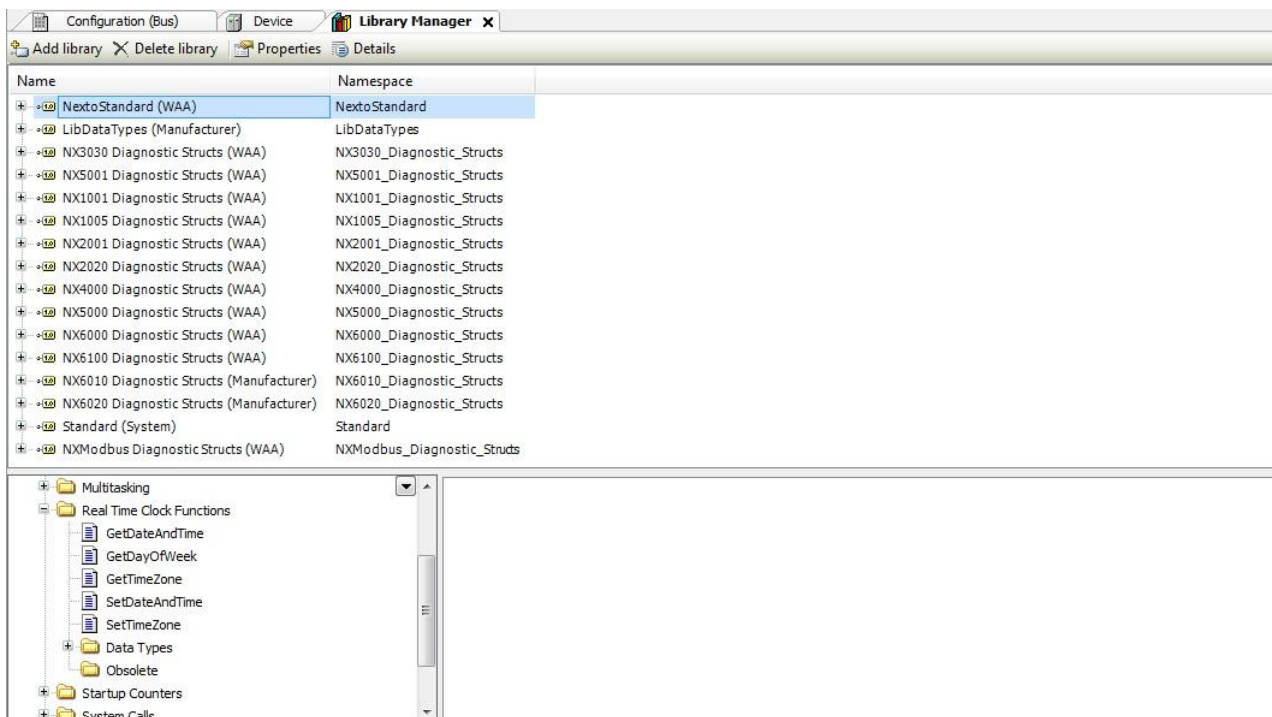


Рисунок 96: Блоки для считывания и записи часов

### ВНИМАНИЕ:

Функциональные блоки считывания и записи часов реального времени (RTC), ранее доступные в версии 2.00 MasterTool IEC XE или более ранней, устарели: начиная с версии 2.00 или более поздней, следующие блоки больше не используются: *NextoGetDateAndTime*, *NextoGetDateAndTimeMs*, *NextoGetTimeZone*, *NextoSetDateAndTime*, *NextoSetDateAndTimeMs* и *NextoSetTimeZone*.

### 5.7.1. Функциональные блоки для считывания и записи часов реального времени (RTC)

Среди прочих функциональных блоков есть очень важные, используемые для считывания показаний часов (*GetDateAndTime*, *GetDayOfWeek* и *GetTimeZone*) и для конфигурирования новых данных даты и времени (*SetDateAndTime* и *SetTimeZone*). Данные функции всегда используют местное время, то есть учитывают значение, определяемое часовым поясом (*Time Zone*).

Процедуры конфигурирования данных двух блоков описаны ниже.

#### 5.7.1.1. Функциональные блоки для считывания часов реального времени (RTC)

Считывание показаний часов может осуществляться с помощью следующих функций:

##### 5.7.1.1.1. Функциональный блок *GetDateAndTime*

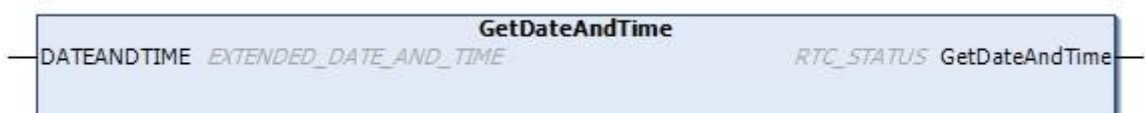


Рисунок 97: Считывание даты и часов

Входные параметры	Тип	Описание
-------------------	-----	----------

## 5. КОНФИГУРАЦИЯ

DATEANDTIME	EXTENDED_DATE_AND_TIME	Данная переменная возвращает значение даты и часа часов реального времени (RTC) в формате, приведенном в таблице 115.
-------------	------------------------	---

Таблица 106: Входные параметры GetDateAndTime

Выходные параметры	Тип	Описание
GETDATEANDTIME	RTC_STATUS	Возвращает состояние ошибки функции, см. таблицу 117.

Таблица 107: Выходные параметры GetDateAndTime

Пример использования на языке структурированного текста (ST):

```

ПРОГРАММА UserPrg
VAR
Результат: RTC_STATUS;
DATEANDTIME : EXTENDED_DATE_AND_TIME; xEnable : BOOL;
END_VAR
-----
ЕСЛИ xEnable = TRUE («ИСТИНА») ТОГДА
Результат:= GetDateAndTime(DATEANDTIME); xEnable := FALSE
(«ЛОЖЬ»);
END_IF
    
```

### 5.7.1.1.2. Функциональный блок GetTimeZone

Следующая функция считывает конфигурацию часового пояса. Данная функция непосредственно связана со временем в часовом поясе при синхронизации по протоколу SNTP (простой протокол сетевого времени):

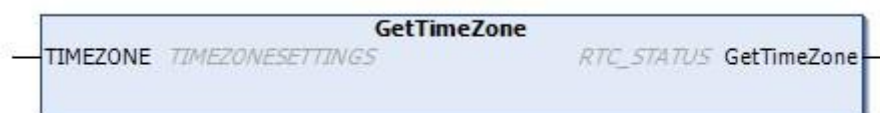


Рисунок 98: Конфигурация считывания часового пояса

Входные параметры	Тип	Описание
TIMEZONE	TIMEZONESETTINGS	Данная переменная представляет собой показания конфигурации часового пояса.

Таблица 108: Входные параметры GetTimeZone

Выходные параметры	Тип	Описание
GetTimeZone	RTC_STATUS	Возвращает состояние ошибки функции, см. таблицу 117.

Таблица 109: Выходные параметры GetTimeZone

## 5. КОНФИГУРАЦИЯ

Пример использования на языке структурированного текста (ST):

```
ПРОГРАММА UserPrg
VAR
GetTimeZone_Status : RTC_STATUS; TimeZone :
TIMEZONESETTINGS; xEnable : BOOL;
END_VAR
-----
ЕСЛИ xEnable = TRUE («ИСТИНА») ТОГДА
GetTimeZone_Status := GetTimeZone(TimeZone); xEnable := FALSE («ЛОЖЬ»);
END_IF
```

### 5.7.1.1.3. Функция *GetDayOfWeek*

Функция *GetDayOfWeek* используется для считывания дня недели.



Рисунок 99: Считывание дня недели

Выходные параметры	Тип	Описание
GetDayOfWeek	DAYS_OF_WEEK	Возвращает день недели. См. раздел <a href="#">116</a> .

Таблица 110: Выходные параметры *GetDayOfWeek*

При вызове функция считывает день недели и заполняет структуру *DAYS\_OF\_WEEK*.  
Пример использования на языке структурированного текста (ST):

```
ПРОГРАММА UserPrg
VAR
DayOfWeek : DAYS_OF_WEEK;
END_VAR
-----DayOfWeek := GetDayOfWeek();
```

### 5.7.1.2. Функциональные блоки и функции записи и конфигурации часов реального времени (RTC)

Настройки часов осуществляются с помощью функций и функциональных блоков следующим образом:

#### 5.7.1.2.1. Функция *SetDateAndTime*

Функция *SetDateAndTime* используется для записи настроек на часы. Как правило, точность составляет порядка сотен миллисекунд.

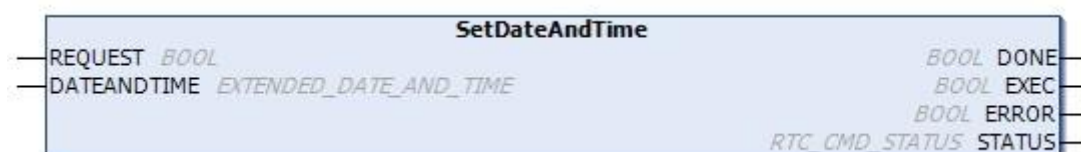


Рисунок 100: Установка даты и времени

Входные параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, получив нарастающий фронт импульса, разрешает запись значений часов.
DATEANDTIME	EXTENDED_DATE _AND_TIME	Получает значения даты и часа с миллисекундами. См. раздел <a href="#">115</a> .

Таблица 111: Входные параметры SetDateAndTime

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная, если она равна true («истина»), указывает на то, что действие было успешно завершено.
EXEC	BOOL	Данная переменная, если она равна true («истина»), указывает на то, что функция обрабатывает значения.
ERROR	BOOL	Эта переменная, если она равна true, указывает на ошибку во время выполнения операции записи.
STATUS	RTC_CMD_STATUS	Возвращает ошибку, возникшую при конфигурировании. См. таблицу <a href="#">117</a> .

Таблица 112: Выходные параметры SetDateAndTime

При появлении нарастающего фронта импульса на входе *REQUEST* («ЗАПРОС») функциональный блок запишет в часы новые значения *DATEANDTIME*. Если запись прошла успешно, то выход *DONE* («ВЫПОЛНЕНО») будет равен *TRUE* («ИСТИНА»). В противном случае выход *ERROR* («ОШИБКА») будет равен *TRUE* («ИСТИНА»), а ошибка отобразится в переменной *STATUS* («СОСТОЯНИЕ»).

Пример использования на языке структурированного текста (ST):

```

ПРОГРАММА UserPrg
VAR
SetDateAndTime : SetDateAndTime; xRequest : BOOL;
DateAndTime : EXTENDED_DATE_AND_TIME; xDone :
BOOL; xExec : BOOL; xError : BOOL; xStatus :
RTC_STATUS;
END_VAR
-----
ЕСЛИ xRequest ТОГДА
  SetDateAndTime.REQUEST:=TRUE («ИСТИНА»);
  SetDateAndTime.DATEANDTIME:=DateAndTime; xRequest:= FALSE («ЛОЖЬ»);
END_IF
SetDateAndTime();
SetDateAndTime.REQUEST:= FALSE («ЛОЖЬ»);
IF SetDateAndTime.DONE THEN xExec:=SetDateAndTime.EXEC;
  xError:=SetDateAndTime.ERROR; xStatus:=SetDateAndTime.STATUS;
END_IF

```

ВНИМАНИЕ:

## 5. КОНФИГУРАЦИЯ

При попытке записи значений времени, выходящих за пределы диапазона часов реального времени (RTC), значения преобразуются в действительные, если они не выходят за пределы допустимого диапазона от 01/01/2000 до 12/31/2035. Например, если пользователь попытается записать значение 2000 мс, то оно будет преобразовано в 2 секунды, если записать значение 100 секунд, то оно будет преобразовано в 1 мин 40 секунд. Если записать значение типа 30 часов, то оно будет преобразовано в 1 день и 6 часов, и так далее.

### 5.7.1.2.2. Функциональный блок SetTimeZone

Следующий функциональный блок выполняет запись настроек часового пояса:



Рисунок 101: Запись настроек часового пояса

Входные параметры	Тип	Описание
TIMEZONE	TIMEZONESETTINGS	Структура с настраиваемым часовым поясом. См. таблицу <a href="#">118</a> .

Таблица 113: Входные параметры SetTimeZone

Выходные параметры	Тип	Описание
SetTimeZone	RTC_STATUS	Возвращает ошибку, возникшую при считывании/настройке. См. таблицу <a href="#">117</a> .

Таблица 114: Выходные параметры SetTimeZone

При вызове функция конфигурирует *TIMEZONE* с новой конфигурацией часового пояса системы. Результаты конфигурирования возвращаются функцией.

Пример использования на языке структурированного текста (ST):

```
ПРОГРАММА UserPrg
VAR
Статус : RTC_STATUS;
TimeZone : TIMEZONESETTINGS; xWrite : BOOL;
END_VAR
-----//FB SetTimeZone
ЕСЛИ (xWrite = TRUE («ИСТИНА»)) ТОГДА
Статус := SetTimeZone(TimeZone);
  ЕСЛИ Status = RTC_STATUS.NO_ERROR ТОГДА
    xWrite := FALSE («ЛОЖЬ»); END_IF
END_IF
```

ВНИМАНИЕ:

## 5. КОНФИГУРАЦИЯ

Для работы часов необходимо использовать значения времени и даты в следующем допустимом диапазоне: 00:00:00 часов от 01/01/2000 до 12/31/2035 23:59:59 часов, иначе через выходной параметр *STATUS* («СТАТУС») будет сообщено об ошибке. Подробную информацию о выходном параметре *STATUS* («СТАТУС») см. В разделе «*RTC\_STATUS*».

### 5.7.2. Структуры данных часов реального времени (RTC)

Функциональные блоки считывания и настройки часов реального времени (RTC) центральных процессоров (ЦП) серии Nexто используют в своей конфигурации следующие структуры данных:

#### 5.7.2.1. Структура «EXTENDED\_DATE\_AND\_TIME»

Данная структура используется для хранения даты часов реального времени (RTC) при использовании функциональных блоков для считывания/установки даты с точностью до миллисекунды. Ее описание приведено 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	Хранит секунды установленной даты.
	WORD	wMilliseconds	Сохраняет миллисекунды установленной даты.

Таблица 115: Структура «EXTENDED\_DATE\_AND\_TIME»

#### 5.7.2.2. Структура «DAYS\_OF\_WEEK»

Данная структура используется для хранения дня недели:

Счетное значение	Значение	Описание
DAYS_OF_WEEK	0	INVALID_DAY
	1	SUNDAY (ВОСКРЕСЕНЬЕ)
	2	MONDAY (ПОНЕДЕЛЬНИК)
	3	TUESDAY (ВТОРНИК)
	4	WEDNESDAY (СРЕДА)
	5	THURSDAY (ЧЕТВЕРГ)
	6	FRIDAY (ПЯТНИЦА)
	7	SATURDAY (СУББОТА)

Таблица 116: Структура «DAYS\_OF\_WEEK»

#### 5.7.2.3. Структура RTC\_STATUS

Данный элемент перечисления используется для возврата типа ошибки при установке или считывании часов реального времени (RTC) и описан в следующей таблице:

Энумератор (перечислитель)	Значение	Описание
RTC_STATUS	NO_ERROR (0)	Ошибка отсутствует.
	UNKNOWN_COMMAND (1)	Неизвестная команда.
	DEVICE_BUSY (2)	Устройство занято.
	DEVICE_ERROR (3)	Устройство с ошибкой.



ERROR_READING_OSF (4)	Ошибка при считывании флага действительной даты и часа.
ERROR_READING_RTC (5)	Ошибка при считывании даты и часа.
ERROR_WRITING_RTC (6)	Ошибка при записи даты и часа.
ERROR_UPDATING_SYSTEM_TIME (7)	Ошибка при обновлении даты и часа в системе.
INTERNAL_ERROR (8)	Внутренняя ошибка.
INVALID_TIME (9)	Недопустимые дата и час.
INPUT_OUT_OF_RANGE (10)	Выход за пределы допустимых для системы даты и часа.
SNTP_NOT_ENABLE (11)	Ошибка возникает, когда служба простого протокола сетевого времени (SNTP) не включена и предпринимается попытка изменить часовой пояс.

Таблица 117: Структура «RTC\_STATUS»

#### 5.7.2.4. Структура TIMEZONESETTINGS

Данная структура используется для хранения значения часового пояса в запросах считывания/установки функциональных блоков часов реального времени (RTC) и описана в следующей таблице:

Структура	Тип	Переменная	Описание
TIMEZONESETTINGS	INT	iHour	Установить час часового пояса.
	INT	iMinutes	Установить минуту часового пояса.

Таблица 118: Структура «TIMEZONESETTINGS»

Примечание:

Функциональные блоки для записи и считывания даты и времени: не указаны различные библиотеки *NextoStandard*, имеющие функциональные блоки или функции, которые могут осуществлять обеспечение доступа к считыванию и записи даты и времени в системе. Библиотека *NextoStandard* имеет соответствующие интерфейсы для записи и считывания даты и времени в системе, соответственно, и для информирования правильной диагностики.

## 5.8. Память пользовательских файлов

В центральных процессорах (ЦП) серии Nexto имеется область памяти, предназначенная для хранения общих данных. Иными словами, пользователь может хранить в памяти центрального процессора (ЦП) несколько файлов проектов любого формата. Данная область памяти зависит от модели центрального процессора (см. раздел «Память»).

Для того чтобы воспользоваться данной областью, пользователь должен зайти в проект в программном обеспечении MasterTool IEC XE и щелкнуть по *дереву устройств (Devices Tree)*, расположенном слева от программы. Дважды щелкнуть по элементу «Устройство» (*Device*) и, выбрав центральный процессор (ЦП) на открывшейся вкладке «Настройки связи» (*Communication Settings*), выбрать вкладку «Файлы» (*Files*) и нажать кнопку «Обновить» (*Refresh*), как в столбце «Компьютерные файлы» (*computer files*) (слева), так и в столбце «Файлы центрального процессора (ЦП)» (*CPU files*) (справа), как показано на рисунке, приведенном ниже.

## 5. КОНФИГУРАЦИЯ

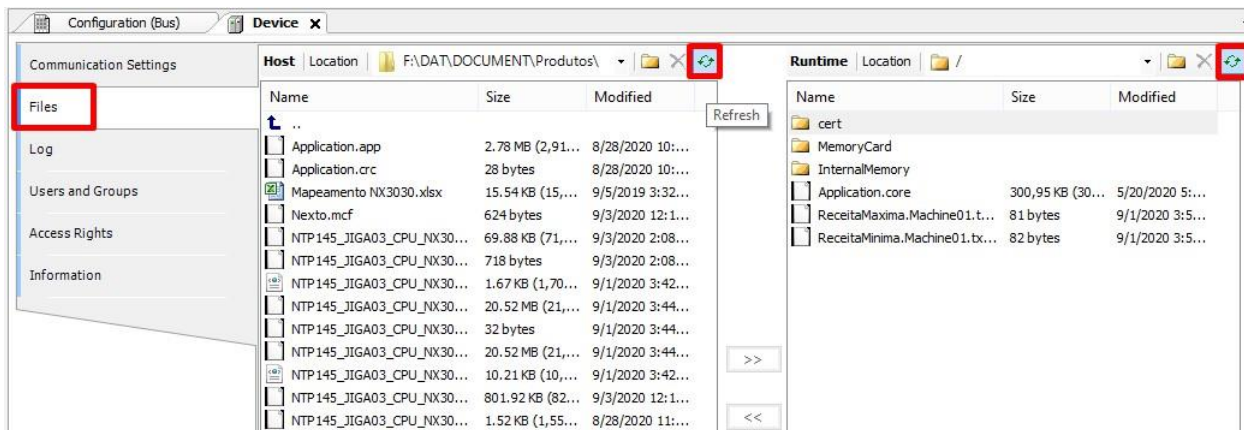


Рисунок 102: Доступ к пользовательским файлам

После обновления столбца файлов центрального процессора (ЦП) будет показан корневой каталог файлов, хранящихся в центральном процессоре (ЦП). Затем можно будет выбрать папку, в которую будут перенесены файлы. По умолчанию для хранения файлов во внутренней памяти центрального процессора используется папка «*InternalMemory*», поскольку передача файлов в корневой каталог невозможна. При необходимости пользователь может создать другие папки в корневом каталоге или вложенные папки внутри папки «*InternalMemory*».

Для того чтобы выполнить передачу файла с микрокомпьютера на центральный процессор (ЦП), достаточно выбрать нужный файл в левом столбце и нажать клавишу «*«*», расположенную в центре экрана, как показано на рисунке, приведенном ниже. Время загрузки будет зависеть от размера файла и времени цикла (выполнения) текущего приложения центрального процессора, и может занять несколько минут.

Для выполнения передачи данных пользователю не обязательно находиться в режиме «*Выполнение*» (*Run*) или подключаться к центральному процессору (ЦП), поскольку он имеет возможность автоматического подключения при выполнении пользователем передачи данных.

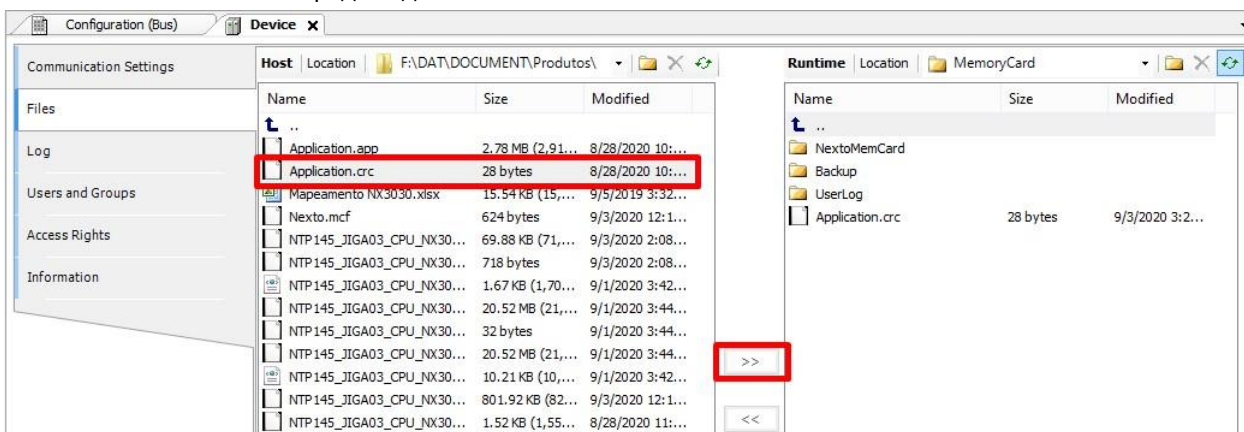


Рисунок 103: Передача файлов




### ВНИМАНИЕ:

Файлы, содержащиеся в папке проекта, созданного MasterTool IEC XE, имеют специальные имена, зарезервированные системой, поэтому не могут быть переданы через вкладку «*Файлы*» (*Files*). Если пользователь хочет перенести проект в память пользователя, необходимо сжать папку, а затем загрузить сжатый файл (например, \*.zip).

В случае необходимости переноса документов из центрального процессора (ЦП) на ПК, на котором установлено программное обеспечение MasterTool IEC XE, пользователь должен выполнить процедуру, очень похожую на описанную ранее: выбрать файл из правого столбца и нажать кнопку «*«*», расположенную в центре экрана.

Помимо этого, в области сохранения файлов пользователю доступны следующие опции:

## 5. КОНФИГУРАЦИЯ

-  Новый каталог : позволяет создать новую папку в пользовательской области памяти.
-  Удалить элемент: позволяет удалить файлы, исключенные из папок в пользовательской области памяти.
-  Обновить : позволяет обновлять на экране MasterTool IEC XE файлы, находящиеся в пользовательской области памяти и на компьютере.

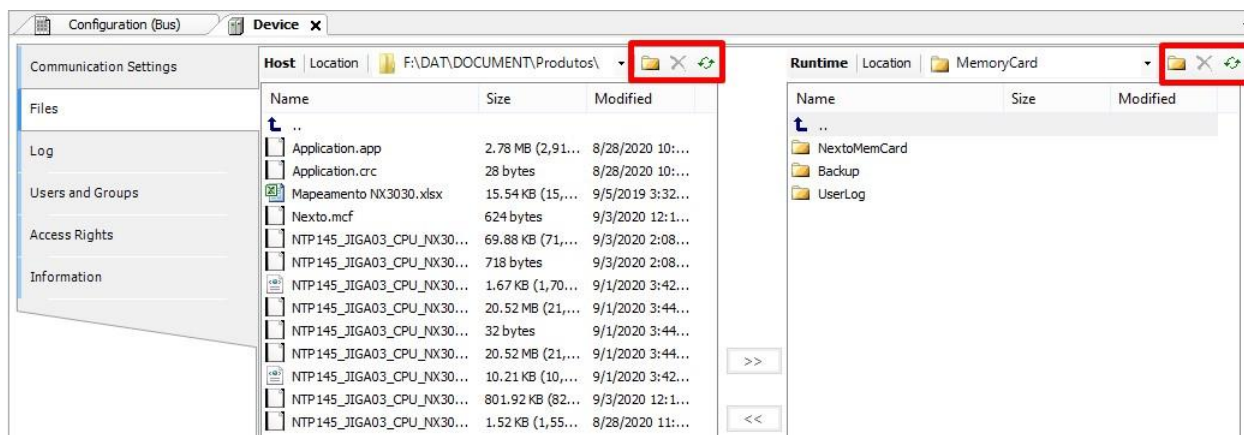


Рисунок 104: Варианты использования

### ВНИМАНИЕ:

Для центрального процессора (ЦП), находящегося в режиме останова или без приложения, скорость передачи данных во внутреннюю память составляет примерно 150 Кбайт/с.

## 5.9. Информационное и конфигурационное меню центрального процессора (ЦП)

Доступ к *информационному меню, конфигурации* центрального процессора (ЦП) Nexto и подробной диагностике осуществляется по уровням, и для получения доступа к информации меню, смены уровня и изменения конфигурации необходимо долгое нажатие на кнопку диагностики, а для перемещения по пунктам одного уровня – короткое нажатие на кнопку диагностики. Для проверки функционирования и различий между типами нажатия диагностических кнопок см. раздел «*Диагностика одним касанием (One Touch Diag)*».

В таблице ниже приведены уровни меню и каждый тип экрана, доступные на центральных процессорах, их информативность, возможность конфигурирования или возврата на уровень.

Уровень 1	Уровень 2	Уровень 3	Тип
АППАРАТНОЕ ОБЕСПЕЧЕНИЕ	ТЕМПЕРАТУРА	-	Информационный
	КОНТРАСТНОСТЬ	УРОВЕНЬ КОНТРАСТНОСТИ	Конфигурируемый
	ДАТА И ВРЕМЯ	-	Информационный
	НАЗАД	-	Уровень возврата
ЯЗЫКИ	АНГЛИЙСКИЙ	>АНГЛИЙСКИЙ	Конфигурируемый
	ПОРТУГАЛЬСКИЙ	>ПОРТУГАЛЬСКИЙ	Конфигурируемый
	ИСПАНСКИЙ	>ИСПАНСКИЙ	Конфигурируемый
	НАЗАД	-	Уровень возврата
СЕТЬ	IP-АДРЕС NET 1		Информационный
	МАСКА NET 1		Информационный
	НАЗАД		Уровень возврата

## 5. КОНФИГУРАЦИЯ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	МИКРОПРОГРАММА (ПРОШИВКА)	-	Информационный
	ПРОГРАММА-ЗАГРУЗЧИК		Информационный
	ВСПОМОГАТЕЛЬНЫЙ ПРОЦЕССОР		Информационный
	НАЗАД		Уровень возврата
НАЗАД	-	-	Уровень возврата

Таблица 119: Уровни меню центрального процессора (ЦП)

Как показано в таблице 119, между доступными опциями для визуализации и изменения находятся основные данные, необходимые пользователю, а именно:

- Информация о ресурсах аппаратного обеспечения:
  - «ТЕМПЕРАТУРА» (TEMPERATURE) – Внутренняя температура центрального процессора (ЦП) (Пример: 36 C 97 F)
  - «КОНТРАСТНОСТЬ» (CONTRAST) – Настройка контрастности фронтального дисплея центрального процессора (ЦП)
  - «ДАТА И ВРЕМЯ» (DATE AND TIME) – Дата и время, установленные в центральном процессоре (ЦП) (Пример: 2001.01.31 00:00):
- Изменение языка меню на центральном процессоре (ЦП):
  - «ПОРТУГАЛЬСКИЙ» (PORTUGUES) – Изменяет язык на португальский
  - «АНГЛИЙСКИЙ» (ENGLISH) – Изменяет язык на английский
  - «ИСПАНСКИЙ» (ESPANOL) – Изменяет язык на испанский
- Визуализация информации о сети, установленной в устройстве:
  - «IP-АДРЕС NET 1» (NET 1 IP ADDR.) – Адрес (Пример: 192.168.0.1)
  - «МАСКА NET 1» (NET 1 MASK) – Маска подсети (Пример: 255.255.255.0)
- Информация о версиях программного обеспечения:
  - «МИКРОПРОГРАММА (ПРОШИВКА)» (FIRMWARE) – Версия программного обеспечения центрального процессора (ЦП) (Пример: 1.0.0.0)
  - «ПРОГРАММА-ЗАГРУЗЧИК» (BOOTLOADER) – Версия программы-загрузчика центрального процессора (ЦП) (Пример: 1.0.0.0)
  - «ВСПОМОГАТЕЛЬНЫЙ ПРОЦЕССОР» (AUX. PROC.) – Версия вспомогательного процессора центрального процессора (ЦП) (Пример: 1.0.0.0)

На рисунке, приведенном ниже, описан пример работы с меню центральных процессоров (ЦП) Nexto через процедуру настройки контрастности с экрана *состояния* (Statusscreen). Помимо этого, в целях упрощения настройки можно определить все уровни экрана и тип касания для перемещения по ним, а также изменить другие параметры, такие как *Язык* (Language) и *Карта памяти* (Memory Card), используя ту же логику доступа. Короткое касание показывает, что контрастность увеличивается (становится более четкой), а при следующем касании после максимального значения она возвращается к минимальному значению (становится менее четкой). Длительное нажатие демонстрирует подтверждение желаемой контрастности и ее возврат к предыдущему уровню.

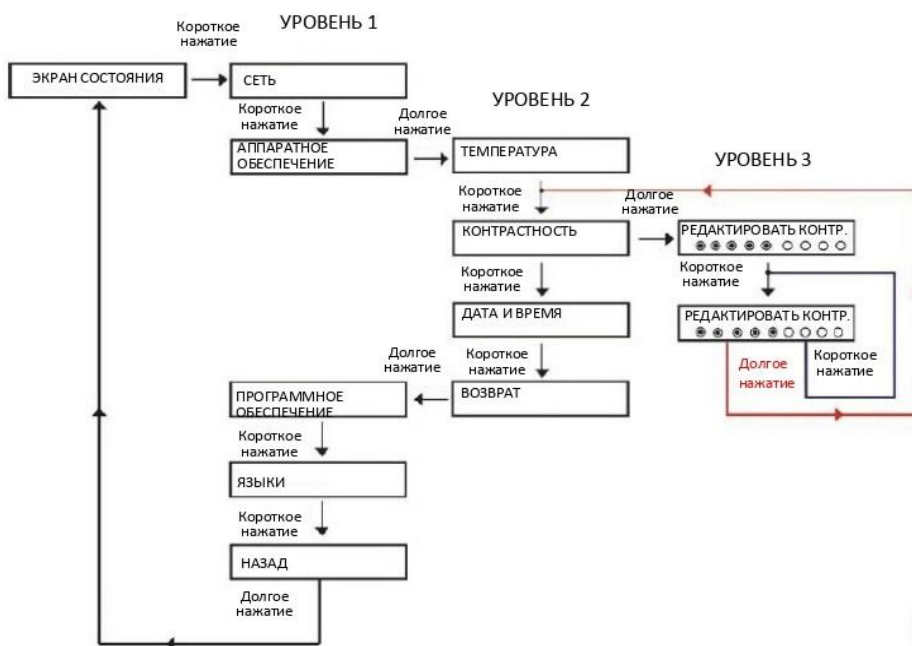


Рисунок 105: Настройка контрастности

Помимо возможности закрытия меню центральных процессоров (ЦП) Nexto путем длительного нажатия на диагностическую кнопку экрана «НАЗАД» (BACK) с уровня 1, существуют и другие условия выхода, которые описаны ниже:

- Короткое нажатие в любой момент на другие модули, имеющиеся на шине, заставляет центральный процессор (ЦП) выйти из меню и показать диагностику нужного модуля.
- Время бездействия на любом уровне более 5 с.

## 5.10. Функциональные блоки и функции

### 5.10.1. 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 собственные протоколы и работать с последовательными портами по своему усмотрению, следуя языкам МЭК 61131-3 (IEC 61131-3), доступным в программе MasterTool IEC XE. Блоки доступны в библиотеке *NextoSerial*, которая должна быть добавлена в проект, чтобы их можно было использовать (процедуру вставки библиотеки см. в Руководстве по программированию MasterTool IEC XE – MP399609, раздел «Библиотека»).

Специальным функциональным блокам для последовательных интерфейсов может потребоваться несколько циклов (последовательных вызовов) для завершения выполнения задачи. Иногда выполнение блока может быть завершено за один цикл, но в общем случае требуется несколько циклов. Выполнение задачи, связанной с блоком, может иметь много этапов, некоторые из которых зависят от внешних событий, что может привести к существенным задержкам. Функциональный блок не может реализовать стандартные подпрограммы, занимающие время в ожидании данных событий, поскольку это приведет к загрузке центрального процессора (ЦП). Решением может быть создание блокирующих функциональных блоков, но это нецелесообразно, поскольку приведет к увеличению сложности пользовательского приложения в связи с тем, что, как правило, многозадачное программирование недоступно. Таким образом, при ожидании внешнего события выполнение последовательных функциональных блоков завершается, и управление возвращается к основной программе. Обработка задачи продолжается в следующем цикле, иными словами, при следующем обращении к блоку.

Прежде чем описывать специальные функциональные блоки для последовательных интерфейсов, необходимо знать типы данных, то есть тип данных, используемый блоками.

Тип данных	Опции	Описание
------------	-------	----------

5. КОНФИГУРАЦИЯ

SERIAL_BAUDRATE	BAUD200	Перечисляет все возможности скорости передачи данных (бит в секунду)
	BAUD300	
	BAUD600	
	BAUD1200	
	BAUD1800	
	BAUD2400	
	BAUD4800	
	BAUD9600	
	BAUD19200	
	BAUD38400	
	BAUD57600	
	BAUD115200	
SERIAL_DATABITS	DATABITS_5	Перечисляет все возможности битов данных.
	DATABITS_6	
	DATABITS_7	
	DATABITS_8	
SERIAL_HANDSHAKE	Определяет все возможности сигналов модема для конфигураций:	
	RS232_RTS	Управляет портом RS-232C центрального процессора (ЦП) Nexto. Передатчик включается для начала передачи и выключается сразу же после ее окончания. Например, может использоваться для управления внешним преобразователем RS232/RS-485.
	RS232_RTS_OFF	Управляет портом RS-232C центрального процессора (ЦП) Nexto. Сигнал RTS всегда выключен.
	RS232_RTS_ON	Управляет портом RS-232C центрального процессора (ЦП) Nexto. Сигнал RTS всегда включен.

Тип данных	Опции	Описание
	RS232_RTS_CTS	Управляет портом RS-232C центрального процессора (ЦП) Nexto. В случае если CTS отключен, включаются часы RTS. Затем ожидает включения CTS для получения передачи, а часы RTS перезапускаются как можно быстрее, по окончании передачи. Например: Управление радиомодемами с помощью одного и того же сигнала модема.
	RS232_MANUAL	Управляет портом RS-232C центрального процессора (ЦП) Nexto. Пользователь отвечает за управление всеми сигналами (RTS, DTR, CTS, DSR, DCD).
SERIAL_MODE	NORMAL_MODE	Нормальный режим работы последовательной связи.

## 5. КОНФИГУРАЦИЯ

	EXTENDED_MODE	Расширенный режим работы последовательной связи, в котором предоставляется информация о принятом кадре (фрейме) данных.
SERIAL_PARAMETERS	Определяет все параметры конфигурации последовательного порта:	
	BAUDRATE	Определено в SERIAL_BAUDRATE.
	DATABITS	Определено в SERIAL_DATABITS.
	STOPBITS	Определено в SERIAL_STOPBITS.
	PARITY	Определено в SERIAL_PARITY.
	HANDSHAKE	Определено в SERIAL_HANDSHAKE.
	UART_RX_THRESHOLD	Количество байтов, которое должно быть получено для генерации нового прерывания универсального асинхронного приемопередатчика (UART). Меньшие значения создают TIMESTAMP («Отметка времени») более точной при использовании режима EXTENDED («Расширенный») и сводят к минимуму ошибки превышения значения. Тем не менее, слишком низкие значения могут привести к слишком большому количеству прерываний и задержке работы центрального процессора (ЦП).
	MODE	Определено в SERIAL_MODE.
	ENABLE_RX_ON_TX	При значении true («истина») все принятые байты во время передачи будут разряжаться, а не поступать на линию RX. Используется для отключения полнодуплексной работы в интерфейсе RS-422.
	ENABLE_DCD_EVENT	Если значение true («истина»), то при изменении DCD генерируется внешнее событие.
ENABLE_CTS_EVENT	Если значение true («истина»), то при изменении CTS генерируется внешнее событие.	
SERIAL_PARITY	PARITY_NONE	Перечисляет все возможности четности.
	PARITY_ODD	
	PARITY_EVEN	
	PARITY_MARK	
	PARITY_SPACE	

Тип данных	Опции	Описание
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).
	COM 2	
SERIAL_RX_CHAR_EXTENDED	Определяет символ в очереди RX в расширенном режиме.	
	RX_CHAR	Байт данных.
	RX_ERROR	Код ошибки.
	RX_TIMESTAMP	Молчание из-за предыдущего символа или из-за другого события, произошедшего до этого

5. КОНФИГУРАЦИЯ

		символа (конфигурация последовательного порта, окончание передачи).
SERIAL_RX_QUEUE_STATUS	Он имеет несколько полей, предоставляющих информацию о состоянии/ошибке очереди RX, которые используются, когда используется обычный формат (без информации об ошибке и отметке времени):	
	RX_FRAMING_ERRORS	Счетчик ошибок кадра: неправильное формирование символов - отсутствие стопового бита, неправильная скорость передачи данных, в том числе - с момента конфигурирования последовательного порта. Возвращается к нулю при достижении максимального значения (65535).
	RX_PARITY_ERRORS	Счетчик ошибок четности, начиная с конфигурации последовательного порта. Возвращается к нулю при достижении максимального значения (65535).
	RX_BREAK_ERRORS	Счетчик ошибок прерывания, начиная с конфигурации последовательного порта, другими словами, активная линия больше символического времени. Возвращается к нулю при достижении максимального значения (65535).
	RX_FIFO_OVERRUN_ERRORS	Счетчик ошибок превышения RX в порядке поступления (FIFO), начиная с момента конфигурации последовательного порта, другими словами, ошибок в RX в порядке поступления (FIFO), настроенного порога. Возвращается к нулю при достижении максимального значения (65535).
	RX_QUEUE_OVERRUN_ERRORS	Счетчик ошибок превышения очереди RX, т.е. максимальное количество символов (1024) было переполнено и данные перезаписываются. Возвращается к нулю при достижении максимального значения (65535).
	RX_ANY_ERRORS	Сумма последних 5 счетчиков ошибок (кадр (фрейм), четность, прерывание, выход за пределы RX в порядке поступления (FIFO), выход за пределы очереди RX).
	RX_REMAINING	Количество символов в очереди RX.

Тип данных	Опции	Описание
SERIAL_STATUS	Список критических кодов ошибок, которые могут быть возвращены последовательным функциональным блоком. Каждый блок возвращает определенные ошибки, которые будут описаны ниже:	
	NO_ERROR	Нет ошибок.



5. КОНФИГУРАЦИЯ

ILLEGAL_*	Возврат параметров с недопустимыми значениями или выходящих за пределы диапазона: <ul style="list-style-type: none"> <li>- SERIAL_PORT</li> <li>- SERIAL_MODE</li> <li>- BAUDRATE</li> <li>- DATA_BITS</li> <li>- PARITY</li> <li>- STOP_BITS</li> <li>- HANDSHAKE</li> <li>- UART_RX_THRESHOLD</li> <li>- TIMEOUT</li> <li>- TX_BUFF_LENGTH</li> <li>- HANDSHAKE_METHOD</li> <li>- RX_BUFF_LENGTH</li> </ul>	
PORT_BUSY	Указывает на то, что последовательный порт используется другим экземпляром.	
HW_ERROR_UART	Обнаружена аппаратная ошибка в универсальном асинхронном приемопередатчике (UART).	
HW_ERROR_REMOTE	Аппаратная ошибка при обмене данными с удаленным последовательным портом.	
CTS_TIMEOUT_ON	Блокировка по превышению лимита времени при ожидании разрешения CTS, в синхронизации интерфейса и периферийного оборудования RS-232 RTS/CTS, в блоке SERIAL_TX.	
CTS_TIMEOUT_OFF	Блокировка по превышению лимита времени при ожидании отключения CTS, в синхронизации интерфейса и периферийного оборудования RS-232 RTS/CTS, в блоке SERIAL_TX.	
TX_TIMEOUT_ERROR	Блокировка по превышению лимита времени при ожидании окончания передачи в SERIAL_TX.	
RX_TIMEOUT_ERROR	Блокировка по превышению лимита времени при ожидании всех символов в блоке SERIAL_RX или блоке SERIAL_RX_EXTENDED.	
FB_SET_CTRL_NOT_ALLOWED	Блок SET_CTRL не может быть использован в случае, если синхронизация интерфейса и периферийного оборудования отличается от RS232_MANUAL.	
FB_GET_CTRL_NOT_ALLOWED	Блок GET_CTRL не может быть использован в случае, если синхронизация интерфейса и периферийного оборудования 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, нормальный режим.	
Тип данных	Опции	Описание
	DCD_INTERRUPT_NOT_ALLOWED	Прерывание по сигналу DCD не может быть включено, если последовательный порт не имеет соответствующего вывода.

## 5. КОНФИГУРАЦИЯ

	CTS_INTERRUPT_NOT_ALLOWED	Прерывание по сигналу CTS не может быть включено, если синхронизация интерфейса и периферийного оборудования отличается от RS232_MANUAL или если последовательный порт не имеет соответствующего вывода.
	DSR_INTERRUPT_NOT_ALLOWED	Прерывание по сигналу DSR не может быть включено, если последовательный порт не имеет соответствующего вывода. (Центральные процессоры Nexto не имеют данного сигнала в локальных портах)
	NOT_CONFIGURED	Функциональный блок не может быть использован до конфигурации последовательного порта.
	INTERNAL_ERROR	Указывает на то, что в последовательном порту возникла внутренняя проблема.
SERIAL_STOPBITS	STOPBITS_1	Перечисляет все возможности стоповых битов.
	STOPBITS_2	
	STOPBITS_1_5	

Таблица 120: Типы данных последовательных функциональных блоков

### 5.10.1.1. Функциональный блок «SERIAL\_CFG»

Данный функциональный блок используется для конфигурирования и инициализации нужного последовательного порта. После вызова данного блока перезапускаются все очереди RX и TX, связанные с последовательными портами, а также порядок поступления (FIFO) RX и TX.



Рисунок 106: Блок последовательной конфигурации

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
PARAMETERS	SERIAL_PARAMETERS	Данная структура определяет параметры конфигурации последовательного порта, описанные в типе данных SERIAL_PARAMETERS.

Таблица 121: Входные параметры SERIAL\_CFG

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).

EXEC	BOOL	Данная переменная равна true («истина») во время выполнения блока. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: <ul style="list-style-type: none"> <li>- NO_ERROR</li> <li>- ILLEGAL_SERIAL_PORT</li> <li>- ILLEGAL_SERIAL_MODE</li> <li>- ILLEGAL_BAUDRATE</li> <li>- ILLEGAL_DATA_BITS</li> <li>- ILLEGAL_PARITY</li> <li>- ILLEGAL_STOP_BITS</li> <li>- ILLEGAL_HANDSHAKE</li> <li>- ILLEGAL_UART_RX_THRESHOLD</li> <li>- PORT_BUSY</li> <li>- HW_ERROR_UART</li> <li>- HW_ERROR_REMOTE</li> <li>- DCD_INTERRUPT_NOT_ALLOWED</li> <li>- CTS_INTERRUPT_NOT_ALLOWED</li> <li>- DSR_INTERRUPT_NOT_ALLOWED</li> </ul>

Таблица 122: Выходные параметры SERIAL\_CFG

Пример использования на языке структурированного текста (ST), после вставки в проект библиотеки Nexto Serial:

```

ПРОГРАММА UserPrg
VAR
Config: SERIAL_CFG;
Порт: SERIAL_PORT := COM1;
Параметры: 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 («ЛОЖЬ»),

```

## 5. КОНФИГУРАЦИЯ

```

ENABLE_DCD_EVENT := FALSE («ЛОЖЬ»),
ENABLE_CTS_EVENT := FALSE («ЛОЖЬ»);
Статус: SERIAL_STATUS;

END_VAR // ВХОДЫ:
Config.REQUEST := TRUE («ИСТИНА»);
Config.PORT := Порт;
Config.PARAMETERS := Parameters; // ФУНКЦИЯ:
Config(); // ВЫХОДЫ:
Config.DONE;
Config.EXEC;
Config.ERROR;
Статус:= Config.STATUS; //Если необходимо провести обработку ошибки.
    
```

### 5.10.1.2. Функциональный блок «SERIAL\_GET\_CFG»

Функциональный блок используется для захвата требуемой конфигурации последовательного порта.



Рисунок 107: Блок захвата последовательной конфигурации

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 123: Входные параметры SERIAL\_GET\_CFG

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.
Выходные параметры	Тип	Описание

STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
PARAMETERS	SERIAL_PARAMETERS	В данную структуру передаются параметры конфигурации последовательного порта, описанные в типе данных SERIAL_PARAMETERS.

Таблица 124: Выходные параметры SERIAL\_GET\_CFG

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект:

```

ПРОГРАММА UserPrg
VAR
GetConfig: SERIAL_GET_CFG;
Порт: SERIAL_PORT := COM1;
Параметры: SERIAL_PARAMETERS;
Статус: SERIAL_STATUS;
END_VAR // ВХОДЫ:
GetConfig.REQUEST := TRUE («ИСТИНА»);
GetConfig.PORT := Порт; //ФУНКЦИЯ:
GetConfig(); // ВЫХОДЫ:
GetConfig.DONE;
GetConfig.EXEC;
GetConfig.ERROR;
Статус:= GetConfig.STATUS; //Если необходимо провести обработку ошибки.
Параметры:= GetConfig.PARAMETERS; // Получение параметров нужного последовательного порта.
    
```

### 5.10.1.3. Функциональный блок «SERIAL\_GET\_CTRL»

Данный функциональный блок предназначен для считывания управляющих сигналов CTS (готовность к передаче), DSR (источник данных готов) и DCD (активность и готовность модема к передаче), если таковые присутствуют в последовательном порту. При отсутствии управляющих сигналов будет возвращено ложное значение.



Рисунок 108: Блок, используемый для визуализации управляющих сигналов

## 5. КОНФИГУРАЦИЯ

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 125: Входные параметры SERIAL\_GET\_CTRL

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: <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_GET_CTRL_NOT_ALLOWED</li> <li>- NOT_CONFIGURED</li> </ul>
CTS_VALUE	BOOL	Значение, считываемое в управляющем сигнале CTS.
DSR_VALUE	BOOL	Значение, считываемое в управляющем сигнале DSR.
DCD_VALUE	BOOL	Значение, считываемое в управляющем сигнале DCD.

Таблица 126: Выходные параметры SERIAL\_GET\_CTRL

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект и выполнения конфигурирования последовательного порта:

```
ПРОГРАММА UserPrg
```

## 5. КОНФИГУРАЦИЯ

```

VAR
Get_Control: SERIAL_GET_CTRL;
Порт: SERIAL_PORT := COM1;
Статус: SERIAL_STATUS;
END_VAR // ВХОДЫ:
Get_Control.REQUEST := TRUE («ИСТИНА»);
Get_Control.PORT := Порт; //ФУНКЦИЯ:
Get_Control(); // ВЫХОДЫ:
Get_Control.DONE;
Get_Control.EXEC;
Get_Control.ERROR;
Статус:= Get_Control.STATUS;
Get_Control.CTS_VALUE;
Get_Control.DSR_VALUE;
Get_Control.DCD_VALUE;

```

//Если необходимо провести обработку ошибки.

### 5.10.1.4. Блок «SERIAL\_GET\_RX\_QUEUE\_STATUS»

Данный блок предназначен для считывания некоторой информации о состоянии очереди RX, специально разработан для нормального режима, но также может быть использован и в расширенном режиме.



Рисунок 109: Блок, используемый для визуализации состояния очереди RX

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 127: Входные параметры SERIAL\_GET\_RX\_QUEUE\_STATUS

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).

## 5. КОНФИГУРАЦИЯ

Выходные параметры	Тип	Описание
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: <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>- NOT_CONFIGURED</li> </ul>
RXQ_STATUS	SERIAL_RX_QUEUE_STATUS	Возвращает статус/ошибку очереди RX, как описано в типе данных SERIAL_RX_QUEUE_STATUS.

Таблица 128: Выходные параметры SERIAL\_GET\_RX\_QUEUE\_STATUS

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект и выполнения конфигурирования последовательного порта:

```

ПРОГРАММА UserPrg
VAR
Get_Status: SERIAL_GET_RX_QUEUE_STATUS;
Порт: SERIAL_PORT := COM1;
Статус: SERIAL_STATUS;
Status_RX: SERIAL_RX_QUEUE_STATUS;
END_VAR //INPUTS:
Get_Status.REQUEST := TRUE («ИСТИНА»);
Get_Status.PORT := Порт; //ФУНКЦИЯ:
Get_Status(); // ВЫХОДЫ:
Get_Status.DONE;
Get_Status.EXEC;
Get_Status.ERROR;
Статус := Get_Status.STATUS; //Если необходимо провести обработку ошибки.
Status_RX := Get_Status.RXQ_STATUS; //Если необходимо провести обработку ошибки RX.

```

### 5.10.1.5. Функциональный блок «SERIAL\_PURGE\_RX\_QUEUE»

Данный функциональный блок используется для очистки очереди RX последовательного порта, локального и удаленного. При этом также перезапускается порядок поступления (FIFO) RX универсального асинхронного приёмопередатчика (UART).



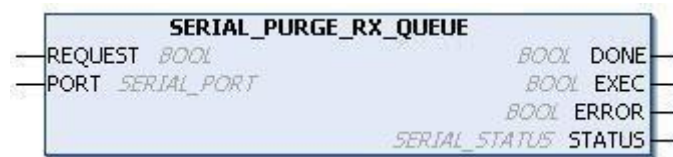


Рисунок 110: Блок, используемый для очистки очереди RX

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 129: Входные параметры SERIAL\_PURGE\_RX\_QUEUE

Входные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), когда блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная является true («истина») во время выполнения блока. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: <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>- NOT_CONFIGURED</li> </ul>

Таблица 130: Выходные параметры SERIAL\_PURGE\_RX\_QUEUE

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект и выполнения конфигурирования последовательного порта:

## 5. КОНФИГУРАЦИЯ

```

ПРОГРАММА UserPrg
VAR
Purge_Queue: SERIAL_PURGE_RX_QUEUE;
Порт: SERIAL_PORT := COM1;
Статус: SERIAL_STATUS;
END_VAR // ВХОДЫ:
Purge_Queue.REQUEST := TRUE («ИСТИНА»);
Purge_Queue.PORT := Порт; //ФУНКЦИЯ:
Purge_Queue(); // ВЫХОДЫ:
Purge_Queue.DONE;
Purge_Queue.EXEC;
Purge_Queue.ERROR;
Статус:= Purge_Queue.STATUS; //Если необходимо провести обработку ошибки.

```

### 5.10.1.6. Функциональный блок «SERIAL\_RX»

Данный функциональный блок предназначен для приема буфера последовательного порта с использованием нормального режима очереди RX. В данном режиме каждый символ в очереди RX занимает один байт, в котором находятся принятые данные, хранящиеся в 5, 6, 7 или 8 битах, в зависимости от конфигурации последовательного интерфейса.



Рисунок 111: Блок, используемый для считывания значений приемного буфера

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
RX_BUFFER_POINTER	POINTER TO BYTE	Указатель байтового массива для получения значений буфера.
RX_BUFFER_LENGTH	UINT	Укажите ожидаемый номер символа в байтовом массиве. В случае если доступно больше ожидаемых байт, из массива байт будет считано только ожидаемое количество, остальное будет оставлено в очереди RX (максимальный размер составляет 1024 символа).
RX_TIMEOUT	UINT	Укажите блокировку по превышению лимита времени для приема ожидаемого количества символов. В случае, если он меньше необходимого для приема символов, будет индцироваться вывод RX_TIME-OUT_ERROR из параметра STATUS («Статус»). Если указанное значение в мс равно нулю, то функция вернет данные, находящиеся в буфере.

Таблица 131: Входные параметры SERIAL\_RX

Выходные параметры	Тип	Описание
--------------------	-----	----------

## 5. КОНФИГУРАЦИЯ

DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: <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>- ILLEGAL_RX_BUFF_LENGTH</li> <li>- RX_TIMEOUT_ERROR</li> <li>- FB_SERIAL_RX_NOT_ALLOWED</li> <li>- NOT_CONFIGURED</li> </ul>
RX_RECEIVED	UINT	Возвращает число принятых символов. Это число может быть в пределах от нуля до заданного значения в RX_BUFFER_LENGTH. Если оно меньше, то функциональный блок выдаст ошибку.
RX_REMAINING	UINT	Возвращает количество символов, оставшихся в очереди RX после выполнения функционального блока.

Таблица 132: Выходные параметры SERIAL\_RX

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект и выполнения конфигурирования последовательного порта:

## 5. КОНФИГУРАЦИЯ

```

ПРОГРАММА UserPrg
VAR
Получение: SERIAL_RX;
Порт: SERIAL_PORT := COM1;
Buffer_Pointer: МАССИВ [0..1023] БАЙТОВ;           //Макс. размер
Статус: SERIAL_STATUS;
END_VAR // ВХОДЫ:
Receive.REQUEST := TRUE («ИСТИНА»);
Receive.PORT := Порт;
Receive.RX_BUFFER_POINTER := ADR(Buffer_Pointer); Receive.RX_BUFFER_LENGTH := 1024; //Макс.
размер
Receive.RX_TIMEOUT := 10000;
// ФУНКЦИЯ:
Receive(); // ВЫХОДЫ:
Receive.DONE;
Receive.EXEC;
Receive.ERROR;
Status := Receive.STATUS;
Receive.RX_RECEIVED;
Receive.RX_REMAINING;           //Если необходимо провести обработку ошибки.

```

### 5.10.1.7. Функциональный блок «SERIAL\_RX\_EXTENDED»

Данный функциональный блок используется для приема буфера последовательного порта с использованием расширенного режима очереди RX, как показано в разделе «[Конфигурация последовательных интерфейсов](#)».



Рисунок 112: Блок, используемый для считывания приемного буфера

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
RX_BUFFER_POINTER	POINTER TO SERIAL_RX_CHAR_EXTENDED	Указатель массива SERIAL_RX_CHAR_EXTENDED для приема значений буфера.
RX_BUFFER_LENGTH	UINT	Укажите номер ожидаемого символа в массиве SERIAL_RX_CHAR_EXTENDED. В случае если доступно больше ожидаемых байт, из массива байт будет считано только ожидаемое количество, остальное будет оставлено в очереди RX (максимальный размер составляет 1024 символа).

5. КОНФИГУРАЦИЯ

RX_TIMEOUT	UINT	Укажите блокировку по превышению лимита времени для приема ожидаемого количества символов. В случае, если он меньше необходимого для приема символов, будет индцироваться вывод RX_TIMEOUT_ERROR из параметра STATUS («Статус»). Если указанное значение в мс равно нулю, то функция вернет данные, находящиеся в буфере.
------------	------	---

Таблица 133: Входные параметры SERIAL\_RX\_EXTENDED

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: <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>- ILLEGAL_RX_BUFF_LENGTH</li> <li>- RX_TIMEOUT_ERROR</li> <li>- FB_SERIAL_RX_EXTENDED_NOT_ALLOWED</li> <li>- NOT_CONFIGURED</li> </ul>
RX_RECEIVED	UINT	Возвращает число принятых символов. Данное число может быть в пределах от нуля до заданного значения в RX_BUFFER_LENGTH. Если оно меньше, то функциональный блок выдаст ошибку.
RX_REMAINING	UINT	Возвращает количество символов, оставшихся в очереди RX после выполнения функционального блока.

## 5. КОНФИГУРАЦИЯ

RX_SILENCE	UINT	Возвращает время паузы в очереди RX, измеренное с момента завершения последнего принятого символа. Единицей измерения времени является 10 мкс. Данный тип выходного параметра важен для определения времени молчания в таких протоколах, как MODBUS RTU. Это может быть не время паузы после последнего принятого символа данным функциональным блоком, поскольку оно имеет значение true («истина») только в том случае, если RX_REMAINING = 0.
------------	------	--

Таблица 134: Выходные параметры SERIAL\_RX\_EXTENDED

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект и выполнения конфигурирования последовательного порта:

```
ПРОГРАММА UserPrg
VAR
Receive_Ex: SERIAL_RX_EXTENDED;
Порт: SERIAL_PORT := COM1;
Buffer_Pointer: МАССИВ [0..1023] SERIAL_RX_CHAR_EXTENDED;
Статус: SERIAL_STATUS;
END_VAR // ВХОДЫ:
Receive_Ex.REQUEST := TRUE («ИСТИНА»);
Receive_Ex.PORT := Порт;
Receive_Ex.RX_BUFFER_POINTER := ADR(Buffer_Pointer);
Receive_Ex.RX_BUFFER_LENGTH := 1024; //Макс. размер
Receive_Ex.RX_TIMEOUT := 10000; // ФУНКЦИЯ:
Receive_Ex(); // ВЫХОДЫ:
Receive_Ex.DONE;
Receive_Ex.EXEC;
Receive_Ex.ERROR;
Статус:= Receive_Ex.STATUS; //Если необходимо провести обработку ошибки.
Receive_Ex.RX_RECEIVED;
Receive_Ex.RX_REMAINING;
Receive_Ex.RX_SILENCE;
```

### 5.10.1.8. Блок «SERIAL\_SET\_CTRL»

Данный блок используется для записи на управляющие сигналы (RTS (запрос на передачу) и DTR (приемник данных готов)), когда они доступны в последовательном порту. Он также может установить состояние занятости передачи через параметр BREAK и может использоваться только в том случае, если сигнал модема сконфигурирован для RS232\_MANUAL.



Рисунок 113: Блок для записи управляющих сигналов

Входные параметры	Тип	Описание
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
RTS_VALUE	BOOL	Значение, записываемое по сигналу RTS.
RTS_EN	BOOL	Разрешает запись параметра RTS_VALUE.
DTR_VALUE	BOOL	Значение, записываемое по сигналу DTR.
DTR_EN	BOOL	Разрешает запись параметра DTR_VALUE.
BREAK	BOOL	В случае, если значение true («истина»), в линию передачи включается логический 0 (занято).

Таблица 135: Входные параметры SERIAL\_SET\_CTRL

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.

## 5. КОНФИГУРАЦИЯ

STATUS	SERIAL_STATUS	<p>Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом:</p> <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 136: Выходные параметры SERIAL\_SET\_CTRL

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект и выполнения конфигурирования последовательного порта:

```

ПРОГРАММА UserPrg
VAR
Set_Control: SERIAL_SET_CTRL;
Порт: SERIAL_PORT := COM1;
Статус: SERIAL_STATUS;
END_VAR
// ВХОДЫ:
Set_Control.REQUEST := TRUE («ИСТИНА»);
Set_Control.PORT := Порт;
Set_Control.RTS_VALUE := FALSE («ЛОЖЬ»);
Set_Control.RTS_EN := FALSE («ЛОЖЬ»);
Set_Control.DTR_VALUE := FALSE («ЛОЖЬ»);
Set_Control.DTR_EN := FALSE («ЛОЖЬ»);
Set_Control.BREAK := FALSE («ЛОЖЬ»); // ФУНКЦИЯ:
Set_Control(); // ВЫХОДЫ:
Set_Control.DONE;
Set_Control.EXEC;
Set_Control.ERROR;
Статус:= Set_Control.STATUS; //Если необходимо провести обработку ошибки.

```

### 5.10.1.9. Функциональный блок «SERIAL\_TX»

Данный функциональный блок предназначен для передачи буфера данных через последовательный порт и завершается только после того, как все байты были переданы или после блокировки по превышению лимита времени (генерации ошибок).

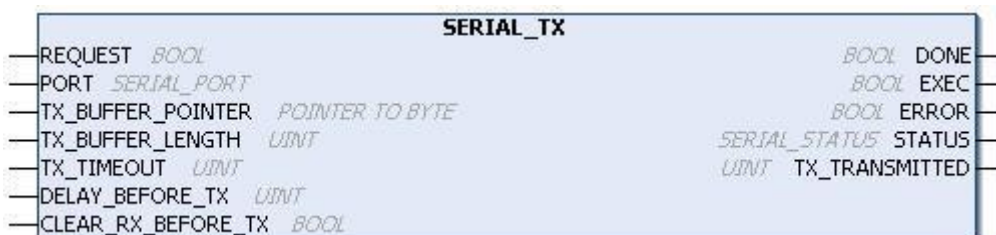




Рисунок 114: Блок для передачи значений по последовательному интерфейсу

Входные параметры	Тип	Описание	
REQUEST	BOOL	Данная переменная, если она имеет значение true («истина»), позволяет использовать функциональный блок.	
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.	
TX_BUFFER_POINTER	POINTER TO BYTE	Указатель байтового массива для передачи значений буфера.	
TX_BUFFER_LENGTH	UINT	Укажите ожидаемый номер символа в передаваемом массиве байт (максимальный размер очереди TX - 1024 символа).	
TX_TIMEOUT	UINT	Укажите время блокировки по превышению лимита времени для завершения передачи, включая фазу синхронизации интерфейса и периферийного оборудования. Указанное значение в мс должно быть положительным и отличным от нуля.	
DELAY_BEFORE_TX	UINT	Укажите задержку в мс между вызовом функционального блока и началом передачи. Данная переменная может быть использована при обмене данными с некоторыми модемами.	
Входные параметры	Тип	Описание	Входные параметры
CLEAR_RX_BEFORE_TX	BOOL	Если значение равно true («истина»), то очередь RX и RX универсального асинхронного приемопередатчика (UART) в порядке поступления (FIFO) стираются перед началом передачи. Такая логика работы характерна для полудуплексных протоколов ведущего/ведомого устройства.	

Таблица 137: Входные параметры SERIAL\_TX

Выходные параметры	Тип	Описание
DONE	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
EXEC	BOOL	Данная переменная равна true («истина»), если блок полностью выполнен. В противном случае она имеет значение false («ложь»).
ERROR	BOOL	Данная переменная равна true («истина»), если блок завершает выполнение с ошибкой. В противном случае она имеет значение false («ложь»). Она связана с переменной DONE («Выполнено»), поскольку ее состояние отображается после завершения блока.

## 5. КОНФИГУРАЦИЯ

STATUS	SERIAL_STATUS	Если переменная ERROR («Ошибка») равна true («истина»), то в структуре STATUS («Состояние») будет показана ошибка, обнаруженная при выполнении блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, выглядят следующим образом: - 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
TX_TRANSMITTED	UINT	Возвращает номер переданного байта, который должен быть равен TX_BUFFER_LENGTH, но может быть и меньше, если при передаче произошла какая-то ошибка.

Таблица 138: Выходные параметры SERIAL\_TX

Пример использования на языке структурированного текста (ST), после вставки библиотеки в проект и выполнения конфигурирования последовательного порта:

```

ПРОГРАММА UserPrg
VAR
  Передать: SERIAL_TX;
  Порт: SERIAL_PORT := COM1;
  Buffer_Pointer: МАССИВ [0..9] БАЙТОВ:= [0,1,2,3,4,5,6,7,8,9];
  Статус: SERIAL_STATUS;
END_VAR

// ВХОДЫ:
Transmit.REQUEST := TRUE («ИСТИНА»);
Transmit.PORT := Порт;
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 («ИСТИНА»); //ФУНКЦИЯ:
Transmit(); // ВЫХОДЫ:
Transmit.DONE;
Transmit.EXEC;
Transmit.ERROR;
Статус:= Transmit.STATUS; //Если необходимо провести обработку ошибки.
Transmit.TX_TRANSMITTED;

```

### 5.10.2. Обновление входов и выходов

По умолчанию локальная шина и интегрированный в центральный процессор вход/выход обновляются на каждом цикле выполнения MainTask. Функции обновления (Refresh) позволяют обновлять данные точки ввода/вывода асинхронно в любой точке кода пользовательского приложения.

Если функциональные блоки для обновления входов и выходов не используются, то обновление выполняется каждый цикл выполнения MainTask.

**ВНИМАНИЕ:**  
При запуске центрального процессора (ЦП) данной серии входы и выходы обновляются для считывания и подготавливаются к записи только при выполнении MainTask. Все остальные системные задачи, выполняемые до MainTask, будут с недействительными входами и выходами.

#### 5.10.2.1. Функциональный блок «REFRESH\_INPUT»

Данный функциональный блок используется для обновления указанных входов модуля без необходимости ожидания завершения цикла. Важно отметить, что фильтры, настроенные в MasterTool IEC XE, и время обновления входов модуля должны будут учитываться в эффективном времени обновления входов в приложении, разрабатываемом пользователем.

**ВНИМАНИЕ:**  
Функция *REFRESH\_INPUT* должна использоваться только в MainTask.  
Для обновления входов в других задачах необходимо выбрать опцию «*Включить обновление ввода-вывода для каждой задачи*» (*Enable I/O update per task*). Подробную информацию о данной опции см. в таблице 36.

**ВНИМАНИЕ:**  
Функция *REFRESH\_INPUT* не поддерживает входы, которые были преобразованы в символические переменные. Для корректной работы необходимо, чтобы входной сигнал был преобразован в переменную в памяти прямого представления входных переменных (%I).

**ВНИМАНИЕ:**  
Функция *REFRESH\_INPUT* обновляет только прямые переменные %I, объявленные на вкладке «Шина: Преобразование входов/выходов» (Bus: I/O Mapping) модуля, адресованного в соответствующую стойку/слот функции. В случае коммуникационных модулей/интерфейсов (MODBUS, Profibus и т.д.) обновление не распространяется на прямые переменные преобразования устройств.

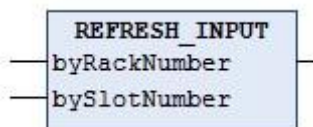


Рисунок 115: Блок для обновления входа

Входные параметры	Тип	Описание
byRackNumber	BYTE	Номер стойки.
bySlotNumber	BYTE	Номер позиции, к которой подключен модуль.

Таблица 139: Входные параметры REFRESH\_INPUT

## 5. КОНФИГУРАЦИЯ

Возможные значения `TYPE_RESULT`:

- `OK_SUCCESS`: Успешное выполнение.
- `ERROR_FAILED`: Данная ошибка возвращается в том случае, если функция вызывается для модуля, имеющего только выходы, или если не отмечена опция «Всегда обновлять переменные» (*Always update variables*) (находится на экране конфигурации модуля, закладка «Преобразование входов / выходов» (*I/O Mapping*)).
- `ERROR_NOTSUPPORTED`: Вызываемая стандартная подпрограмма не поддерживается изделием.
- `ERROR_PARAMETER`: Недопустимый / неподдерживаемый параметр.
- `ERROR_MODULE_ABSENT`: Модуль отсутствует в шине.
- `ERROR_MODULE_NOTCONFIGURED`: Модуль не был сконфигурирован в приложении.
- `ERROR_MODULE_NOTRUNNING`: Модуль не запущен (не находится в рабочем состоянии).
- `ERROR_MODULE_COMMFAIL`: Неисправность связи с модулем.
- `ERROR_MODULE_NOTFOUND`: Модуль не найден в приложении или не поддерживается.

Пример использования на языке структурированного текста (ST):

```
ПРОГРАММА UserPrg
VAR
Информация: TYPE_RESULT;
byRackNumber: BYTE; bySlotNumber: BYTE;
END_VAR // ВХОДЫ:
byRackNumber := 0; bySlotNumber := 10;
// ФУНКЦИЯ:
Информация:= REFRESH_INPUT (byRackNumber, bySlotNumber); // Вызов ФУНКЦИИ.
//Переменная «Информация» («Info») получает возможные ошибки функционирования.
```

### 5.10.2.2. Функциональный блок «REFRESH\_OUTPUT»

Данный функциональный блок используется для обновления указанных выходов модуля. При этом нет необходимости дожидаться завершения цикла. Важно отметить, что время обновления выходов модуля необходимо будет учитывать в эффективно используемом времени обновления выходов в разрабатываемом пользователем приложении.

#### ВНИМАНИЕ:

Функция `REFRESH_OUTPUT` должна использоваться только в `MainTask`.

Для обновления выходов в других задачах необходимо выбрать опцию «Включить обновление ввода-вывода для каждой задачи» (*Enable I/O update per task*). Подробную информацию о данной опции см. В таблице 36.

#### ВНИМАНИЕ:

Функция `REFRESH_OUTPUT` не поддерживает входы, которые были преобразованы в символические переменные. Для корректной работы необходимо, чтобы входной сигнал был преобразован в переменную в рамках прямого представления входных переменных в памяти (%Q).

#### ВНИМАНИЕ:

Функция `REFRESH_OUTPUT` обновляет только прямые переменные %Q, объявленные на вкладке «Шина: Преобразование входов/выходов» (*Bus: I/O Mapping*) модуля, адресованного в соответствующую стойку/слот функции. В случае коммуникационных модулей/интерфейсов (MODBUS, Profibus и т.д.) обновление не включает в себя прямые переменные преобразования устройств.

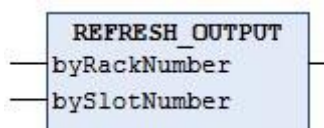


Рисунок 116: Блок для обновления выхода

Входные параметры	Тип	Описание
byRackNumber	BYTE	Номер стойки.
bySlotNumber	BYTE	Номер позиции, к которой подключен модуль.

Таблица 140: Входные параметры REFRESH\_OUTPUT

Потенциально возможное значение *TYPE\_RESULT*:

- OK\_SUCCESS: Успешное выполнение.
- ERROR\_FAILED: Данная ошибка возвращается в том случае, если функция вызывается для модуля, имеющего только входы, или если не отмечена опция «Всегда обновлять переменные» (*Always update variables*) (находится на экране конфигурации модуля, закладка «Преобразование входов / выходов» (*I/O Mapping*)).
- ERROR\_NOTSUPPORTED: Вызываемая стандартная подпрограмма не поддерживается изделием.
- ERROR\_PARAMETER: Недопустимый / неподдерживаемый параметр.
- ERROR\_MODULE\_ABSENT: Модуль отсутствует в шине.
- ERROR\_MODULE\_NOTCONFIGURED: Модуль не был сконфигурирован в приложении.
- ERROR\_MODULE\_NOTRUNNING: Модуль не запущен (не находится в рабочем состоянии).
- ERROR\_MODULE\_COMMFAIL: Неисправность связи с модулем.
- ERROR\_MODULE\_NOTFOUND: Модуль не найден в приложении или не поддерживается.

Пример использования на языке структурированного текста (ST):

```

ПРОГРАММА UserPrg VAR
Информация: TYPE_RESULT; byRackNumber:
BYTE; bySlotNumber: BYTE;
END_VAR // ВХОДЫ:
byRackNumber := 0; bySlotNumber := 10;
// ФУНКЦИЯ:
// Вызов ФУНКЦИИ.
Информация:= REFRESH_OUTPUT (byRackNumber, bySlotNumber); //Переменная
«Информация» («Info») получает возможные ошибки функционирования.

```

### 5.10.3. Функциональный блок PID (Пропорционально-Интегрально-Дифференциальный Регулятор)

#### ВНИМАНИЕ:

Функциональный блок PID, описанный до предыдущей редакции L данного руководства, устарел и был удален из данного руководства.

Функциональные блоки PID, PID\_INT и PID\_REAL, описанные до ревизии C документа MP399609, также устарели и были удалены из новых версий этого руководства. Пользователи, которым по причинам технического обслуживания необходимо описание данных устаревших функциональных блоков, должны использовать редакцию C MP399609. Функциональные блоки PID, PID\_INT и PID\_REAL не должны использоваться в новых проектах. Данные функциональные блоки были заменены новыми функциональными блоками с дополнительными ресурсами, такими как автоматическая настройка и поддержка каскадного, обходного и опережающего управления. Эти новые

## 5. КОНФИГУРАЦИЯ

функциональные блоки описаны в MU214610 и доступны после версии 1.1.0.0 библиотеки *NextoPID*.

### 5.10.4. Сохранение таймера

Сохранение таймера (*timer retain*) представляет собой функциональный блок, разработанный для применения в качестве часов для производственных линий, которым необходимо сохранять значение и возобновлять отсчет с той же точки в случае отказа системы электропитания. Значения, сохраненные функциональным блоком, обнуляются только в случае *Холодного сброса (Reset Cold)*, *Сброса до исходного состояния (Reset Origin)* или Загрузки (*Download*) нового приложения (см. Руководство пользователя MasterTool IEC XE – MU299609), когда счетчики продолжают работать, даже если приложение остановлено (режим «Останов» (Stop)).

#### ВНИМАНИЕ:

Важно подчеркнуть, что для корректной работы блоков сохранения таймера (*Timer Retain*) переменные должны быть объявлены как «Сохраняемые» (*Retain*) (*VAR RETAIN*). Важно также отметить, что в режиме моделирования функциональные блоки сохранения таймера (*Timer Retain*) работают некорректно, поскольку для корректной логики работы требуется центральный процессор (ЦП) *Nexto*.

Ниже описаны три блока, уже имеющиеся в библиотеке *NextoStandard* программного обеспечения MasterTool IEC XE (порядок вставки библиотеки см. в Руководстве по программированию MasterTool IEC XE – MP399609, раздел «Библиотека»).

#### 5.10.4.1. Функциональный блок «TOF\_RET»

Функциональный блок *TOF\_RET* реализует временную задержку на отключение выхода. При изменении состояния входа *IN* с (TRUE) на (FALSE) или при спадающем фронте импульса будет отсчитано заданное время *PT*, по окончании которого выход *Q* будет переведен в состояние (FALSE («ЛОЖЬ»)). Когда на вход *IN* подается логический уровень 1 (TRUE («ИСТИНА»)), выход *Q* остается в том же состоянии (TRUE («ИСТИНА»)), даже если это произошло в середине процесса подсчета. Время *PT* может быть изменено в процессе подсчета, поскольку блок принимает новое значение, если подсчет еще не был завершен. На рисунке 117 изображен блок *TOF\_RET*, а на рисунке 118 показана его логика работы в графическом представлении.



Рисунок 117: Блок *TOF\_RET*

Входные параметры	Тип	Описание
IN	BOOL	Данная переменная, получив спадающий фронт импульса, разрешает счет блока.
PT	TIME	Эта переменная задает предел счета блоков (временную задержку).

Таблица 141: Входные параметры *TOF\_RET*

Входные параметры	Тип	Описание
Q	BOOL	Данная переменная выполняет спадающий фронт импульса, когда переменная <i>PT</i> (временная задержка) достигает своего максимального значения.

## 5. КОНФИГУРАЦИЯ

ET	TIME	Данная переменная показывает текущую задержку времени.
----	------	--

Таблица 142: Выходные параметры TOF\_RET

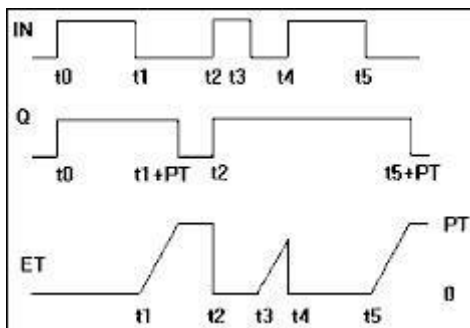


Рисунок 118: Графическое представление логики работы блока TOF\_RET

Пример использования на языке структурированного текста (ST):

```

ПРОГРАММА UserPrg VAR RETAIN
bStart : BOOL := TRUE («ИСТИНА»);
TOF_RET : TOF_RET; END_VAR

// При bStart=FALSE («ЛОЖЬ») начинается отсчет
TOF_RET( IN := bStart,
PT := T#20S);

// Действия, выполняемые по окончании отсчета
ЕСЛИ (TOF_RET.Q = FALSE («ЛОЖЬ»)) TO bStart := TRUE («ИСТИНА»);
END_IF
    
```

### 5.10.4.2. Функциональный блок «TON\_RET»

Функциональный блок «TON\_RET» реализует временную задержку на включение выхода. При изменении состояния входа *IN* с (FALSE («ЛОЖЬ»)) на (TRUE («ИСТИНА»)) либо при нарастающем фронте импульса будет отсчитываться заданное время *PT*, по окончании которого выход *Q* будет переведен в состояние (TRUE («ИСТИНА»)). Когда на входе *IN* находится логический уровень 0 (FALSE («ЛОЖЬ»)), выход *Q* остается в том же состоянии (FALSE («ЛОЖЬ»)), даже если это происходит в середине процесса подсчета. Время *PT* может быть изменено в процессе подсчета, поскольку блок принимает новое значение, если подсчет еще не был закончен. На рисунке 119 изображен блок «TON\_RET», а на рисунке 120 показана его логика работы в графическом представлении.



Рисунок 119: Функциональный блок TON\_RET

Входные параметры	Тип	Описание
IN	BOOL	Данная переменная, получив нарастающий фронт импульса, включает отсчеты функционального блока.

## 5. КОНФИГУРАЦИЯ

PT	TIME	Эта переменная задает предел счета блоков (временную задержку).
----	------	---

Таблица 143: Входные параметры TON\_RET

Выходные параметры	Тип	Описание
Q	BOOL	Данная переменная выполняет нарастающий фронт импульса, когда переменная PT (временная задержка) достигает своего максимального значения.
ET	TIME	Данная переменная показывает текущую задержку времени.

Таблица 144: Выходные параметры TON\_RET

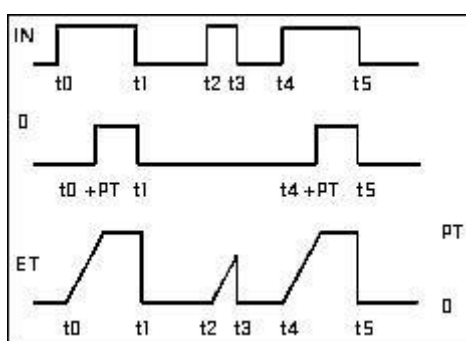


Рисунок 120: Графическое представление логики работы блока TON\_RET

Пример использования на языке структурированного текста (ST):

```

ПРОГРАММА UserPrg VAR
RETAIN bStart : BOOL;
TON_RET : TON_RET;
END_VAR
// При bStart=TRUE («ИСТИНА») начинается отсчет
TON_RET( IN := bStart,
PT := T#20S);
// Действия, выполняемые по окончании отсчета
ЕСЛИ (TON_RET.Q = TRUE («ИСТИНА»)) ТОГДА bStart := FALSE («ЛОЖЬ»); END_IF

```

### 5.10.4.3. Функциональный блок «TP\_RET»

Функциональный блок «TP\_RET» работает как триггер. Таймер, который запускается при изменении состояния входа IN с (FALSE («ЛОЖЬ»)) на (TRUE («ИСТИНА»)), то есть по нарастающему фронту импульса, увеличивается до тех пор, пока не будет достигнут предел времени PT. Во время подсчета выход Q имеет состояние (TRUE («ИСТИНА»)), в противном случае – (FALSE («ЛОЖЬ»)). Время PT может быть изменено во время подсчета, поскольку блок принимает новое значение, если подсчет не был завершен. На рисунке 121 изображен функциональный блок «TP\_RET», а на рисунке 122 показана его логика работы в графическом представлении.



Рисунок 121: Функциональный блок TP\_RET



## 5. КОНФИГУРАЦИЯ

Входные параметры	Тип	Описание
IN	BOOL	Данная переменная, получив нарастающий фронт импульса, включает отсчеты функционального блока.
PT	TIME	Данная переменная задает предел счета функционального блока (временную задержку).

Таблица 145: Входные параметры TP\_RET

Выходные параметры	Тип	Описание
Q	BOOL	Во время подсчета данная переменная имеет значение true («истина»), в противном случае – false («ложь»).
ET	TIME	Данная переменная показывает текущую задержку времени.

Таблица 146: Выходные параметры TP\_RET

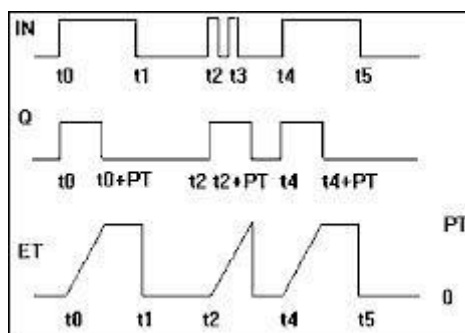


Рисунок 122: Графическое представление логики работы блока TP\_RET

Пример использования на языке структурированного текста (ST):

```

ПРОГРАММА UserPrg
VAR RETAIN bStart :
  BOOL; TP_RET :
  TP_RET; END_VAR

// Конфигурирование TP_RET
TP_RET( IN := bStart, PT :=
T#20S);

bStart := FALSE («ЛОЖЬ»);

// Действия, выполняемые при подсчете
ЕСЛИ (TP_RET.Q = TRUE («ИСТИНА»)) ТО
// Выполняется, пока счетчик активирован
ELSE
// Выполняется, когда счетчик отключен
END_IF

```

## 5.11. Простой протокол сетевого управления (SNMP)

### 5.11.1. Введение

SNMP (*Простой Протокол Сетевого Управления*) представляет собой протокол, широко используемый сетевыми администраторами для предоставления важной информации и диагностики оборудования, присутствующего в данной сети Ethernet.

Данный протокол использует концепцию агента и менеджера, в которой менеджер посылает агенту запросы на считывание или запись определенных объектов. Через MIB (*Информационная База Управления*) менеджер знает о существующих в агенте объектах и, соответственно, может делать запросы к этим объектам, соблюдая разрешения на считывание либо запись. Информационная база управления (MIB) представляет собой набор информации, организованный иерархически, причем каждый объект данного дерева называется OID (*Идентификатор объекта*).

Для всего оборудования с протоколом простого сетевого управления (SNMP) обязательна поддержка MIB-II. В данной информационной базе управления (MIB) описана ключевая информация для управления сетями Ethernet.

### 5.11.2. Простой протокол сетевого управления (SNMP) на центральных процессорах (ЦП) Nexto

Центральные процессоры (ЦП) серии Nexto выступают в роли агентов в передачах данных простого протокола сетевого управления (SNMP). Информация, доступная по простому протоколу сетевого управления (SNMP), не может быть обработана или доступна через пользовательское приложение, поэтому для доступа к ней требуется внешний менеджер протокола SNMP. В таблице ниже приведены объекты, доступные в центральных процессорах Nexto. Данная возможность доступна после версии микропрограммы (прошивки) 1.4.0.33. Центральные процессоры (ЦП) серии Nexto поддерживают протоколы SNMPv1, SNMPv2c и SNMPv3, а также поддерживают MIB-II, где необходимые объекты описаны в RFC-1213.

Идентификатор объекта (OID)	Наименование	Описание
1.3.6.1.2.1.1	System («Система»)	Содержит название, описание, местоположение и другую идентификационную информацию об оборудовании.
1.3.6.1.2.1.2	Interfaces («Интерфейсы»)	Содержит информацию о сетевых интерфейсах машины. В таблице ifTable (Идентификатор объекта (OID) 1.3.6.1.2.1.2.2) доступны индексы 6 и 7, которые могут быть просмотрены статистикой сетевых интерфейсов NET 1 и NET 2, соответственно, центральных процессоров (ЦП) серии Nexto.
Идентификатор объекта (OID)	Наименование	Описание
1.3.6.1.2.1.3	At	Содержит информацию о последних необходимых подключениях к агенту.
1.3.6.1.2.1.4	IP	Содержит статистические соединения по протоколу IP.
1.3.6.1.2.1.5	ICMP	Содержит статистические соединения по протоколу ICMP.
1.3.6.1.2.1.6	TCP	Содержит статистические соединения по протоколу TCP.
1.3.6.1.2.1.7	UDP	Содержит статистические соединения по протоколу UDP.
1.3.6.1.2.1.11	SNMP	Содержит статистические соединения по протоколу SNMP.

1.3.6.1.2.1.31	ifMib	Расширение к интерфейсам, идентификатор объекта (OID) 1.3.6.1.2.1.2.
----------------	-------	--

Таблица 147: Объекты MIB II – Агент SNMP серии Nexto

По умолчанию агент протокола SNMP активирован, т.е. служба инициализируется при запуске центрального процессора (ЦП). Доступ к информации агента осуществляется через интерфейсы сети Ethernet NET 1 и NET 2 центральных процессоров (ЦП) серии Nexto по порту TCP 161. Таким образом, когда сервис активен, доступ к агентской информации может осуществляться через любой из двух интерфейсов сети Ethernet, если таковые доступны. Предоставление агентской информации через интерфейсы сети Ethernet модулей NX5000 не представляется возможным. На рисунке, приведенном ниже, показан пример менеджера протокола SNMP, в котором считываются некоторые значения.

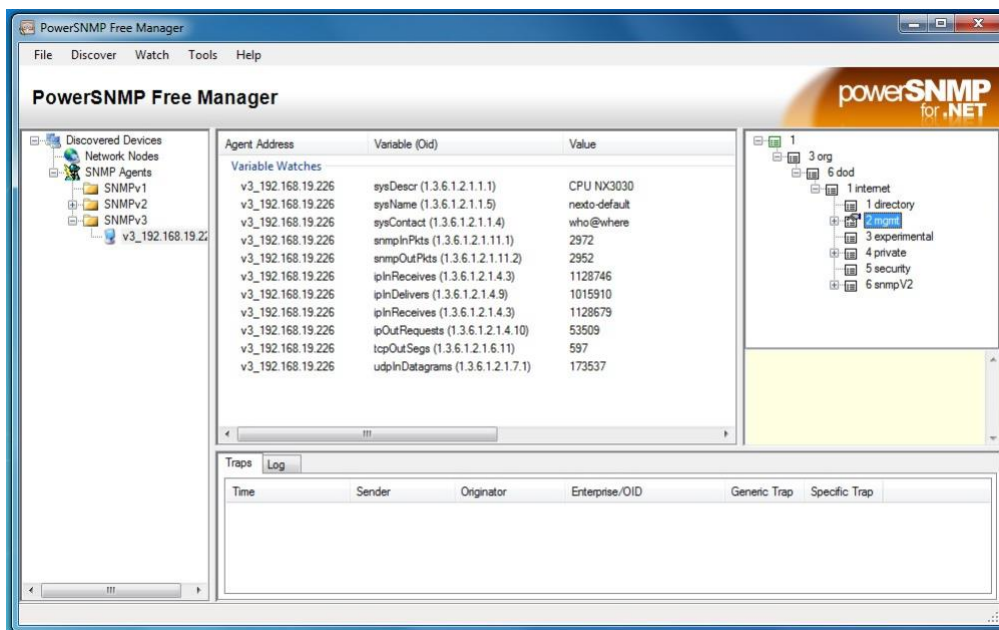


Рисунок 123: Пример менеджера SNMP

В случае SNMPv3, в котором присутствует аутентификация пользователя и пароль для запросов по простому протоколу сетевого управления (SNMP), предусмотрен стандартный пользователь, описанный в разделе «[Пользователь и сообщества простого протокола сетевого управления \(SNMP\)](#)».

Для отключения сервиса, изменения пользователя SNMPv3 или предопределенных сообществ для SNMPv1 / v2c необходимо зайти на веб-страницу центрального процессора (ЦП). Подробности см. в разделе «[Конфигурация простого протокола сетевого управления \(SNMP\)](#)».

### 5.11.3. Частная информационная база управления (MIB)

Частная информационная база управления (MIB) прекратила свое существование с июня 2019 года.

### 5.11.4. Конфигурация простого протокола сетевого управления (SNMP)

Настройки простого протокола сетевого управления (SNMP) можно изменить через веб-страницу, на вкладке «*Управление центральным процессором (ЦП)*» (*CPU Management*) в меню «*Простой протокол сетевого управления (SNMP)*». Для получения доступа к настройкам необходимо сначала выполнить вход в систему, как показано на приведенном ниже рисунке.

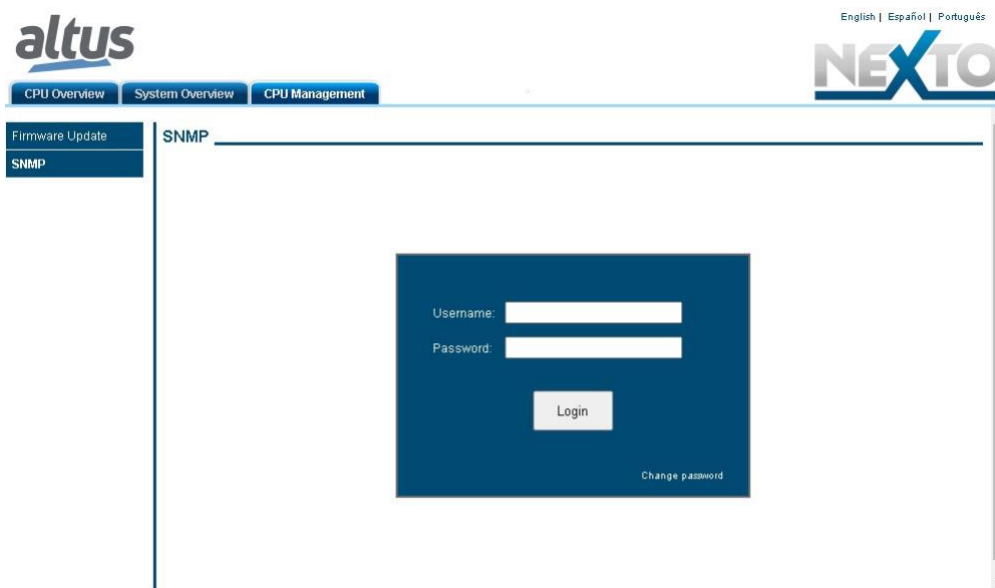


Рисунок 124: Экран входа в систему SNMP

После успешного выполнения входа в систему можно просмотреть текущее состояние сервиса (включен или выключен), а также информацию о пользователе SNMPv3 и сообществах для SNMPv1 / v2c.

Пользователь может включить или выключить сервис с помощью поля для установки флажка в верхней части экрана.

Также можно изменить информацию SNMPv3, нажав кнопку «Изменить» (*Change*), расположенную чуть ниже информации о пользователе. Откроется форма, в которой необходимо ввести старое имя пользователя и пароль, а также новое имя пользователя и пароль. Другая информация о пользователе SNMPv3 не может быть изменена.

Для изменения данных сообществ SNMPv1/v2c процесс аналогичен, достаточно нажать кнопку «Изменить» (*Change*) под информацией сообщества. Откроется новый экран, в который будут вставлены новые данные в поля *rocommunity* и *rwcommunity*. Если не заполнить ни одно из полей, то сообщество будет отключено. Таким образом, если пользователь оставит данные два поля пустыми, то доступ к агенту SNMP будет возможен только по протоколу SNMPv3.

Если же пользователь захочет вернуться к настройкам по умолчанию, то ему необходимо вручную перенастроить их конфигурацию в соответствии с разделом «[Пользователь и сообщества простого протокола сетевого управления \(SNMP\)](#)». Таким образом, все текущие конфигурации простого протокола сетевого управления (SNMP) будут сохранены в процессе обновления микропрограммы (прошивки). Данные опции показаны на рисунке, приведенном ниже.

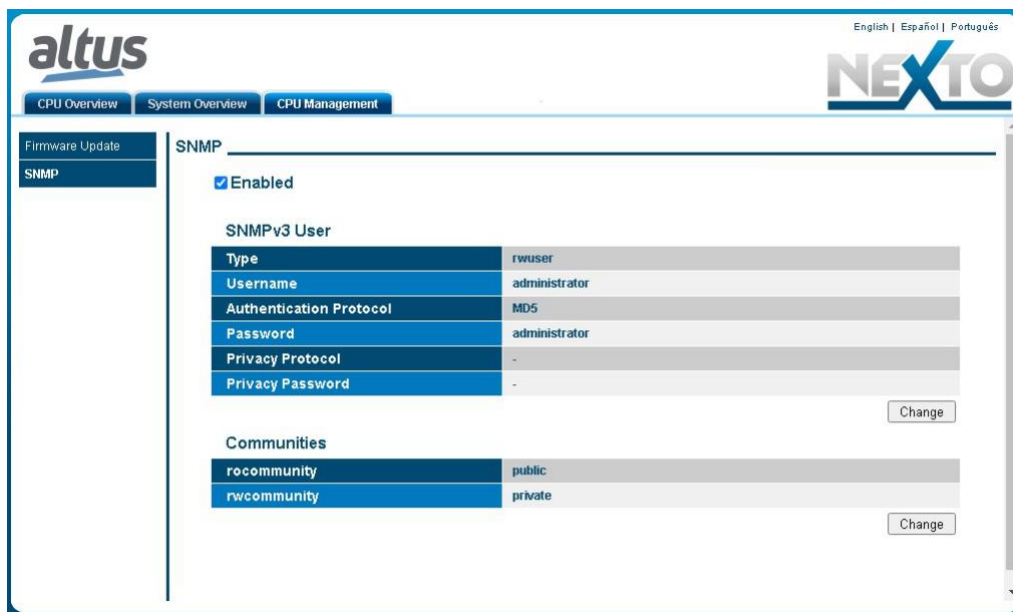


Рисунок 125: Экран конфигурации состояния SNMP

**ВНИМАНИЕ:**  
 Пользователь и пароль для входа в систему на веб-странице настроек SNMP и для получения доступа к агенту по протоколу SNMP одинаковы.

### 5.11.5. Пользователь и сообщества постого протокола сетевого управления (SNMP)

Для получения доступа к SNMPv1 / v2c центральных процессоров (ЦП) серии Nexto существует два сообщества, в соответствии с приведенной ниже таблицей.

Сообщества	Строка по умолчанию	Тип
rocommunity	Public («Общедоступный»)	Только чтение
rwcommunity	Private («Частный»)	Чтение и запись

Таблица 148: Информация о сообществах SNMPv1/v2c по умолчанию

Доступ к SNMPv3 можно получить, используя пользователя по умолчанию, см. таблицу ниже:

Имя пользователя	Тип	Протокол аутентификации	Пароль	Протокол конфиденциальности	Пароль конфиденциальности
administrator	rwuser	MD5	Administrator («Администратор»)	-	-

Таблица 149: Информация о пользователе SNMPv3 по умолчанию

Для всех настроек сообществ, пользователя и пароля должны соблюдаться некоторые ограничения, описанные в следующей таблице:

Конфигурируемый элемент	Минимальный размер	Максимальный размер	Разрешенные символы
rocommunity	-	30	[0-9][a-z][A-Z]@\$*_.
rwcommunity	-	30	[0-9][a-z][A-Z]@\$*_.
V3 User	-	30	[0-9][a-z][A-Z]@\$*_.
V3 Password	8	30	[0-9][a-z][A-Z]@\$*_.

Таблица 150: Пределы настроек SNMP

## 5.12. Управление пользователями и права доступа

Она предоставляет функции для определения учетных записей пользователей и настройки прав доступа к проекту и центральному процессору (ЦП). Используя программное обеспечение MasterTool IEC XE, можно создавать и управлять пользователями и группами, устанавливая различные уровни прав доступа к проекту.

Одновременно в центральных процессорах (ЦП) Nexto присутствует система управления правами пользователей, которая блокирует или разрешает определенные действия для каждой группы пользователей в центральном процессоре. Более подробную информацию можно получить в руководстве пользователя MasterTool IEC XE MT85000 – MU299609, в разделе «Управление пользователями и права доступа».

## 6. Техническое обслуживание

Одной из особенностей серии Nexto является генерация диагностики отклонений, будь то сбои, ошибки или режимы работы, что позволяет оператору с легкостью выявлять и решать проблемы, возникающие в системе.

Центральные процессоры (ЦП) Nexto позволяют использовать множество способов визуализации диагностики, генерируемой системой, а именно:

- [Диагностика одним касанием \(One Touch Diag\)](#)
- [Диагностика с помощью светодиодного индикатора](#)
- [Диагностика с помощью веб-страницы](#)
- [Диагностика с помощью переменных](#)
- [Диагностика с помощью функциональных блоков](#)

Первая из них представляет собой инновационную особенность серии Nexto, позволяющую быстро получить доступ к нештатным ситуациям в приложении. Вторая является чисто визуальной, формируемой двумя светодиодами, расположенными на панели (DG и WD), а также светодиодами, расположенными в разъеме RJ45 (исключительно для подключения к сети Ethernet). Следующей особенностью является графическая визуализация на веб-странице стойки и соответствующих сконфигурированных модулей с возможностью индивидуального доступа к состоянию работы и активной диагностики. Диагностические данные также хранятся непосредственно в переменных центрального процессора (ЦП) в прямом представлении (%Q) или приписываются (переменная директивы AT) и могут быть использованы пользовательским приложением, например, представлены в системе диспетчерского управления. Последние представляют собой специфические условия функционирования системы.

Данная функция диагностики заключается в том, чтобы указать на возможные проблемы в части установки или конфигурации системы, а также на проблемы либо недостатки коммуникационной сети.

### 6.1. Диагностика модуля

#### 6.1.1. Диагностика одним касанием (One Touch Diag)

One Touch Diag (OTD), или диагностика одним касанием, представляет собой эксклюзивную функцию программируемых (логических) контроллеров серии Nexto. Благодаря этой новой концепции пользователь может проверить диагностику любого модуля, подключенного к системе, прямо на графическом дисплее центрального процессора (ЦП) одним нажатием на диагностический переключатель модуля. Это мощный инструмент для диагностики, который можно использовать в автономном режиме (без использования программного обеспечения для диспетчерского управления либо программирования), что облегчает поиск и быстрое решение возможных проблем.

Клавиша диагностики расположена на верхней части центрального процессора (ЦП) в легкодоступном месте и, помимо активной диагностики, позволяет получить доступ к меню навигации, описанному в разделе [«Конфигурация - Информационное и конфигурационное меню центрального процессора \(ЦП\)»](#).

На приведенном ниже рисунке показано расположение переключателя центрального процессора (ЦП):

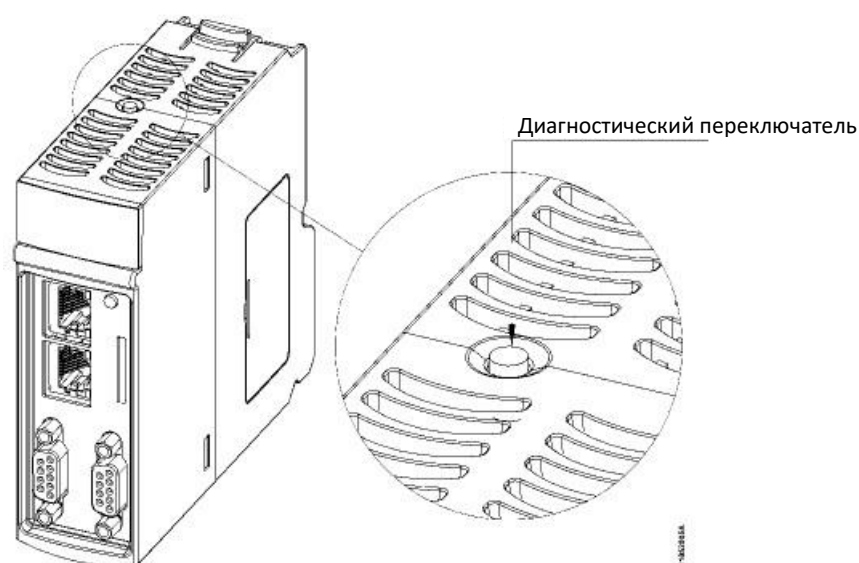


Рисунок 126: Диагностический переключатель

При выполнении одного короткого нажатия центральный процессор (ЦП) начинает показывать диагностику шины (если она активна, в противном случае выводится сообщение «NO DIAG» («ДИАГНОСТИКИ НЕТ»)). Сначала визуализируется тег (метка) (настраивается в свойствах модуля в программном обеспечении MasterTool IEC XE, в соответствии со стандартом МЭК 61131-3 (IEC 61131-3)), т.е. имя, присвоенное центральному процессору (ЦП), а затем отображаются все диагностические данные через сообщения на дисплее центрального процессора. Данный процесс выполняется дважды на дисплее. Все происходит автоматически, поскольку пользователю достаточно выполнить первое короткое нажатие и центральный процессор сам покажет диагностику. Диагностика других модулей, присутствующих на шине, также отображается на графическом дисплее центрального процессора посредством короткого нажатия на кнопку диагностического модуля, в той же модели представления диагностики.

На приведенном ниже рисунке показан процесс, начинающийся с короткого касания, при этом условия и время центрального процессора (ЦП) представлены меньшими прямоугольниками. Важно подчеркнуть, что диагностика может иметь более одного экрана, то есть указанное время на блок-схеме ниже справедливо для одного из них.



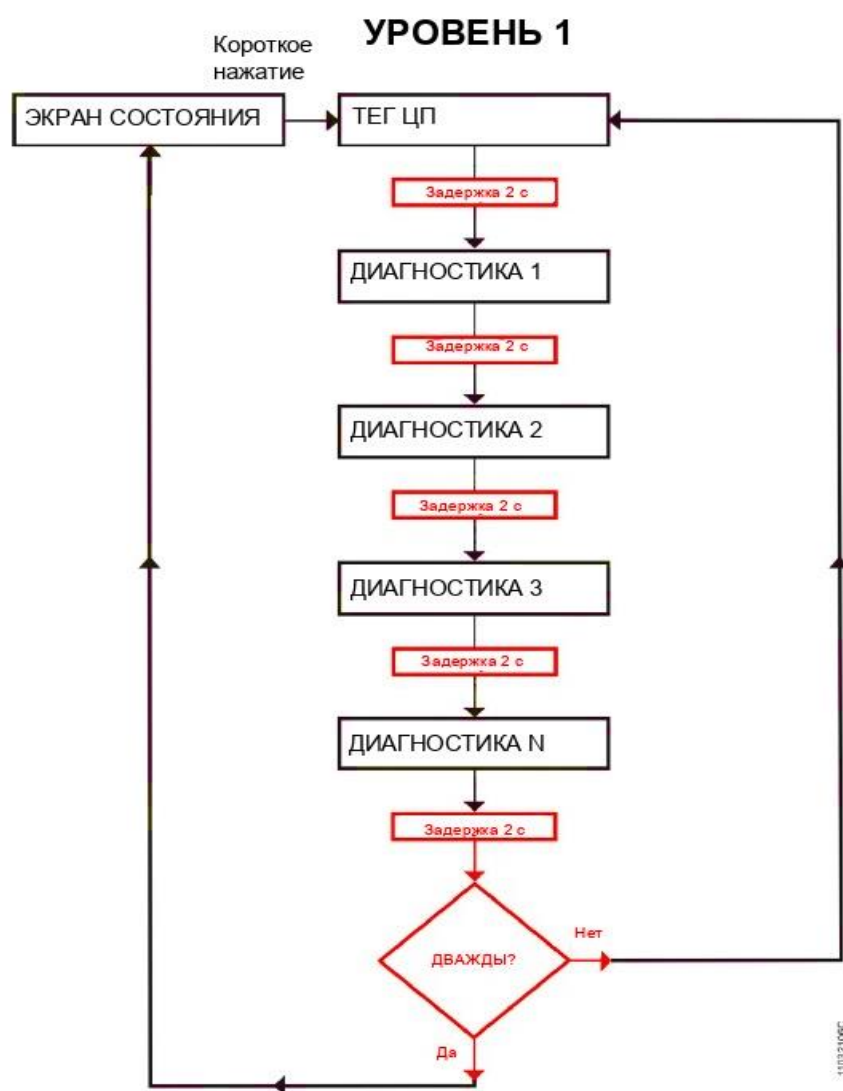


Рисунок 127: Визуализация диагностики центрального процессора (ЦП)

Перед завершением процесса визуализации достаточно в любой момент выполнить короткое касание диагностического переключателя или нажать на него с любого модуля ввода/вывода, подключенного к шине. Также важно отметить, что функция диагностики одним касанием (One Touch Diag) может быть доступна, когда модуль находится в рабочем режиме.

В случае выполнения длинного касания центральный процессор (ЦП) переходит в меню навигации, которое описано в разделе «[Конфигурация - Информационное и конфигурационное меню центрального процессора \(ЦП\)](#)».

В таблице, приведенной ниже, показана разница между временем короткого и длинного касания, а также зависанием кнопки.

Касание	Минимальное время	Максимальное время	Состояние индикации
Без касания	-	59.99 мс	-
Короткое касание	60 мс	0.99 с	Разблокировано
Длительное касание	1 с	20 с	Более 1 с и до 20 с
Заблокированный переключатель	20.01 с	(∞)	Индикация диагностики, см. Таблицу <a href="#">156</a>

Таблица 151: Время одного касания

Сообщения на графическом дисплее центрального процессора (ЦП) Nexto, соответствующие диагностике, описаны в разделе «[Диагностика с помощью переменных](#)», в таблице 156.

Если в одном из модулей ввода/вывода возникнет ситуация зависания кнопки, то при нажатии кнопка диагностики данного модуля перестанет отображать диагностику на графическом дисплее центрального процессора (ЦП). В этом случае центральный процессор укажет на то, что имеется модуль с активной диагностикой. Для удаления данной диагностики из центрального процессора (ЦП) необходимо выполнить «горячую» замену в модуле, где диагностика активна.

Более подробно процедура просмотра диагностики центрального процессора или других модулей шины описана в руководстве пользователя серии Nexto – MU214600.

### 6.1.2. Диагностика с помощью светодиодного индикатора

В данном изделии присутствует светодиод для диагностической индикации (LED DG) и светодиод индикации событий сторожевого таймера (LED WD). В таблицах 152 и 153 приведены значения каждого состояния и их соответствующие описания.

#### 6.1.2.1. DG (Диагностика)

Зеленый	Красный	Описание	Причины	Приоритет
Выкл.	Выкл.	Не используется	Отсутствие источника питания. Аппаратная неисправность	-
Вкл.	Выкл.	Все приложения в режиме выполнения (Run)	-	3 (низкий)
Выкл.	Вкл.	Все приложения в режиме останова (Stop)	-	3 (низкий)
Мигание 2x	Выкл.	Модули шины с диагностикой	По крайней мере, модуль шины, включающий центральный процессор (ЦП), имеет активную диагностику.	1
Мигание 3x	Выкл.	Форсирование данных	Некоторая область памяти принудительно заполняется пользователем через MasterTool IEC XE	2
Выкл.	Мигание 4x	Ошибка конфигурации или аппаратная ошибка в шине	Шина повреждена или неправильно сконфигурирована	0 (высокий)

Таблица 152: Описание состояний светодиодов диагностики

## 6.1.2.2. WD (Сторожевой таймер)

Красный светодиод	Описание	Причины	Приоритет
Выкл.	Нет индикации сторожевого таймера	Нормальная работа	3 (низкий)
Мигание 1х	Сторожевой таймер программного обеспечения	Сторожевой таймер пользовательского приложения	2
Вкл.	Сторожевой таймер аппаратного обеспечения	Поврежденный модуль и/или поврежденная операционная система	1 (высокий)

Таблица 153: Описание состояний светодиодов сторожевого таймера

Примечания:

Сторожевой таймер программного обеспечения: чтобы снять индикацию сторожевого таймера, необходимо выполнить сброс приложения либо выключить и снова включить центральный процессор (ЦП). Данный сторожевой таймер срабатывает, когда время выполнения пользовательского приложения превышает сконфигурированное время сторожевого таймера.

Диагностика может быть проверена в переменной `Exception.wExceptionCode`, см. таблицу 160.

Сторожевой таймер аппаратного обеспечения: для сброса любой индикации сторожевого таймера, например, светодиода WD или операнда `Reset.bWatchdogReset`, модуль должен быть отключен от источника питания.

Для проверки условий применения при перезапуске модуля см. конфигурации в таблице 36.

## 6.1.2.3. Светодиодные индикаторы разъема RJ45

Оба светодиодных индикатора, размещенных в разъемах RJ45, помогают пользователю при обнаружении проблем в установленной физической сети, указывая на скорость сетевого соединения и наличие трафика связи по интерфейсу. Значение светодиодов представлено в таблице, приведенной ниже.

Желтый	Зеленый	Значение
◦	◦	Сетевое ПОДКЛЮЧЕНИЕ отсутствует.
•	◦	Сетевое подключение 10 Мбайт/с
•	•	Сетевое подключение 100 Мбайт/с
X	-	Возникновение передачи или приема по сети Ethernet, для или к данному IP-адресу. Мигает по запросу центрального процессора (ЦП) Nexto, а не при каждой передаче или приеме, другими словами, может мигать с меньшей частотой, чем реальная частота передачи или приема

Таблица 154: Значение светодиодов локальной сети Ethernet

## 6.1.3. Диагностика с помощью веб-страницы

Помимо представленных ранее возможностей, серия Nexto предлагает пользователю инновационный инструмент для обеспечения доступа к диагностике и состояниям работы системы через веб-страницу.

Использование, помимо динамичности, является очень интуитивно понятным и облегчает работу пользователя. Использование системы диспетчерского управления может быть заменено, если оно ограничивается проверкой состояния системы.

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

Чтобы перейти на нужную веб-страницу центрального процессора (ЦП), достаточно воспользоваться стандартным навигатором (Internet Explorer 7 и выше, Mozilla Firefox 3.0 и выше и Google Chrome 8 и выше) и набрать в адресной строке IP-адрес центрального процессора (Например: <http://192.168.1.1>).

Сначала будет представлена информация о центральном процессоре в соответствии с рисунком 128:

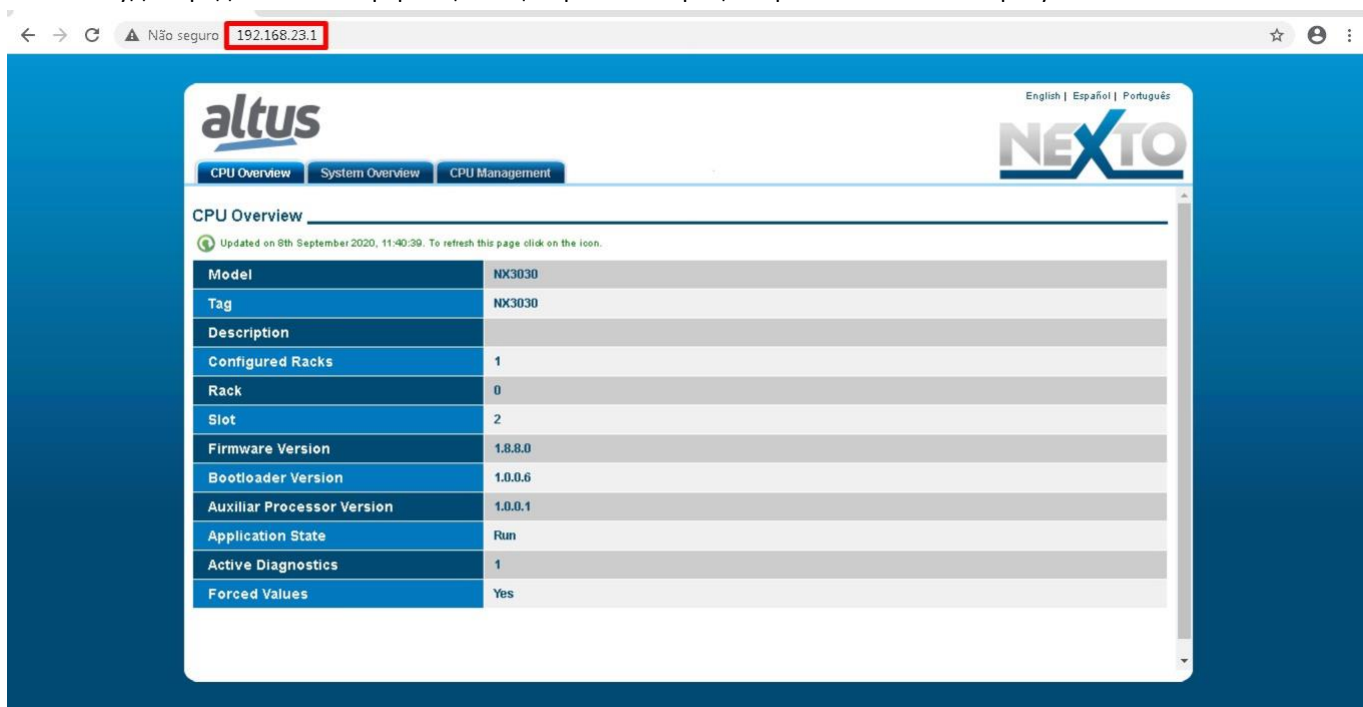


Рисунок 128: Начальный экран

Также имеется вкладка «Обзор системы» («System Overview»), которая может быть визуализирована через стойку либо список присутствующих модулей (опция в правой части экрана). Пока на центральном процессоре (ЦП) не присутствуют какие-либо приложения, на данной странице будет отображаться конфигурация с самой большой доступной стойкой и стандартным источником питания, подключенным к центральному процессору. При использовании визуализации стойки модули, имеющие диагностику, мигают и принимают красный цвет, как показано на рисунке 129. В противном случае отображается список с подключенными к системе модулями, тегами (метками) и номером активной диагностики:

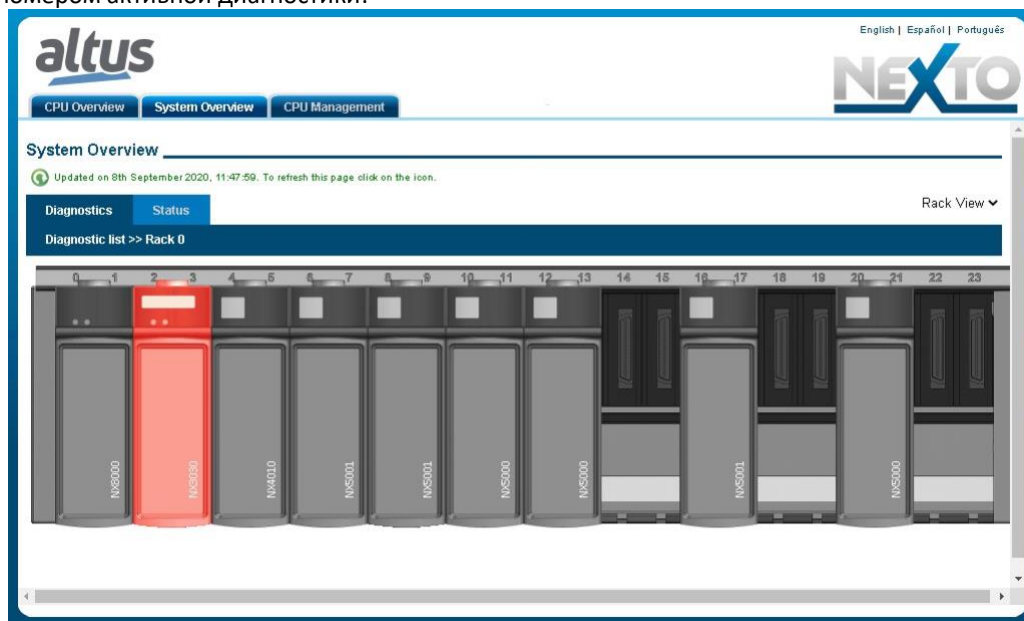


Рисунок 129: Информация о системе

При нажатии кнопки модуля с диагностикой отображается активная(-ые) диагностика(-и) модуля, как показано на рисунке 130:

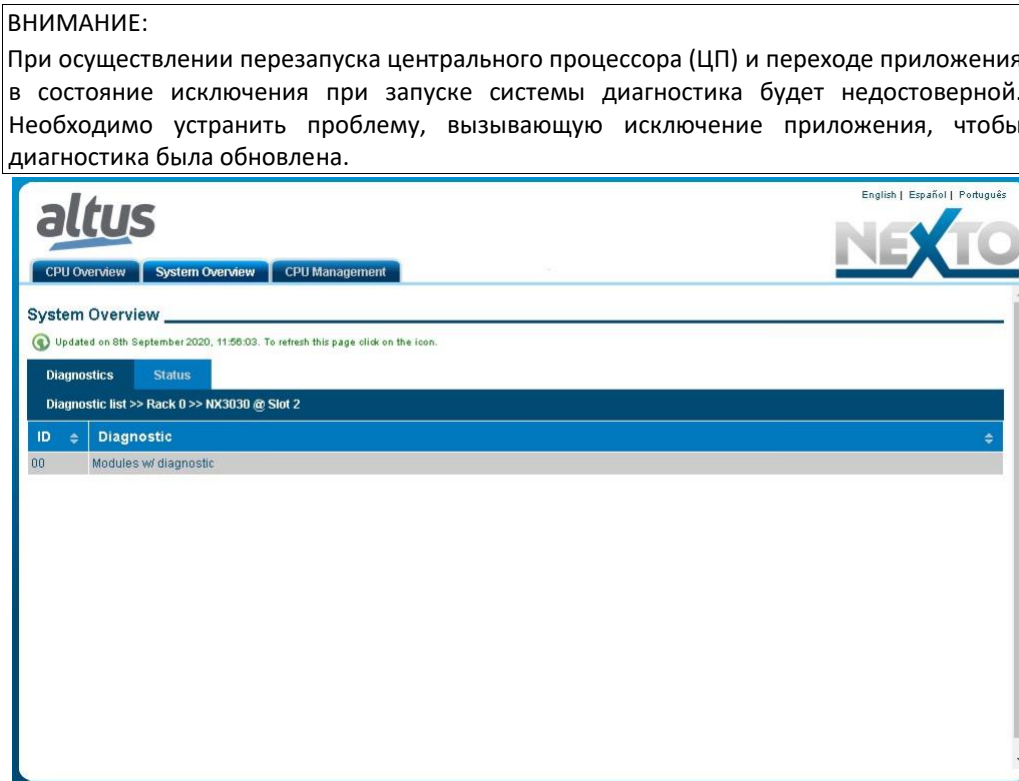


Рисунок 130: Диагностика системы

В случае выбора вкладки «Состояние» (Status) на экране отображается состояние всех подробных диагностик, как показано на рисунке 131:

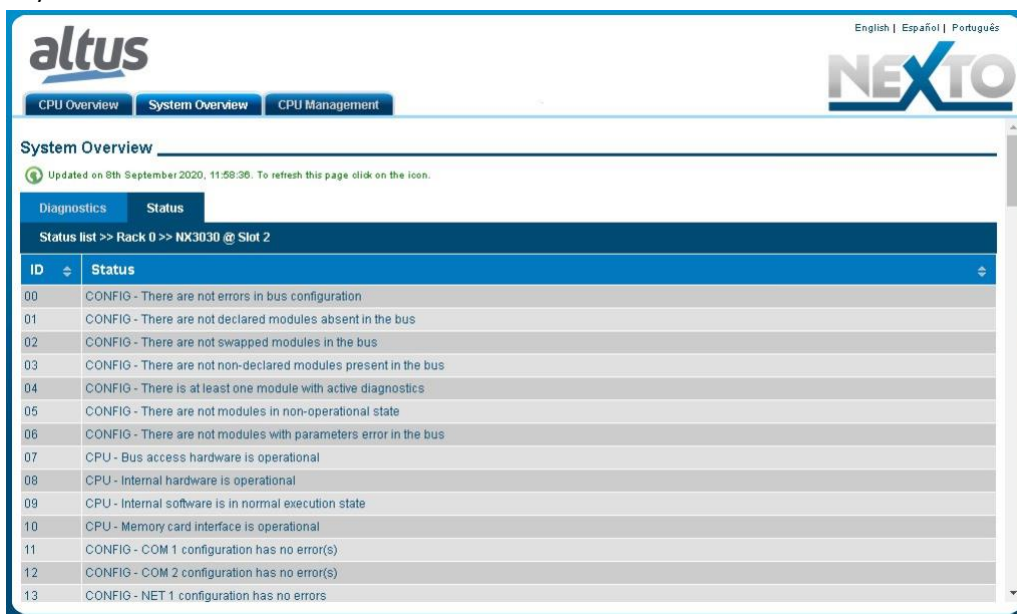


Рисунок 131: Состояние системы

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

Пользователь может выбрать для визуализации три языковых варианта: Португальский, Английский и Испанский. Для этого достаточно переключить правое верхнее меню на нужный язык. В дополнение к этому, на вкладке «Управление центральным процессором» (CPU Management) имеются и другие функции, такие как «Обновление версии микропрограммы (прошивки)» (Firmware Update) и «Простой протокол сетевого управления» (SNMP).

Вкладка «Обновление версии микропрограммы (прошивки)» (Firmware Update) ограничена для пользователя, то есть предназначена только для внутреннего использования компанией Altus. В случаях, когда обновление производится удаленно (например, через радио- или спутниковое соединение), минимальная скорость канала связи должна составлять 128 Кбит/с.

### 6.1.4. Диагностический проводник

Диагностический проводник (Diagnostic Explorer) представляет собой включение диагностики через веб-страницу в MasterTool IEC XE, чтобы сделать процесс более быстрым и непосредственным.

Доступ к данной функции осуществляется двумя способами:

- Зайдите в дерево устройств, расположенное слева на экране MasterTool IEC XE, в опцию «Диагностический проводник» (Diagnostic Explorer) и внесите нужный IP-адрес в поле, указанное на рисунке 132. Помните о том, что для отображения страницы диагностики пользователь должен быть подключен к центральному процессору (ЦП) (раздел «Начальное программирование - Вход в систему»).

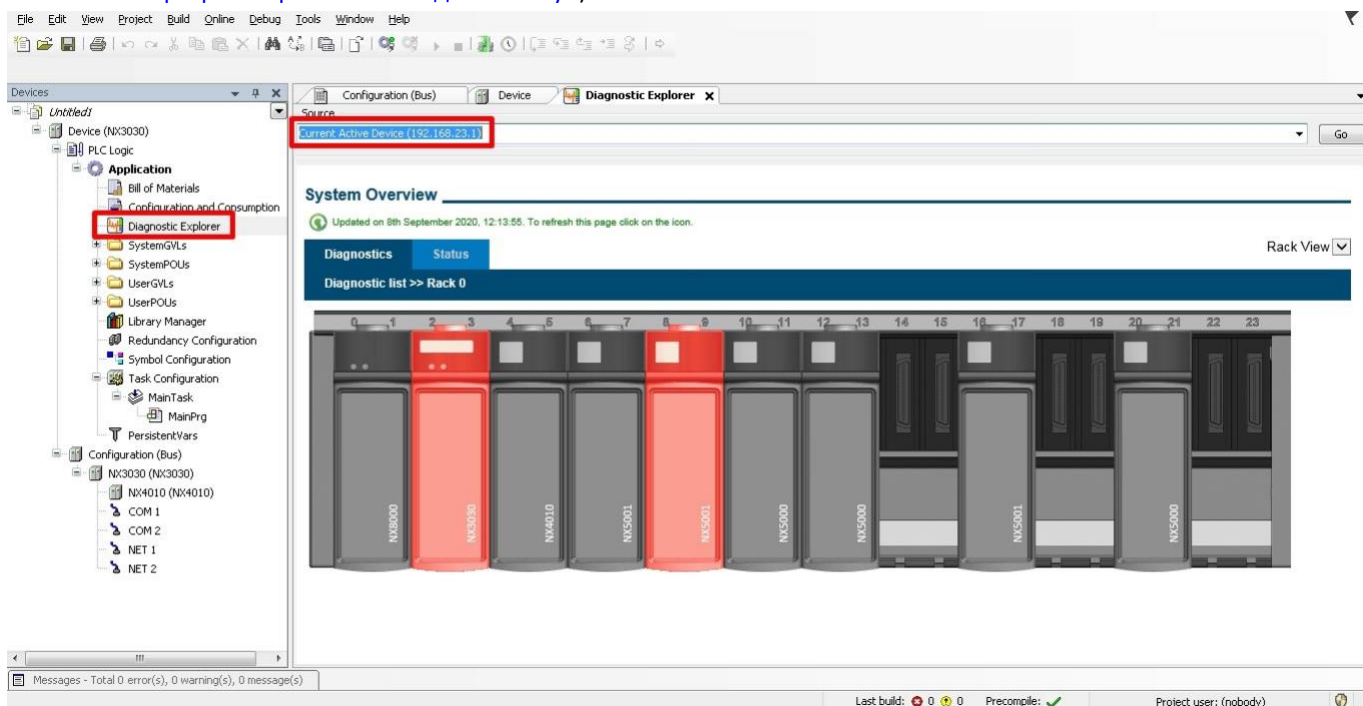


Рисунок 132: Экран диагностического проводника

- При выполнении щелчка правой кнопкой мыши по модулю и выборе пункта «Диагностика» («Diagnostics»), открывается Диагностический проводник (Diagnostic Explorer), перенаправляющий на страницу состояния модуля.

### 6.1.5. Диагностика с помощью переменных

В центральных процессорах (ЦП) серии Nexto имеется множество переменных для индикации диагностики. В списке глобальных переменных (GVL) System\_Diagnostics, автоматически создаваемом MasterTool IEC XE, присутствуют структуры данных с диагностикой всех модулей, объявленных на шине, отображенные на переменные прямого представления %Q и определяемые символически через директиву AT.

В приведенной ниже таблице представлено разделение диагностических байтов/слов:

Байт	Описание
------	----------

от 0 до 3	Сводная диагностика центрального процессора (ЦП).
от 4 до 558	Подробная диагностика центрального процессора (ЦП).

Таблица 155: Подразделение диагностики центральных процессоров (ЦП)

## 6.1.5.1. Сводная диагностика

В приведенной ниже таблице показано значение каждого бита сводной диагностики центрального процессора (ЦП):

Прямая переменная		Диагностическое сообщение	Переменная директивы AT DG_Module.tSummarized.*	Описание
Переменная	Бит			
-	-	NO DIAG («Диагностика не выполняется»)	-	Нет активной диагностики.
%QB(n)	0	CONFIG. MISMATCH («Несоответствие конфигурации»)	bConfigMismatch	«TRUE» («ИСТИНА») – Проблема конфигурации шины, поскольку модуль вставлен в неправильное положение.
				«FALSE» («ЛОЖЬ») – Правильная конфигурация.
	1	ABSENT MODULES («Отсутствующие модули»)	bAbsentModules	«TRUE» («ИСТИНА») – Один или несколько заявленных модулей отсутствуют.
				«FALSE» («ЛОЖЬ») – Все заявленные модули обнаружены на шине.
	2	SWAPPED MODULES («Переставленные модули»)	bSwappedModules	«TRUE» («ИСТИНА») – В шине имеются измененные модули.
				«FALSE» («ЛОЖЬ») – В шине нет измененных модулей.
	3	NON-DECLARED MODULES («Необъявленные модули»)	bNonDeclaredModules	«TRUE» («ИСТИНА») – Один или несколько модулей шины не объявлены в конфигурации.
				«FALSE» («ЛОЖЬ») – Все модули объявлены.
	4	MODULES W/ DIAGNOSTICS («Модули с диагностикой»)	bModulesWithDiagnostic	«TRUE» («ИСТИНА») – Один или несколько модулей в шине находятся в состоянии активной диагностики.
				«FALSE» («ЛОЖЬ») – Нет активной диагностики на модулях шины.
	5	MODULES W/ FATAL ERROR («Модули с фатальной ошибкой»)	bModuleFatalError	«TRUE» («ИСТИНА») – Один или несколько модулей в шине находятся в состоянии фатальной ошибки.

6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

				«FALSE» («ЛОЖЬ») – Все модули работают правильно.
	6	MODULES W/ PARAM. ERROR («Модули с ошибкой параметризации»)	bModuleParameterError	«TRUE» («ИСТИНА») – Один или несколько модулей в шине содержат ошибку параметризации.
				«FALSE» («ЛОЖЬ») – Все модули являются параметризованными.
	7	BUS ERROR («Ошибка шины»)	bWHSBBusError	«TRUE» («ИСТИНА») – Ведущее устройство указывает на сбой в шине WHSB.
				«FALSE» («ЛОЖЬ») – Шина WHSB работает должным образом.
%QB(n+1)	0	HARDWARE FAILURE («Аппаратный сбой»)	bHardwareFailure	«TRUE» («ИСТИНА») – Аппаратный сбой центрального процессора (ЦП).
				«FALSE» («ЛОЖЬ») – Аппаратное обеспечение работает должным образом.
	1	SOFTWARE EXCEPTION («Программное исключение»)	bSoftwareException	«TRUE» («ИСТИНА») – Одно или несколько исключений, сгенерированных программным обеспечением.
				«FALSE» («ЛОЖЬ») – Отсутствие исключений, генерируемых в программном обеспечении.
	4	COM 1 CONF. ERROR («Ошибка конфигурации COM 1»)	bCOM1ConfigError	«TRUE» («ИСТИНА») – Во время или после конфигурирования последовательного интерфейса COM 1 произошла какая-то ошибка.
				«FALSE» («ЛОЖЬ») – Правильная конфигурация последовательного интерфейса COM 1. <i>ALSE – The COM 1 serial interface configuration is correct.</i>
	6	NET 1 CONF. ERROR («Ошибка конфигурации NET 1»)	bNET1ConfigError	«TRUE» («ИСТИНА») – Во время или после конфигурирования последовательного интерфейса NET 1 сети Ethernet произошла какая-то ошибка.
				«FALSE» («ЛОЖЬ») – Правильная конфигурация интерфейса NET 1 сети Ethernet.
Прямая переменная		Диагностическое сообщение	Переменная директивы AT DG_Module.tSummarized.*	Описание
Переменная	Бит			
%QB(n+2)	0	INVALID DATE/TIME («Неверная дата/время»)	bInvalidDateTime	«TRUE» («ИСТИНА») – Неверная дата/время.



6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

				«FALSE» («ЛОЖЬ») – Верная дата/время.
	1	RUNTIME RESET («Сброс времени выполнения операции»)	bRTSReset	«TRUE» («ИСТИНА») – Система RTS (система поддержки выполнения программ) была перезапущена как минимум один раз. Данная диагностика очищается только во время перезапуска системы. «FALSE» («ЛОЖЬ») – RTS (система поддержки выполнения программ) работает нормально.
	2	OTD SWITCH ERROR («Ошибка переключателя OTD (диагностика одним касанием)»)	bOTDSwitchError	«TRUE» («ИСТИНА») – Клавиша диагностики одним касанием (OTD) была заблокирована в течение более чем 20 с, когда на ЦП было подано напряжение. Очистка данной диагностики выполняется только во время перезапуска системы. «FALSE» («ЛОЖЬ») – Ключ в данный момент не заблокирован или был заблокирован, когда на центральный процессор (ЦП) подавалось напряжение.
	3	WRONG CPU SLOT («Неправильное гнездо (слот) центрального процессора (ЦП)»)	bWrongCPUSlot	«TRUE» («ИСТИНА») – Центральный процессор (ЦП) установлен в гнездо (слот), отличное от заявленного в пользовательском приложении. «FALSE» («ЛОЖЬ») – Центральный процессор (ЦП) правильно установлен в гнездо (слот), заявленное в пользовательском приложении.
%QB(n+3)	0	ABSENT RACK («Отсутствует стойка»)	bAbsentRacks	«TRUE» («ИСТИНА») – Отсутствует одна или несколько заявленных стоек. «FALSE» («ЛОЖЬ») – Нет отсутствующих стоек.
	1	DUPLICATED RACK («Повторяющаяся стойка»)	bDuplicatedRacks	«TRUE» («ИСТИНА») – Стойки с повторяющимся идентификационным номером. «FALSE» («ЛОЖЬ») – Нет стоек с повторяющимся идентификационным номером.
	2	INVALID RACK («Недействительная стойка»)	bInvalidRacks	«TRUE» («ИСТИНА») – Стойки с недействительным идентификационным номером. «FALSE» («ЛОЖЬ») – Отсутствие стоек с недействительным идентификационным номером.
	3	NON DECLARED RACK («Незаявленная стойка»)	bNonDeclaredRacks	«TRUE» («ИСТИНА») – Стойки с незаявленным идентификационным номером

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

			«FALSE» («ЛОЖЬ») – Отсутствие стоек с незаявленным идентификационным номером.
	4	DUPLICATED SLOT («Повторяющийся слот»)	bDuplicatedSlots
			«TRUE» («ИСТИНА») – Повторяющийся адрес слота.
			«TRUE» («ИСТИНА») – Нет повторяющихся адресов слотов

Таблица 156: Сводная диагностика центрального процессора (ЦП)

### Примечания:

Переменная прямого представления: «n» представляет собой значение, установленное в центральном процессоре (ЦП) с помощью программного обеспечения MasterTool IEC XE, например, при диагностике начального адреса.

Директива AT: В описании символических переменных, использующих директиву AT для преобразования в переменные прямой адресации, перед требуемой обобщенной диагностикой должен быть поставлен синтаксис DG\_Module.tSummarized, когда слово Module («Модуль») заменяется на используемый центральный процессор (ЦП). Например, для диагностики несовместимой конфигурации необходимо использовать переменную: DG\_NX3010.tSummarized.bConfigMismatch. Директива AT представляет собой слово, зарезервированное в программном обеспечении для программирования и используемое только для индикации диагностики.

Несоответствие конфигурации: Диагностика несоответствующей конфигурации генерируется, если один или несколько модулей стойки не соответствуют заявленным, то есть при отсутствии или разных условиях модулей. Вставленные в шину модули, которые не были заявлены в проекте, не учитываются.

Переставленные модули: Если в шине меняются местами только два модуля, то можно определить измененную диагностику. В противном случае проблема рассматривается как «Несовместимая конфигурация».

Модули с фатальной ошибкой: В случае, если диагностика модулей с фатальной ошибкой имеет значение «true» («истина»), необходимо проверить, какой именно модуль в шине является проблемным, и отправить его в службу технической поддержки компании Altus, поскольку в нем имеет место быть аппаратный сбой.

Модуль с ошибкой параметризации: Если диагностика ошибки параметризации имеет значение «true» («истина»), необходимо проверить правильность конфигурации модуля в шине, а также является ли версия микропрограммы (прошивки) и программного обеспечения MasterTool IEC XE верной. Если проблема возникла при установке модуля на шину, убедитесь в том, что модуль поддерживает «горячую» замену.

Ошибка шины: Считается фатальной ошибкой, прерывающей обеспечение доступа к модулям в шине. Если диагностика ошибки шины имеет значение «true» («истина»), возможно, возникла нештатная ситуация, связанная с выбранной конфигурацией «горячей» замены, либо аппаратная проблема в линиях связи шины, и, в таком случае, следует обратиться в службу технической поддержки компании Altus.

Аппаратный отказ: Если диагностика «Аппаратный отказ» имеет значение «true» («истина»), то центральный процессор (ЦП) необходимо отправить в службу технической поддержки компании Altus, поскольку в нем присутствуют проблемы, связанные с часами реального времени (RTC), вспомогательным процессором или другими аппаратными ресурсами.

Программное исключение: Если диагностика программного исключения имеет значение «true» («истина»), пользователь должен проверить свое приложение, чтобы убедиться в том, что оно не обращается к памяти неправильным образом. Если проблема сохраняется, необходимо обратиться в сектор технической поддержки компании Altus. Коды программных исключений описаны далее в таблице подробной диагностики центрального процессора (ЦП).

Диагностическое сообщение: Диагностические сообщения могут быть визуализированы на графическом дисплее центрального процессора (ЦП) с помощью клавиши диагностики одним касанием (OTD) или с помощью веб-страницы, через страницу диагностики центрального процессора (ЦП).

## 6.1.5.2. Подробная диагностика

Приведенные ниже таблицы содержат подробную диагностику центральных процессоров (ЦП) серии Nexto. Перед обращением к ним необходимо иметь в виду приведенные ниже замечания:

- Визуализация диагностических структур: Добавленные в проект диагностических структур можно заметить в пункте «Менеджер библиотек» (*Library Manager*) древовидного представления MasterTool IEC XE. Там можно увидеть все типы данных, определенные в структуре.
- Счетчики: Все счетчики диагностики центрального процессора (ЦП) возвращаются к нулю при превышении их предельного значения.
- Переменная прямого представления: «n» представляет собой значение, сконфигурированное в центральном процессоре (ЦП) через MasterTool IEC XE в качестве начального адреса диагностики.

Директива AT: При описании символических переменных, использующих директиву AT для их преобразования в переменные прямого отображения, перед требуемой обобщенной диагностикой следует использовать синтаксис *DG\_Module.tDetailed.*, где слово *Module* («Модуль») должно быть заменено на используемый центральный процессор (ЦП). Директива AT является зарезервированным словом устройства программирования (программатора), и некоторые символические переменные, использующие данную директиву, обозначают диагностику.

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QD(n+4)	DWORD	Target. dwCPUModel	NX3003 = 0x3003 NX3004 = 0x3004 NX3005 = 0x3005 NX3010 = 0x3010 NX3020 = 0x3020 NX3030 = 0x3030
%QB(n+8)	BYTE ARRAY(4)	Target. abyCPUVersion	Версия микропрограммы (прошивки).
%QB(n+12)	BYTE ARRAY(4)	Target. abyBootloaderVersion	Версия загрузчика операционной системы.

Таблица 157: Описание группы подробной диагностики Target («Цель»)

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QX(n+20).1	BIT	Hardware. bRTCFailure	Основной процессор не имеет возможности обмена данными с часов реального времени (RTC) (часы центрального процессора (ЦП)).
%QX(n+20).2	BIT	Hardware. bThermometerFailure <sup>1</sup>	Неисправность связи между термометром и основным процессором.

Таблица 158: Описание группы подробной диагностики Hardware («Аппаратное обеспечение»)

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QW(n+21)	WORD	Exception. wExceptionCode	Код исключения, генерируемый часами реального времени (RTS). См. таблицу 160.
%QB(n+23)	BYTE	Exception. byProcessorLoad	Уровень, в процентах (%), заряда в процессоре.

Таблица 159: Описание группы подробной диагностики Exception («Исключение»)

Примечание:

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

Код исключения: код исключения, сгенерированного RTS (системой поддержки выполнения программ), можно посмотреть ниже:

Код	Описание	Код	Описание
0x0000	Код исключения отсутствует.	0x0051	Нарушение доступа.
0x0010	Истекло время сторожевого таймера задачи IEC (сторожевой таймер программного обеспечения).	0x0052	Привилегированная инструкция.
0x0012	Ошибка конфигурации ввода/вывода.	0x0053	Сбой страницы.
0x0013	Ошибка контрольной суммы после загрузки программы.	0x0054	Переполнение стека.
0x0014	Ошибка промышленной шины.	0x0055	Недопустимое расположение.
0x0015	Ошибка обновления ввода/вывода.	0x0056	Неверный маневр.
0x0016	Превышено время цикла (выполнения).	0x0057	Защищенная страница.
0x0017	Слишком долгое обновление программы в диалоговом режиме («онлайн»).	0x0058	Двойной сбой.
0x0018	Внешние ссылки не разрешены.	0x0059	Неверный операционный код.
0x0019	Загрузка отклонена.	0x0100	Несоответствие типа данных.
0x001A	Проект не загружен, поскольку сохраняемые переменные не могут быть перераспределены.	0x0101	Превышен лимит массивов.
0x001B	Проект не загружен и удален.	0x0102	Деление на ноль.
0x001C	Закончился стек памяти.	0x0103	Переполнение.
0x001D	Сохраняющая память повреждена и не может быть преобразована.	0x0104	Продолжение выполнения невозможно.
0x001E	Проект может быть загружен, но в дальнейшем приводит к падению.	0x0105	Обнаружен сторожевой таймер в загрузке процессора всей задачи IEC.
0x0021	Цель запускаемого приложения не совпадает с текущей целью.	0x0150	FPU: Ошибка не указана.
0x0022	Ошибка запланированных задач.	0x0151	FPU: Операнд не является нормальным.
		0x0152	FPU: Деление на ноль.
0x0023	Код исключения отсутствует.	0x0153	FPU: Неточный результат.
0x0024	Сохраняющая идентичность не соответствует текущей идентичности программы загрузочного проекта	0x0154	FPU: Недопустимая операция.

6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

0x0025	Сбой конфигурации задачи IEC.	0x0155	FPU: Переполнение.
0x0026	Приложение работает с неправильной целью.	0x0156	FPU: Проверка стека.
0x0050	Недействительная инструкция.	0x0157	FPU: Недостаточное заполнение.

Таблица 160: Коды исключений RTS

Прямое представление	Размер	AT DG Variable Module.tDetailed.*	Описание
%QB(n+25)	BYTE	RetainInfo. byCPUInitStatus	Состояние запуска центрального процессора (ЦП): 01: Горячий запуск 02: Теплый запуск 03: Холодный запуск Примечание: Данные переменные сбрасываются при каждом включении питания.
%QW(n+26)	WORD	RetainInfo. wCPUColdStartCounter	Счетчик холодных запусков: Он будет увеличиваться только в результате горячего извлечения центрального процессора в шине, но не в результате команды холодного сброса MasterTool IEC XE (MasterTool IEC XE Cold Reset) (от 0 до 65535).
%QW(n+28)	WORD	RetainInfo. wCPUWarmStartCounter	Счетчик теплых запусков: Инкрементируется только во время последовательности включений системы, но не по команде горячего сброса MasterTool IEC XE (MasterTool IEC XE Hot Reset) (0...65535).
%QW(n+30)	WORD	RetainInfo. wReserved_0	Зарезервировано.
%QW(n+32)	WORD	RetainInfo. wRTSResetCounter	Счетчик сбросов, выполненных RTS (система поддержки выполнения программ) (0...65535).

Таблица 161: Подробная диагностика группы RetainInfo

Прямое представление	Размер	AT DG Module.tDetailed.* Variable	Описание
%QX(n+36).0	BIT	Reset. bBrownOutReset	При последнем запуске центральный процессор (ЦП) был перезапущен из-за сбоя в источнике питания.
%QX(n+36).1	BIT	Reset. bWatchdogReset	При последнем запуске центральный процессор (ЦП) был перезапущен из-за активного сторожевого таймера.

Таблица 162: Описание группы подробной диагностики Reset («Сброс»)

Примечание:

Сброс кратковременного провала напряжения питания: Диагностика сброса кратковременного провала напряжения питания (brownout reset) справедлива только в том случае, если источник питания превышает минимальный предел, предусмотренный его техническими характеристиками, оставаясь в режиме низкого напряжения, т.е. без каких-либо прерываний. Центральный процессор (ЦП) определит падение напряжения питания и выдаст диагностику перебоа в подаче электропитания. Когда напряжение восстановится, центральный процессор

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

автоматически перезагрузится и выдаст сообщение о диагностике сброса кратковременного провала напряжения питания (brownout reset).

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QX(n+37).0	BIT	Thermometer. bOverTemperatureAlarm <sup>1</sup>	Сигнал тревоги формируется из-за внутренней температуры 85°C или выше.
%QX(n+37).1	BIT	Thermometer. bUnderTemperatureAlarm <sup>1</sup>	Сигнал тревоги, вызванный внутренней температурой 0°C или ниже.
%QD(n+38)	DINT	Thermometer. diTemperature <sup>1</sup>	Температура считывается внутренним датчиком центрального процессора (ЦП).

Таблица 163: Описание группы подробной диагностики Thermometer («Термометр»)

Примечание:

Температура: Для того чтобы увидеть температуру непосредственно в адресе памяти, необходимо произвести преобразование, поскольку используемым размером данных является DINT, а мониторинг осуществляется в 4 байта. Поэтому рекомендуется использовать связанную символическую переменную, поскольку она уже предоставляет конечное значение температуры.

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QB(n+42)	BYTE	Serial.COM1. byProtocol	Протокол, выбранный COM 1: 00: Без протокола 01: Ведущее устройство MODBUS RTU (MODBUS RTU Master) 02: Ведомое устройство MODBUS RTU (MODBUS RTU Slave) 03: Другой протокол
%QD(n+43)	DWORD	Serial.COM1. dwRXBytes	Счетчик символов, полученных с COM 1 (от 0 до 4294967295).
%QD(n+47)	DWORD	Serial.COM1. dwTXBytes	Счетчик символов, переданных с COM 1 (от 0 до 4294967295).
%QW(n+51)	WORD	Serial.COM1. wRXPendingBytes	Количество символов, оставшихся в буфере чтения в COM 1 (от 0 до 65535).
%QW(n+53)	WORD	Serial.COM1. wTXPendingBytes	Количество символов, оставшихся в буфере передачи в COM 1 (от 0 до 65535).
%QW(n+55)	WORD	Serial.COM1. wBreakErrorCounter	Счетчики ошибок COM 1 (от 0 до 65535). Эти счетчики перезапускаются при следующих условиях: Включение . Конфигурирование последовательного порта COM 1 Удаление очередей RX и TX
%QW(n+57)	WORD	Serial.COM1. wParityErrorCounter	
%QW(n+59)	WORD	Serial.COM1. wFrameErrorCounter	
%QW(n+61)	WORD	Serial.COM1. wRXOverrunCounter	

Таблица 164: Описание группы подробной диагностики Serial COM 1

Примечание:

Счетчик ошибок четности: Если последовательный COM 1 сконфигурирован без четности, то при получении сообщения с другой четностью значение данного счетчика ошибок не будет увеличиваться. В этом случае будет показана ошибка фрейма (кадра).

Прямое представление	Размер	Переменная директивы AT DG_Modulo.tDetailed.*	Описание
----------------------	--------	---	----------

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

%QX(n+92).0	BIT	Ethernet.NET1. bLinkDown	Указывает состояние соединения в сети NET 1.
%QW(n+93)	WORD	Ethernet.NET1. wProtocol	Выбранный протокол в NET 1: 00: Нет протокола
%QX(n+93).0	BIT	Ethernet.NET1. wProtocol. bMODBUS_RTU_ETH_Client	Клиент MODBUS RTU через TCP (MODBUS RTU client via TCP)
%QX(n+93).1	BIT	Ethernet.NET1. wProtocol. bMODBUS_ETH_Client	Сервер MODBUS TCP (MODBUS TCP Client)
%QX(n+93).2	BIT	Ethernet.NET1. wProtocol. bMODBUS_RTU_ETH_Server	Сервер MODBUS RTU через TCP (MODBUS RTU Server via TCP)
%QX(n+93).3	BIT	Ethernet.NET1. wProtocol. bMODBUS_ETH_Server	Сервер MODBUS TCP (MODBUS TCP Server)
%QB(n+95)	STRING (15)	Ethernet.NET1. szIP	IP-адрес 1 NET
%QB(n+111)	STRING (15)	Ethernet.NET1. szMask	Маска подсети NET 1
%QB(n+127)	STRING (15)	Ethernet.NET1. szGateway	Адрес шлюза NET 1
%QB(n+143)	STRING (17)	Ethernet.NET1. szMAC	MAC-адрес NET 1
%QB(n+161)	BYTE ARRAY(4)	Ethernet.NET1. abyIP	IP-адрес 1 NET
%QB(n+165)	BYTE ARRAY(4)	Ethernet.NET1. abyMask	Маска подсети NET 1
%QB(n+169)	BYTE ARRAY(4)	Ethernet.NET1. abyGateway	Адрес шлюза NET 1
Прямое представление	Размер	Переменная директивы AT DG_Modulo.tDetailed.*	Описание
%QB(n+173)	BYTE ARRAY(6)	Ethernet.NET1. abyMAC	MAC-адрес NET 1
%QD(n+179)	DWORD	Ethernet.NET1. dwPacketsSent	Счетчик пакетов, отправленных через порт NET 1 (от 0 до 4294967295).
%QD(n+183)	DWORD	Ethernet.NET1. dwPacketsReceived	Счетчик пакетов, полученных через порт NET 1 (от 0 до 4294967295).
%QD(n+187)	DWORD	Ethernet.NET1. dwBytesSent	Счетчик байтов, отправленных через порт NET 1 (от 0 до 4294967295).
%QD(n+191)	DWORD	Ethernet.NET1. dwBytesReceived	Счетчик принятых байтов через порт NET 1 (от 0 до 4294967295).
%QW(n+199)	WORD	Ethernet.NET1. wTXDropErrors	Счетчик ошибок передачи через порт NET 1 (от 0 до 65535).
%QW(n+201)	WORD	Ethernet.NET1. wTXCollisionErrors	Счетчик ошибок в буфере передачи через порт NET 1 (от 0 до 65535).
%QW(n+209)	WORD	Ethernet.NET1. wRXDropErrors	Счетчик потери соединения при передаче через порт NET 1 (от 0 до 65535).
%QW(n+211)	WORD	Ethernet.NET1. wRXFrameErrors	Счетчик ошибок коллизий при передаче через порт NET 1 (от 0 до 65535).

Таблица 165: Подробная диагностика группы локальной сети Ethernet NET1

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QB(n+219)	BYTE	UserFiles. byMounted	Указывает, может ли память, используемая для записи пользовательских файлов, принимать данные.
%QD(n+220)	DWORD	UserFiles. dwFreeSpacekB	Свободное место в памяти для файлов пользователя в Кбайт.
%QD(n+224)	DWORD	UserFiles. dwTotalSizekB	Объем памяти для хранения пользовательских файлов в Кбайтах.

Таблица 166: Описание группы подробной диагностики UserFiles

### Примечание:

Пользовательский раздел: Пользовательский раздел представляет собой область памяти, зарезервированной для хранения данных в центральном процессоре (ЦП). Например: файлы с расширением PDF, файлы с расширением DOC и другие данные.

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QB(n+229)	BYTE	UserLogs. byMounted	Состояние памяти, в которую вставлены пользовательские журналы.
%QW(n+230)	WORD	UserLogs. wFreeSpacekB	Свободное место в памяти пользовательских журналов в Кбайт.
%QW(n+232)	WORD	UserLogs. wTotalSizekB	Объем памяти пользовательских журналов в Кбайтах.

Таблица 167: Описание группы подробной диагностики UserLogs

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QB(n+245)	BYTE	WHSB. byHotSwapAndStartupStatus	Информирует о нештатной ситуации в шине, вызвавшей останов приложения, для каждого режима горячей замены. Дополнительную информацию см. в таблице 169.
%QD(n+247)	DWORD ARRAY (32)	WHSB. adwRackIOErrorStatus	Выявление ошибок в модулях ввода/вывода, по отдельности. Более подробную информацию об этой диагностике см. в примечаниях ниже.
Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QD(n+375)	DWORD ARRAY (32)	WHSB. adwModulePresenceStatus	Состояние присутствия заявленных модулей ввода/вывода в шинах, индивидуально. Более подробную информацию о данной диагностике см. в примечаниях ниже.
%QB(n+503)	BYTE	WHSB. byWHSBBusErrors	Счетчик отказов в шине WHSB. Данный счетчик перезапускается при подаче напряжения (от 0 до 255).

Таблица 168: Описание группы подробной диагностики WHSB

### Примечания:

Диагностика ошибок модулей шины: каждое DWORD из данного диагностического массива представляет собой стойку, позиции которой представлены битами данных DWORD. Таким образом, бит-0 DWORD-0 эквивалентен



## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

нулевой позиции стойки с нулевым адресом. Каждый из данных битов является результатом логической операции ИЛИ между диагностикой несовместимой конфигурации (bConfigMismatch), отсутствующих модулей (bAbsentModules), переставленных модулей (bSwappedModules), модуля с фатальной ошибкой (bModuleFatalError) и рабочим состоянием модуля в определенной позиции.

Состояние присутствия модуля: Каждое DWORD из данного диагностического массива представляет собой стойку, позиции которой представлены битами данных DWORD. Таким образом, бит-0 из DWORD-0 эквивалентен нулевой позиции стойки с нулевым адресом. Таким образом, если модуль присутствует, то данный бит будет истинным. Важно отметить, что данная диагностика справедлива для всех модулей, за исключением блоков питания, центральных процессоров (ЦП) и не объявленных модулей, т.е. тех, которые не находятся в стойке на соответствующей позиции (бит остается в значении false («ложь»)).

Ситуации, при которых происходит останов приложения: Ниже приведены коды возможных ситуаций, при которых происходит останов приложения:

Код	Счетное значение	Описание
00	INITIALIZING	Данное состояние представлено в то время, когда другие состояния не готовы.
01	RESET_WATCHDOG	Приложение в режиме останова из-за аппаратного сброса сторожевого таймера или сброса времени выполнения, когда опция «Start User Application After a Watchdog Reset» («Запуск пользовательского приложения после сброса сторожевого таймера») не отмечена.
02	ABSENT_MODULES_HOT_SWAP_DISABLED	Диагностика «Приложение в режиме останова из-за отсутствия модулей» устанавливается, если режим горячей замены «Отключен» или «Отключен, только для заявленных модулей».
03	CFG_MISMATCH_HOT_SWAP_DISABLED	Диагностика «Приложение в режиме останова из-за несоответствия конфигурации» устанавливается, если режим горячей замены «Отключен» или «Отключен, только для заявленных модулей».
04	ABSENT_MODULES_HOT_SWAP_STARTUP_CONSISTENCY	Диагностика «Приложение в режиме останова из-за отсутствия модулей» устанавливается, если режим горячей замены «Включен, с согласованностью при запуске» или «Включен, с согласованностью при запуске только для объявленных модулей».
05	CFG_MISMATCH_HOT_SWAP_STARTUP_CONSISTENCY	Диагностика «Приложение в режиме останова из-за несовместимой конфигурации» устанавливается, если режим горячей замены «Включен, с согласованностью запуска» или «Включен, с согласованностью запуска только для объявленных модулей».
06	APPL_STOP_ALLOWED_TO_RUN	Приложение находится в режиме останова (Stop Mode) и все непротиворечивости успешно выполнены. Приложение может быть переведено в режим выполнения (Run Mode).
07	APPL_STOP_MODULES_NOT_READY	Приложение находится в режиме останова (Stop Mode), все непротиворечивости выполнены успешно, но модули ввода/вывода не могут запустить систему. Перевести приложение в режим выполнения (Run Mode) невозможно.
08	APPL_STOP_MODULES_GETTING_READY_TO_RUN	Приложение находится в режиме останова (Stop Mode) и все непротиворечивости успешно выполнены. Модули ввода/вывода готовятся к запуску системы. Перевести приложение в режим выполнения (Run Mode) невозможно.
09	NORMAL_OPERATING_STATE	Приложение в режиме выполнения (Run Mode)

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

10	MODULE_CONSISTENCY_OK	Внутреннее использование.
Код	Счетное значение	Описание
11	APPL_STOP_DUE_TO_EXCEPTION	Приложение в режиме останова из-за исключения в центральном процессоре (ЦП).
12	DUPLICATED_SLOT_HOT_SWAP_DISABLED	Приложение находится в режиме останова (Stop Mode) из-за установки диагностики дублированных слотов, когда для режима «горячей» замены установлено значение «Выключено» или «Выключено, только для объявленных модулей».
13	DUPLICATED_SLOT_HOT_SWAP_STARTUP_CONSISTENCY	Приложение находится в режиме останова из-за диагностики дублированных слотов, когда режим «горячей» замены установлен в значение «Включено, с согласованностью запуска» или «Включено, с согласованностью запуска только для объявленных модулей».
14	DUPLICATED_SLOT_HOT_SWAP_ENABLED	Приложение находится в режиме останова из-за диагностики дублированных слотов, если для режима «горячей» замены установлено значение «Включено, без согласованности запуска».
15	NON_DECLARED_MODULE_HOT_SWAP_STARTUP_CONSISTENCY	Приложение находится в режиме останова из-за диагностики необъявленных модулей, когда режим «горячей» замены установлен на «Включено, с согласованностью при запуске» установлен.
16	NON_DECLARED_MODULE_HOT_SWAP_DISABLED	Приложение находится в режиме останова из-за того, что диагностика необъявленных модулей установлена, когда режим «горячей» замены «Выключен».

Таблица 169: Коды ситуаций, в которых происходит останов приложения

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QB(n+504)	BYTE	Application. byCPUState	Информирует о состоянии работы центрального процессора (ЦП): 01: Все пользовательские приложения находятся в режиме работы 03: Все пользовательские приложения находятся в режиме останова
%QX(n+505).0	BIT	Application. bForcedIOs	Имеется одна или несколько точек форсированного (принудительного) ввода/вывода.

Таблица 170: Описание группы подробной диагностики Application («Приложение»)

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QD(n+506)	DWORD	Rack. dwAbsentRacks	Каждый бит представляет собой идентификационный номер стойки, если бит равен TRUE («ИСТИНА»), то это означает, что стойка отсутствует.
%QD(n+510)	DWORD	Rack. dwDuplicatedRacks	Каждый бит представляет идентификационный номер стойки, если бит равен TRUE («ИСТИНА»), то это означает, что более одной стойки имеют одинаковый идентификационный номер.

6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

%QD(n+514)	DWORD	Rack.dwNonDeclaredRacks	Каждый бит представляет идентификационный номер стойки, если бит равен TRUE («ИСТИНА»), то это означает, что имеется стойка, сконфигурированная с необъявленным идентификационным номером.
------------	-------	-------------------------	--

Таблица 171: Описание группы подробной диагностики Rack («Стойка»)

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QD(n+520)	DWORD	ApplicationInfo.dwApplicationCRC	32-битный CRC приложения. При модификации приложения и передаче его в центральный процессор (ЦП) генерируется новый CRC.

Таблица 172: Описание группы подробной диагностики ApplicationInfo

Прямое представление	Размер	Переменная директивы AT DG_Module.tDetailed.*	Описание
%QX(n+532).0	BIT	SNTP.bServiceEnabled	Служба простого протокола сетевого времени (SNTP).
%QB(n+533)	BYTE	SNTP.byActiveTimeServer	Указывает, какой сервер является активным: 00: Нет активного сервера. 01: Активный первичный сервер. 02: Активный вторичный сервер.
%QW(n+534)	WORD	SNTP.wPrimaryServerDownCount	Счетчик времени, в течение которого первичный сервер недоступен (от 0 до 65535).
%QW(n+536)	WORD	SNTP.wSecondaryServerDownCount	Счетчик времени недоступности вторичного сервера (от 0 до 65535).
%QD(n+538)	DWORD	SNTP.dwRTCTimeUpdatedCount	Счетчик времени обновления часов реального времени (RTC) службой простого протокола сетевого времени (SNTP) (от 0 до 4294967295).
%QB(n+542)	BYTE	SNTP.byLastUpdateSuccessful	Указывает статус последнего обновления: 00: Обновление не производилось. 01: Последнее обновление прошло неудачно. 02: Последнее обновление прошло успешно.
%QB(n+543)	BYTE	SNTP.byLastUpdateTimeServer	Указывает, какой сервер использовался при последнем обновлении: 00: Нет обновления. 01: Первичный сервер. 02: Вторичный сервер.
%QB(n+544)	BYTE	SNTP.sLastUpdateTime.byDayOfMonth	День последнего обновления часов реального времени (RTC).
%QB(n+545)	BYTE	SNTP.sLastUpdateTime.byMonth	Месяц последнего обновления часов реального времени (RTC).
%QW(n+546)	WORD	SNTP.sLastUpdateTime.wYear	Год последнего обновления часов реального времени (RTC).
%QB(n+548)	BYTE	SNTP.sLastUpdateTime.byHours	Час последнего обновления часов реального времени (RTC).

%QB(n+549)	BYTE	SNTP. sLastUpdateTime. byMinutes	Минута последнего обновления часов реального времени (RTC).
%QB(n+550)	BYTE	SNTP. sLastUpdateTime. bReservedAlign	Зарезервировано для выравнивания.
%QB(n+551)	BYTE	SNTP. sLastUpdateTime. bySeconds	Секунда последнего обновления часов реального времени (RTC).
%QW(n+552)	WORD	SNTP. sLastUpdateTime. wMilliseconds	Миллисекунда последнего обновления часов реального времени (RTC).

Таблица 173: Описание группы подробной диагностики SNTP

### 6.1.6. Диагностика с помощью функциональных блоков

Функциональные блоки позволяют визуализировать некоторые параметры, к которым невозможно получить доступ иным способом. Функции, касающиеся расширенной диагностики, находятся в библиотеке *NextoStandard* описаны ниже.

#### 6.1.6.1. Функция «GetTaskInfo»

Данная функция возвращает информацию о задаче конкретного приложения.



Рисунок 133: Функция GetTaskInfo

Ниже описаны параметры, которые необходимо передать в функцию, чтобы она вернула информацию о приложении.

Входной параметр	Тип	Описание
psAppName	POINTER TO STRING	Имя приложения.
psTaskName	POINTER TO STRING	Имя задачи.
pstTaskInfo	POINTER TO stTask-Info	Указатель для получения информации о приложении.

Таблица 174: Входные параметры GetTaskInfo

Данные, возвращаемые функцией через указатель, сообщенный во входных параметрах, описаны в таблице ниже.

Возвращаемые параметры	Размер	Описание
dwCurScanTime	DWORD	Время цикла задачи (выполнения) с разрешением 1 мкс.
dwMinScanTime	DWORD	Минимальное время цикла задачи с разрешением 1 мкс.
dwMaxScanTime	DWORD	Максимальное время цикла задачи с разрешением 1 мкс.

dwAvgScanTime	DWORD	Среднее время цикла задачи с разрешением 1 мкс.
dwLimitMaxScan	DWORD	Максимальное время цикла задачи до срабатывания сторожевого таймера.
dwIECCycleCount	DWORD	Счетчик циклов ИЕС.

Таблица 175: Выходные параметры GetTaskInfo

Возможные значения ERRORCODE:

- NoError: выполнено успешно;
- TaskNotPresent: желаемая задача не существует.

Пример использования на языке структурированного текста (ST):

```

ПРОГРАММА UserPrg VAR
sAppName : STRING; psAppName : POINTER TO
STRING; sTaskName : STRING; psTaskName :
POINTER TO STRING; pstTaskInfo : POINTER TO
stTaskInfo;
TaskInfo : stTaskInfo;
Информация : ERRORCODE;
END_VAR //INPUTS:
sAppName := 'Application'; // Переменная получает имя приложения.
psAppName := ADR(sAppName); // Указатель с названием приложения. sTaskName := 'MainTask';
// Переменная получает имя задачи. psTaskName := ADR(sTaskName); // Указатель с именем задачи.
pstTaskInfo := ADR(TaskInfo); // Указатель, получающий информацию о задаче.
// ФУНКЦИЯ:
// Вызов ФУНКЦИИ.
Информация := GetTaskInfo (psAppName, psTaskName, pstTaskInfo); // Переменная «Info»
(«Информация») получает возможные ошибки функционирования.

```

## 6.2. Графический дисплей

Графический дисплей, имеющийся в данном изделии, является важным инструментом для управления процессом, поскольку с его помощью можно распознать возможные состояния ошибки, активные компоненты или наличие диагностики. Помимо этого, все данные диагностики, включая данные о модулях ввода/вывода, представляются пользователю через графический дисплей. Более подробная информация об использовании клавиш диагностики и их визуализации приведена в разделе «[Диагностика одним касанием](#)» ([One Touch Diag](#)).

На рисунке, приведенном ниже, показаны символы, имеющиеся на графическом дисплее данного изделия, а также их значения.

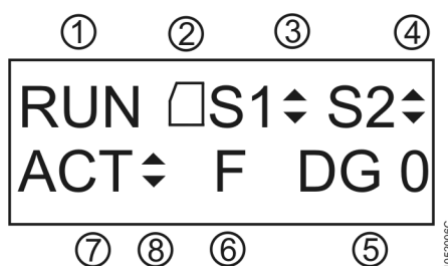


Рисунок 134: Экран состояния центрального процессора (ЦП)

Условные обозначения:

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

- Индикация состояния работы центрального процессора (ЦП). Если приложение центрального процессора (ЦП) работает, то состояние – «Выполнение» (RUN). Если приложение центрального процессора остановлено, то состояние – «Останов» (STOP), а при остановке в метке очистки приложения - BRKP. Более подробная информация приведена в разделе «Состояния работы центрального процессора (ЦП)».
- Индикация наличия карты памяти. Не используется в данной модели центрального процессора (ЦП).
- Индикация трафика COM 1. Стрелка вверх (▲) указывает на передачу данных, а стрелка вниз (▼) - на их прием. Более подробная информация об интерфейсе COM 1 приведена в разделе «Последовательные интерфейсы».
- Индикация трафика COM 2. Не используется в данной модели центрального процессора (ЦП).
- Индикация количества активных диагностик центрального процессора (ЦП). Если отображаемое число отлично от 0 (нуля), то в центральном процессоре присутствует активная диагностика. Подробнее об их визуализации на графическом дисплее центрального процессора с помощью клавиши диагностики см. в разделе «Диагностика одним касанием (One Touch Diag)».
- Форсированные (принудительно заданные) переменные в индикации центрального процессора (ЦП). Если на графическом дисплее отображается символ «F», то это означает, что пользователь принудительно вводит переменную, будь то символическая переменная, переменная прямого представления или переменная директивы AT. Более подробная информация о форсировании переменных приведена в разделе «Запись и форсирование переменных».
- Идентификация состояния резервирования центрального процессора (ЦП). Не используется в данной модели центрального процессора (ЦП).
- Указание на то, что выполняется синхронизация проекта. Не используется в данной модели центрального процессора (ЦП).
  - Помимо описанных выше символов, центральные процессоры (ЦП) Nexto могут выводить на графический дисплей некоторые сообщения, соответствующие выполняемому в данный момент процессу.
  - В таблице, приведенной ниже, представлены сообщения и их соответствующие описания:

Сообщение	Описание
FORMATTING... («Форматирование...»)	Центральный процессор (ЦП) форматирует карту памяти.
FORMATTING ERROR («Ошибка форматирования»)	Произошла ошибка во время форматирования карты памяти центральным процессором (ЦП).
WRONG FORMAT («Неправильный формат»)	Указывает на неправильный формат карты памяти.
INCORRECT PASSWORD («Неправильный пароль»)	Указывает на то, что введенный пароль отличается от настроенного пароля.
TRANSFERRING... («Передача...»)	Указывает на то, что идет перенос проекта.
TRANSFERRING ERROR («Ошибка при передаче»)	Указывает на ошибку при переносе проекта, вызванную проблемами с картой памяти или ее извлечением во время переноса.
TRANSFERRING COMPLETE («Передача завершена»)	Указывает на успешное выполнение переноса.
TRANSFERRING TIMEOUT («Превышение времени ожидания при передаче»)	Указывает на то, что во время переноса проекта произошла блокировка по превышению лимита времени (истекло время обмена данными).
Сообщение	Описание
CPU TYPE MISMATCH («Несоответствие типа центрального процессора»)	Указывает на то, что модель центрального процессора (ЦП) отличается от модели, сконфигурированной в проекте на карте памяти.
VERSION MISMATCH («Несоответствие версий»)	Версия центрального процессора (ЦП) отличается от версии, установленной в проекте на карте памяти.
APPLICATION CORRUPTED («Приложение повреждено»)	Повреждено приложение на карте памяти.

## 6. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ


APPLICATION NOT FOUND («Приложение не найдено»)	Указывает на отсутствие приложения на карте памяти, которое должно быть передано в центральный процессор (ЦП).
CRC NOT FOUND («CRC не найден»)	Указывает на то, что приложение CRC не существует.
MCF FILE NOT FOUND («Файл MCF не найден»)	Указывает на отсутствие файла MCF на карте памяти.
NO TAG («Тег отсутствует»)	В MasterTool IEC XE отсутствует сконфигурированная метка для центрального процессора (ЦП).
NO DESC («Описание отсутствует»)	В MasterTool IEC XE отсутствует сконфигурированное описание центрального процессора (ЦП).
MSG. ERROR («Ошибка сообщения»)	Указывает на наличие ошибки (ошибок) в диагностическом сообщении (сообщениях) запрашиваемого модуля (модулей).
SIGNATURE MISSING («Отсутствует подпись»)	Указывает на то, что с продуктом возникла непредвиденная проблема. Свяжитесь с отделом технической поддержки компании Altus.
APP. ERROR RESTARTING («Перезапуск при ошибке приложения»)	Указывает на то, что в приложении произошла ошибка и время выполнения перезапускает приложение.
APP. NOT LOADED («Приложение не было загружено»)	Указывает на то, что время выполнения не загружает приложение.
LOADING APP. («Загрузка приложения»)	Указывает на то, что время выполнения загрузит приложение.
WRONG SLOT («Неверный слот»)	Указывает на то, что центральный процессор (ЦП) находится в неправильном положении в стойке.
FATAL ERROR («Фатальная ошибка»)	Указывает на наличие серьезных проблем при запуске центрального процессора (ЦП), например, неправильно смонтированные секции центрального процессора (ЦП). Обратитесь в службу поддержки компании Altus.
HW-SW MISMATCH («Несовместимость аппаратного и программного обеспечения»)	Указывает на несовместимость аппаратного и программного обеспечения центрального процессора (ЦП), так как возникла непредвиденная проблема. Обратитесь в службу поддержки компании Altus.
UPDATING FIRMWARE («Обновление микропрограммы (прошивки)»)	Указывает на обновление микропрограммы (прошивки) в центральном процессоре (ЦП).
RECEIVING FIRMWARE («Получение микропрограммы (прошивки)»)	Указывает на то, что файл обновления передается в центральный процессор (ЦП).
UPDATED: («Обновлено:»)	Показывает версию микропрограммы (прошивки), обновленную в ЦП (центральном процессоре).
UPDATE ERROR («Ошибка обновления»)	Указывает на ошибку, возникшую во время обновления микропрограммы (прошивки) центрального процессора (ЦП), вызванную сбоем связи или проблемами конфигурации.
REBOOTING SYSTEM... («Перезагрузка системы...»)	Указывает на то, что центральный процессор (ЦП) перезапускается, чтобы обновление вступило в силу.


Таблица 176: Прочие сообщения графического дисплея

### 6.3. Системный журнал

*Системный журнал (System Log)* представляет собой доступную функцию в устройстве программирования (программаторе) MasterTool IEC XE. Он является важным инструментом для управления технологическими процессами, поскольку позволяет находить события на центральном процессоре (ЦП), которые могут указывать на наличие ошибок, присутствие активных компонентов или активную диагностику. Такие события можно просматривать в хронологическом порядке с разрешением в миллисекунды, при этом во внутренней памяти центрального процессора хранится до тысячи записей журнала, которые нельзя удалить.

Чтобы получить доступ к этим журналам, достаточно перейти в *дерево устройств (Device Tree)* и дважды щелкнуть на «Устройство» (*Device*), затем перейти на вкладку «Журнал» (*Log*), где можно увидеть сотни операций, таких как: максимальные циклы задач, доступ пользователя, изменение режима работы, загрузка и выгрузка приложений, синхронизация приложений между центральными процессорами (ЦП), обновление микропрограммы (прошивки) между другими событиями и действиями.

Для просмотра журналов достаточно подключиться к центральному процессору (выбранный активный путь) и нажать кнопку . Когда данная кнопка нажата, журналы отображаются и обновляются мгновенно. Когда кнопка не нажата, журналы будут оставаться на экране, т.е. данная кнопка имеет две стадии: в одной из них журналы находятся в состоянии обновления, а в другой - в состоянии отключения обновления.

Чтобы прекратить показ журналов, нажмите кнопку .

Журналы можно отфильтровать по 4 типам: предупреждение (предупреждения), ошибка (ошибки), исключение (исключения) и информация.

Другим способом фильтрации сообщений, отображаемых пользователю, является выбор компонента, который необходимо просмотреть.

*Отметка времени* вкладки «Журнал» (*Log*) отображается MasterTool после информации, предоставленной устройством (центральным процессором). MasterTool может отображать отметку времени в локальном времени (время компьютера) или в формате UTC, если отмечено поле для установки флажка «*Время в формате UTC*» (*UTC time*).

**ВНИМАНИЕ:**

Если время устройства или параметр часового пояса неверны, то и отметка времени, отображаемая в MasterTool, будет неверной.

Более подробную информацию о системных журналах можно найти в руководстве пользователя MasterTool IEC XE - MU299609 и в подразделе «Часы реального времени (RTC) и синхронизация времени» данного руководства.

**ВНИМАНИЕ:**

Системные журналы центральных процессоров (ЦП) серии Nexto, начиная с версии микропрограммы (прошивки) 1.4.0.33, перезагружаются в случаях перезапуска центрального процессора (ЦП) либо перезагрузки системы поддержки выполнения программ. Иными словами, при возникновении одной из данных ситуаций можно будет проверить старые журналы.

### 6.4. Отсутствие загрузки приложения при запуске

В случае необходимости пользователь может не загружать существующее приложение в центральный процессор (ЦП) при его запуске. Для этого достаточно включить питание центрального процессора с нажатой кнопкой диагностики и держать ее нажатой до тех пор, пока на экране не появится сообщение «APP. NOT LOADED» («ПРИЛОЖЕНИЕ НЕ ЗАГРУЖЕНО»). При попытке входа в систему программное обеспечение MasterTool IEC XE сообщит, что на центральном процессоре нет приложения. Для повторной загрузки приложения необходимо перезагрузить центральный процессор или загрузить новое приложение.



## 6.5. Распространенные проблемы

Если при включении центрального процессора (ЦП) он не работает, то необходимо проверить следующие элементы:

- Находится ли температура в помещении в пределах поддерживаемого устройством диапазона?
- Подается ли на блок питания стойки правильное напряжение?
- Установлен ли модуль блока питания крайним слева в стойке (если смотреть на стойку спереди), а затем центральный процессор серии Nexto?
- Подано ли питание на сетевые устройства, такие как концентраторы, коммутаторы или маршрутизаторы? Они подключены, настроены и работают должным образом?
- Правильно ли подключен сетевой кабель Ethernet к порту NET 1 или NET 2 центрального процессора Nexto и к сетевому устройству?
- Включен ли центральный процессор серии Nexto, находится ли он в режиме «Выполнение» (Run) и нет ли диагностики, связанной с аппаратным обеспечением?

Если центральный процессор (ЦП) Nexto указывает режим «Выполнение» (Run), но не отвечает на запрошенные коммуникации, будь то через MasterTool IEC XE или протоколы, необходимо проверить следующие элементы:

- Правильно ли настроены параметры Ethernet в центральном процессоре (ЦП)?
- Правильно ли сконфигурирован соответствующий протокол связи в центральном процессоре (ЦП)?
- Правильно ли включены переменные, обеспечивающие связь с MODBUS?

Если проблема не выявлена, обратитесь в службу технической поддержки компании Altus.

## 6.6. Поиск и устранение неисправностей

В таблице ниже приведены симптомы некоторых проблем с указанием их возможных причин и способов решения. Если проблема сохраняется, обратитесь в службу технической поддержки компании Altus.

Симптом	Возможная причина	Решение
Не включается питание	Отсутствие электропитания или неправильное электропитание	Проверьте правильность подключения центрального процессора (ЦП) в стойке. Отключите питание и снимите с шины все модули, за исключением блока питания и центрального процессора. Включите питание шины и проверьте работу источников питания, внешнего и находящегося в стойке. Проверьте, поступает ли напряжение питания на контакты блока питания Nexto и правильно ли они поляризованы.
На экране центрального процессора (ЦП) появляется сообщение WRONG CPU SLOT («НЕПРАВИЛЬНЫЙ СЛОТ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА (ЦП)»)	Центральный процессор (ЦП) находится в неправильном положении	Центральный процессор (ЦП) должен быть помещен в слоты 0 и 1 стойки 0. Установите его в правильное положение, выключите и снова включите..
	Центральный процессор (ЦП) или стойка повреждены	Замените центральный процессор (ЦП) или стойку-место и проверьте, сохраняется ли проблема.
Не осуществляет передачу данных	Плохой контакт или неправильная конфигурация	Проверьте все соединения кабелей связи. Проверьте конфигурацию последовательного интерфейса и интерфейса сети Ethernet в программном обеспечении MasterTool IEC XE.

Таблица 177: Поиск и устранение неисправностей

### 6.7. Планово-предупредительное техническое обслуживание

- Необходимо ежегодно проверять надежность соединения соединительных кабелей, отсутствие скопления пыли, в частности защитных устройств.
- В средах, подверженных чрезмерному загрязнению, оборудование необходимо периодически очищать от пыли, мусора и т.п.
- TVS-диоды, используемые для защиты от переходных процессов, вызванных атмосферными разрядами, должны периодически проверяться, поскольку они могут быть повреждены или разрушены в случае превышения предельной поглощенной энергии. Во многих случаях отказ может быть невидимым. В критических приложениях рекомендуется периодически заменять TVS-диоды, даже если они не демонстрируют видимых признаков отказа.
- Проверяйте герметичность и чистоту шины каждые шесть месяцев.
- Дополнительную информацию см. в руководстве по серии Nexto - MU214600.