



Руководство пользователя
Серия Nexto CPU
NX3030

MU214615 Ред. С

28 Сентября 2022

Никакая часть данного документа не может быть скопирована или воспроизведена в какой-либо 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, отличается прочностью и надежностью благодаря строгому контролю качества, которому оно подвергается. Однако любое электронное промышленное оборудование (программируемые контроллеры, числовые команды и т.д.) при наличии дефектных компонентов и/или при ошибке программирования или монтажа может нанести ущерб машинам или управляемым ими процессам. Это может даже поставить под угрозу жизнь людей. Пользователь должен учитывать возможные последствия дефектов и в целях безопасности предусмотреть дополнительные внешние установки. Этой проблеме уделяется больше внимания при первоначальном вводе в эксплуатацию и испытаниях.

Оборудование, выпускаемое компанией Altus, не подвергает окружающую среду непосредственной опасности, поскольку в процессе эксплуатации не выделяет загрязняющих веществ. Однако, говоря об утилизации оборудования, необходимо отметить, что встроенная электроника при неправильной утилизации может содержать вредные для природы материалы. Поэтому при утилизации такого рода изделий рекомендуется направлять их на перерабатывающие предприятия, которые гарантируют правильную утилизацию отходов.

Перед установкой и использованием изделия необходимо ознакомиться с документацией на него, например, с руководством по эксплуатации и техническими характеристиками. Примеры и рисунки, представленные в данном документе, носят исключительно иллюстративный характер. В связи с возможными модернизациями и усовершенствованиями продуктов компания Altus не несет ответственности за использование этих примеров и рисунков в реальных приложениях. Они должны использоваться только для обучения пользователей и расширения их опыта работы с продуктами и их возможностями.

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

Altus гарантирует работу своего оборудования в соответствии с четкими инструкциями, содержащимися в его руководствах и/или технических характеристиках, не гарантируя успешного применения оборудования в каком-либо конкретном виде.

Altus не признает никаких других гарантий, прямых или подразумеваемых, в основном, когда конечные клиенты имеют дело с поставщиками третьих лиц. Запросы на получение дополнительной информации о поставке, характеристиках оборудования и/или любых других услугах Altus должны быть сделаны в письменной форме. Компания Altus не несет ответственности за предоставление информации о своем оборудовании без официального запроса. В этих изделиях может использоваться технология EtherCAT® (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 О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ С ОТКРЫТЫМ ИСХОДНЫМ КОДОМ

Для получения исходного кода под лицензиями GPL, LGPL, MPL и другими открытыми лицензиями, содержащегося в данном продукте, обращайтесь по адресу opensource@altus.com.br. Помимо исходного кода, по запросу могут быть предоставлены все упомянутые лицензионные условия, отказы от гарантий и уведомления об авторских правах.

Содержимое

1.	Введение	1
1.1.	Серия Nexto	1
1.2.	Инновационные возможности	2
1.3.	Документы, относящиеся к данному Руководству	3
1.4.	Визуальный осмотр	4
1.5.	Техническая поддержка	4
1.6.	Предупреждающие сообщения, используемые в данном руководстве	5
2.	Техническая характеристика	6
2.1.	Панели и соединения	6
2.2.	Общие характеристики	8
2.2.1.	Общие характеристики	8
2.2.2.	Память	10
2.2.3.	Протоколы	12
2.2.4.	Последовательные интерфейсы	12
2.2.4.1.	COM 1	12
2.2.4.2.	COM 2	13
2.2.5.	Интерфейсы Ethernet	13
2.2.5.1.	NET 1	13
2.2.5.2.	NET 2	14
2.2.6.	Интерфейс карты памяти	14
2.2.7.	Характеристики окружающей среды	15
2.3.	Совместимость с другими продуктами	15
2.4.	Производительность	15
2.4.1.	Сроки применения	16
2.4.2.	Время выполнения инструкций	16
2.4.3.	Времена инициализации	16
2.4.4.	Интервальное время	16
2.5.	Физические размеры	16
2.6.	Данные о закупках	17
2.6.1.	Входящие в комплект поставки предметы	17
2.6.2.	Код продукта	17

СОДЕРЖИМОЕ

2.7.	Сопутствующие товары.....	18
3.	Установка	20
3.1.	Механическая установка	20
3.2.	Электромонтаж	20
3.3.	Подключение к сети Ethernet.....	21
3.3.1.	IP Адрес.....	21
3.3.2.	Безвозмездный ARP.....	22
3.3.3.	Прокладка сетевого кабеля	22
3.4.	Последовательное сетевое соединение RS-232.....	23
3.4.1.	Связь по интерфейсу RS-232C.....	23
3.5.	Последовательное сетевое соединение RS-485/422	23
3.5.1.	Связь по RS-485 без оконечного устройства.....	24
3.5.2.	Связь по RS-485 с внутренней заделкой.....	25
3.5.3.	Связь по RS-485 с внешней клеммой	26
3.5.4.	Пример подключения сети RS-485 с внешней заделкой и резервированием ведущего устройства.....	26
3.5.5.	Связь по RS-422 без оконечного устройства	27
3.5.6.	Связь по RS-422 с внутренней заделкой.....	28
3.5.7.	Связь по RS-422 с внешней заделкой	29
3.5.8.	Пример сети RS-422.....	29
3.6.	Установка карты памяти.....	30
3.7.	Установка структуры.....	31
3.7.1.	Установка модулей в стойку основной объединительной панели ...	31
3.8.	Установка программирующего устройства.....	31
4.	Начальное программирование.....	32
4.1.	Организация памяти и доступ к ней.....	32
4.2.	Профили проектов.....	34
4.2.1.	Одиночный	34
4.2.2.	Базовый	35
4.2.3.	Нормальный	35
4.2.4.	Экспертный.....	35
4.2.5.	Пользовательский	36
4.2.6.	Профиль машины.....	36
4.2.7.	Общая таблица.....	37
4.2.8.	Максимальное количество задач.....	38
4.3.	Конфигурация процессора	38
4.4.	Библиотеки.....	39
4.5.	Вставка экземпляра протокола.....	40
4.5.1.	MODBUS Ethernet	40
4.6.	Поиск устройства.....	41
4.7.	Вход в систему.....	43
4.8.	Режим работы	45

СОДЕРЖИМОЕ

4.9.	Режим остановки.....	46
4.10.	Запись и принудительное использование переменных.....	46
4.11.	Выход из системы.....	47
4.12.	Загрузка проекта.....	48
4.13.	Рабочие состояния процессора.....	49
4.13.1.	Работа.....	49
4.13.2.	Остановка.....	49
4.13.3.	Точка останова.....	49
4.13.4.	Исключение.....	50
4.13.5.	Сброс тепла.....	50
4.13.6.	Сброс холода.....	50
4.13.7.	Сброс исходных данных.....	50
4.13.8.	Команда сброса процесса (IEC 60870-5-104).....	50
4.14.	Программный компонент (POU) и глобальные списки переменных (GVL).....	50
4.14.1.	Программа MainPrg.....	50
4.14.2.	Программа StartPrg.....	51
4.14.3.	Программа UserPrg.....	51
4.14.4.	GVL System_Diagnostics.....	51
4.14.5.	GVL Disables.....	52
4.14.6.	GVL IOQualities.....	53
4.14.7.	GVL Module_Diagnostics.....	53
4.14.8.	GVL Qualities.....	54
4.14.9.	GVL ReqDiagnostics.....	56
4.14.10.	Prepare_Start Function.....	57
4.14.11.	Prepare_Stop Function.....	58
4.14.12.	Start_Done Function.....	58
4.14.13.	Stop_Done Function.....	58
5.	Конфигурация.....	59
5.1.	CPU Конфигурация.....	59
5.1.1.	Общие параметры.....	59
5.1.1.1.	Горячая замена.....	61
5.1.1.1.1.	Горячая замена отключена, только для заявленных модулей.....	62
5.1.1.1.2.	Горячая замена отключена.....	62
5.1.1.1.3.	Отключение горячей замены без согласованности запуска.....	62
5.1.1.1.4.	Горячая замена включена, согласованность запуска только для заявленных модулей.....	62
5.1.1.1.5.	Горячая замена с поддержкой согласованности при запуске.....	62
5.1.1.1.6.	Включение горячей замены без согласованности запуска.....	63
5.1.1.1.7.	Как выполнить горячую замену.....	63
5.1.1.2.	Сохраняемые и постоянные области памяти.....	65
5.1.1.3.	Конфигурации ТСП.....	66
5.1.1.4.	Параметры проекта.....	68
5.1.2.	Конфигурация внешних событий.....	68

СОДЕРЖИМОЕ

5.1.3.	Конфигурация SOE	70
5.1.4.	Синхронизация времени.....	72
5.1.4.1.	IEC 60870-5-104	73
5.1.4.2.	SNTP	74
5.1.4.3.	Переход на летнее время (DST).....	74
5.1.5.	Внутренние точки.....	75
5.1.5.1.	Качественные преобразования.....	76
5.1.5.1.1.	Внутреннее качество.....	77
5.1.5.1.2.	Преобразование IEC 60870-5-104.....	78
5.1.5.1.3.	Внутреннее качество MODBUS.....	79
5.1.5.1.4.	Качество модулей ввода/вывода локальной шины.....	80
5.1.5.1.5.	Качество модулей ввода/вывода PROFIBUS.....	81
5.1.5.1.6.	Качество цифровых входов PROFIBUS.....	81
5.1.5.1.7.	Качество цифрового выхода PROFIBUS.....	82
5.1.5.1.8.	Качество аналоговых входов PROFIBUS.....	83
5.1.5.1.9.	Качество аналогового выхода PROFIBUS.....	84
5.2.	Конфигурация последовательных интерфейсов	86
5.2.1.	COM 1	86
5.2.1.1.	Расширенные конфигурации.....	88
5.2.2.	COM 2	89
5.2.2.1.	Расширенные конфигурации.....	89
5.3.	Конфигурация интерфейсов Ethernet.....	90
5.3.1.	Локальные интерфейсы Ethernet	90
5.3.1.1.	NET 1	90
5.3.1.2.	NET 2	91
5.3.2.	Удаленные интерфейсы Ethernet.....	91
5.3.2.1.	NET 1	91
5.3.3.	Зарезервированные TCP/UDP-порты.....	91
5.4.	Конфигурация модуля NX5000.....	92
5.5.	Конфигурация протоколов.....	94
5.5.1.	Поведение протокола x Состояние процессора.....	96
5.5.2.	Двойные пункты.....	97
5.5.3.	Очередь событий процессора	97
5.5.3.1.	Потребители	98
5.5.3.2.	Принципы функционирования очередей.....	98
5.5.3.2.1.	Знак переполнения.....	99
5.5.3.3.	Производители	99
5.5.4.	Перехват команд, поступающих из центра управления.....	99

5.5.5.	Ведущее устройство MODBUS RTU.....	104
5.5.5.1.	Конфигурация протокола MODBUS Master путем символьного отображения.....	104
5.5.5.1.1.	Общие параметры протокола MODBUS Master - Конфигурация символьного отображения.....	104
5.5.5.1.2.	Конфигурация устройств - конфигурация символьного отображения.....	107
5.5.5.1.3.	Конфигурация отображений - Настройки символьных отображений.....	108
5.5.5.1.4.	Конфигурация запросов - Настройки символьного отображения.....	109
5.5.5.2.	Конфигурация протокола MODBUS Master для прямого представления (%Q).....	114
5.5.5.2.1.	Общие параметры протокола MODBUS Master - настройка через прямое представление (%Q).....	114
5.5.5.2.2.	Конфигурация устройств - Конфигурация для прямого представления (%Q).....	115
5.5.5.2.3.	Конфигурация отображений - Конфигурация для прямого представления (%Q).....	116
5.5.6.	MODBUS RTU Slave.....	118
5.5.6.1.	Конфигурирование протокола MODBUS Slave через символьное отображение.....	118
5.5.6.1.1.	Общие параметры протокола MODBUS Slave - конфигурирование через символьное отображение.....	119
5.5.6.1.2.	Конфигурация отношений - настройка символьного отображения.....	123
5.5.6.2.	Конфигурирование протокола MODBUS Slave через прямое представление (%Q).....	124
5.5.6.2.1.	Общие параметры протокола MODBUS Slave - конфигурирование через прямое представление (%Q).....	124
5.5.6.2.2.	Конфигурирование отображений - Конфигурирование через прямое представление (%Q).....	125
5.5.7.	MODBUS Ethernet.....	127
5.5.8.	Клиент MODBUS Ethernet.....	129
5.5.8.1.	Конфигурирование клиента MODBUS Ethernet с помощью символьного отображения.....	129
5.5.8.1.1.	Общие параметры клиентского протокола MODBUS – конфигурирование через символьное отображение.....	130
5.5.8.1.2.	Конфигурация устройства - Конфигурация через символьное отображение.....	131
5.5.8.1.3.	Конфигурация отображений - Конфигурация через символическое отображение.....	133
5.5.8.1.4.	Конфигурация запросов - Конфигурация через символьное отображение.....	135
5.5.8.1.5.	Конфигурирование клиента MODBUS Ethernet через прямое представление (%Q).....	139
5.5.8.2.	Общие параметры клиента протокола MODBUS - конфигурация для прямого представления (%Q).....	139

5.5.8.2.2.	Конфигурация устройства - Конфигурация через прямое представление (%Q) . . .	140
5.5.8.2.3.	Конфигурация отображения - конфигурация через прямое представление (%Q) . .	141
5.5.8.3.	Начало клиентской связи MODBUS в ациклической форме	143
5.5.9.	Сервер MODBUS Ethernet	143
5.5.9.1.	Конфигурация протокола Ethernet сервера MODBUS для символьного отображения.	143
5.5.9.1.1.	Общие параметры протокола сервера MODBUS - конфигурирование через символическое отображение.	143
5.5.9.1.2.	Диагностика сервера MODBUS - Конфигурация через символьное отображение. . .	145
5.5.9.1.3.	Конфигурация отображения - конфигурация через символическое отображение.	147
5.5.9.2.	Конфигурация протокола Ethernet сервера MODBUS через прямое представление (%Q) 148	
5.5.9.2.1.	Общие параметры протокола MODBUS Server - конфигурирование через прямое представление (%Q)	149
5.5.9.2.2.	Конфигурация отображения - конфигурация через прямое представление (%Q) . .	150
5.5.10.	Сервер OPC DA.	152
5.5.10.1.	Создание проекта для связи с OPC DA.	154
5.5.10.2.	Конфигурирование ПЛК на сервере OPC DA.	157
5.5.10.2.1.	Импорт конфигурации проекта	159
5.5.10.3.	Конфигурация с ПЛК на сервере OPC DA с резервированием соединений. 159	
5.5.10.4.	Переменные состояния и качества связи OPC DA	160
5.5.10.5.	Пределы связи с OPC DA Сервера.	162
5.5.10.6.	Доступ к данным через клиент OPC DA.	162
5.5.11.	Сервер OPC UA.	164
5.5.11.1.	Создание проекта для связи с OPC UA.	165
5.5.11.2.	Типы поддерживаемых переменных.	167
5.5.11.3.	Ограничение количества подключенных клиентов на сервере OPC UA.	167
5.5.11.4.	Ограничение переменных связи на сервере OPC UA	167
5.5.11.5.	Настройки шифрования.	167
5.5.11.6.	Основные параметры связи, настраиваемые в клиенте OPC UA.	168
5.5.11.6.1.	URL конечной точки.	168
5.5.11.6.2.	Интервал публикации (мс) Интервал выборки (мс).	169
5.5.11.6.3.	Счетчик срока службы и Счетчик времени ожидания	169
5.5.11.6.4.	Размер очереди и старая очередь	169
5.5.11.6.5.	Тип фильтра и тип мертвой зоны.	169
5.5.11.6.6.	Публикация включена, максимальное количество уведомлений на публикацию приоритетно	170
5.5.11.7.	Доступ к данным через клиента OPC UA.	170
5.5.12.	EtherCAT Master	172
5.5.12.1.	Установка и вставка EtherCAT-устройств.	172

5.5.12.1.1.	Сканирование устройств.....	173
5.5.12.2.	Диагностические переменные.....	174
5.5.12.3.	Настройки EtherCAT-ведущего.....	177
5.5.12.3.1.	Основные параметры.....	177
5.5.12.3.2.	EtherCAT Master - отображение входов/выходов.....	179
5.5.12.3.3.	Вкладки "Состояние" и "Информация".....	180
5.5.12.4.	Конфигурация ведомого устройства EtherCAT.....	180
5.5.12.4.1.	Параметры ведомого.....	180
5.5.12.4.2.	FMMU/Sync.....	184
5.5.12.4.3.	FMMU/Sync - FMMU.....	184
5.5.12.4.4.	FMMU/Sync - Sync Manager.....	184
5.5.12.4.5.	Данные процесса и экспертные данные процесса.....	185
5.5.12.4.6.	Данные процесса и экспертные данные процесса - редактирование списка PDO.....	187
5.5.12.4.7.	Данные процесса и экспертные данные процесса - определение содержания PDO.....	187
5.5.12.4.8.	Параметры ввода в эксплуатацию.....	188
5.5.12.4.9.	Онлайн.....	188
5.5.12.4.10.	EtherCAT Slave - отображение входов/выходов.....	189
5.5.12.4.11.	Вкладки "Состояние" и "Информация".....	189
5.5.13.	EtherNet/IP.....	189
5.5.13.1.	EtherNet/IP.....	190
5.5.13.2.	Конфигурация сканера EtherNet/IP.....	191
5.5.13.2.1.	Общие сведения.....	192
5.5.13.2.2.	Соединения.....	192
5.5.13.2.3.	Сборки.....	194
5.5.13.2.4.	Сопоставление входов/выходов EtherNet/IP.....	195
5.5.13.3.	Конфигурация EtherNet/IP-адаптера.....	195
5.5.13.3.1.	Типы модулей.....	196
5.5.13.3.2.	EtherNet/IP Module I/O Mapping.....	196
5.5.14.	Сервер IEC 60870-5-104.....	196
5.5.14.1.	Тип данных.....	197
5.5.14.2.	Двойные точки.....	198
5.5.14.2.1.	Двойные точки цифрового входа.....	199
5.5.14.2.2.	Двойные точки цифрового выхода.....	200
5.5.14.3.	Общие параметры.....	205
5.5.14.4.	Сопоставление данных.....	205
5.5.14.5.	Связующий уровень.....	207
5.5.14.6.	Прикладной уровень.....	209
5.5.14.7.	Диагностика сервера.....	211
5.5.14.8.	Квалификатор команд.....	212
5.6.	Коммуникационные показатели.....	213
5.6.1.	Сервер MODBUS.....	213
5.6.1.1.	Локальные интерфейсы процессора.....	213
5.6.1.2.	Удаленные интерфейсы.....	214

СОДЕРЖИМОЕ

5.6.2.	Сервер OPC DA.....	214
5.6.3.	Сервер OPC UA.....	215
5.6.4.	Сервер IEC60870-5-104.....	215
5.7.	Производительность системы.....	216
5.7.1.	Время сканирования ввода/вывода.....	216
5.7.2.	Карта памяти.....	216
5.8.	Часы RTC.....	217
5.8.1.	Функциональные блоки для чтения и записи RTC.....	217
5.8.1.1.	Функциональные блоки для чтения RTC.....	217
5.8.1.1.1.	Получение даты и времени.....	218
5.8.1.1.2.	Получение часового пояса.....	218
5.8.1.1.3.	Получение дня недели.....	219
5.8.1.2.	Функции записи в RTC.....	220
5.8.1.2.1.	Установка даты и времени.....	220
5.8.1.2.2.	Установка часового пояса.....	221
5.8.2.	Структуры данных RTC.....	222
5.8.2.1.	РАСШИРЕННАЯ_ДАТА_И_ВРЕМЯ.....	223
5.8.2.2.	ДНИ_НЕДЕЛИ.....	223
5.8.2.3.	RTC_СТАТУС.....	223
5.8.2.4.	НАСТРОЙКИ_ЧАСОВОГО_ПОЯСА.....	224
5.9.	Память пользовательских файлов.....	224
5.10.	Карта памяти.....	226
5.10.1.	Подготовка проекта.....	226
5.10.2.	Перенос проекта.....	227
5.10.3.	Доступ к MasterTool.....	228
5.11.	Информационное и конфигурационное меню процессора.....	228
5.12.	Функциональные блоки и функции.....	231
5.12.1.	Специальные функциональные блоки для последовательных интерфейсов.....	231
5.12.1.1.	SERIAL_CFG.....	235
5.12.1.2.	SERIAL_GET_CFG.....	237
5.12.1.3.	SERIAL_GET_CTRL.....	238
5.12.1.4.	SERIAL_GET_RX_QUEUE_STATUS.....	240
5.12.1.5.	SERIAL_PURGE_RX_QUEUE.....	241
5.12.1.6.	SERIAL_RX.....	243
5.12.1.7.	SERIAL_RX_EXTENDED.....	245
5.12.1.8.	SERIAL_SET_CTRL.....	247
5.12.1.9.	SERIAL_TX.....	249
5.12.2.	Обновление входов и выходов.....	251
5.12.2.1.	REFRESH_ВХОД.....	251
5.12.2.2.	REFRESH_ВЫХОД.....	252
5.12.3.	Функциональный блок ПИД-регулятора.....	254
5.12.4.	Сохранение таймера.....	254
5.12.4.1.	TOF_RET.....	254
5.12.4.2.	TON_RET.....	255

5.12.4.3.	TP_RET	257
5.12.5.	Таймер без резервирования.....	258
5.12.5.1.	TOF_NR	258
5.12.5.2.	TON_NR	259
5.12.5.3.	TP_NR.....	259
5.12.6.	Журнал пользователя.....	260
5.12.6.1.	Добавление журнала пользователя.....	261
5.12.6.2.	Журнал пользователя Удалить все.....	262
5.12.7.	Очистить диагностику Rtu.....	263
5.12.8.	Очистить очередь событий	263
5.13.	SNMP.....	264
5.13.1.	Введение.....	264
5.13.2.	SNMP nas UCPs Nexto	264
5.13.3.	Частная MIB.....	265
5.13.4.	Конфигурация SNMP	265
5.13.5.	Сообщества пользователей и SNMP	266
5.14.	Управление пользователями и правами доступа.....	267
6.	Резервирование с помощью процессора NX3030	268
6.1.	Введение	268
6.2.	Техническое описание и конфигурация.....	269
6.2.1.	Минимальная конфигурация резервного процессора (без использования панели PX2612)	269
6.2.2.	Типовые конфигурации резервного процессора.....	270
6.2.2.1.	Добавление модулей NX5001 для сетей PROFIBUS.....	271
6.2.2.2.	Добавление модулей NX5000 для сетей Ethernet.....	271
6.2.3.	NX4010 Модуль	271
6.2.3.1.	Характеристики NX4010.....	272
6.2.4.	Панель управления резервированием PX2612.....	272
6.2.4.1.	Особенности PX2612	274
6.2.5.	Соединения между полукластерами и панелью управления резервированием PX.....	274
6.2.6.	Общие характеристики резервного CP.....	275
6.2.7.	Данные о закупках.....	278
6.3.	Принципы работы	279
6.3.1.	Идентификация процессора NX3030.....	279
6.3.2.	Единый резервный проект.....	279
6.3.3.	Избыточная структура проекта.....	279
6.3.3.1.	Шаблон резервирования.....	279
6.3.3.2.	Однократная и циклическая задача MainTask.....	279
6.3.3.3.	Программа MainPrg.....	279
6.3.3.4.	Программа ActivePrg	280
6.3.3.5.	Программа NonSkippedPrg.....	280
6.3.3.6.	Избыточные и нередуцированные переменные.....	281
6.3.3.7.	Избыточные и нередуцированные переменные %I.....	281
6.3.3.8.	Избыточные и нередуцированные переменные %Q	281
6.3.3.9.	Избыточные и нередуцированные переменные %M.....	282

6.3.3.10.	Избыточные и нередуцированные символьные переменные	283
6.3.4.	Множественное отображение	283
6.3.5.	Диагностика, команды и структура данных пользователя	284
6.3.6.	Услуги циклической синхронизации через NETA и NETB	285
6.3.6.1.	Диагностика и обмен командами	285
6.3.6.2.	Синхронизация избыточных данных	285
6.3.6.3.	Синхронизация резервных списков принуждения	286
6.3.7.	Услуги спорадической синхронизации через NETA и NETB	286
6.3.7.1.	Синхронизация проектов	286
6.3.8.	Отключение синхронизации проектов	287
6.3.9.	Конфигурация сети PROFIBUS	288
6.3.9.1.	Резервирование PROFIBUS	288
6.3.9.2.	Режимы отказов PROFIBUS важные и неважные	288
6.3.10.	Резервирование сетей Ethernet с объединением сетевых карт	288
6.3.11.	Методы изменения IP	289
6.3.11.1.	Исправленный IP	289
6.3.11.2.	IP-адрес обмена	289
6.3.11.3.	Активный IP	290
6.3.11.4.	Несколько IP	291
6.3.12.	Объединение сетевых карт и комбинированное использование активного IP	292
6.3.13.	Использование интерфейсов Ethernet с индикацией жизненно важных неисправностей	292
6.3.13.1.	Сбой в интерфейсе Ethernet	292
6.3.13.2.	Сбой в работе подключенного сервера MODBUS	292
6.3.14.	Использование связи OPC DA с резервными проектами	292
6.3.15.	Состояния резервного процессора	293
6.3.15.1.	Ненастроенное состояние	293
6.3.15.2.	Начальное состояние	294
6.3.15.3.	Активное состояние	294
6.3.15.4.	Состояние ожидания	294
6.3.15.5.	Неактивное состояние	294
6.3.16.	Функции командной панели резервирования PX2612	295
6.3.16.1.	Кнопки PX2612	295
6.3.16.2.	Светодиоды PX2612	296
6.3.16.3.	Реле PX2612	296
6.3.17.	Переход между состояниями резервирования	296
6.3.17.1.	Переход 1 - из состояния "не настроен" в состояние "запущен"	297
6.3.17.2.	Переход 2 - Запуск в неконфигурированном состоянии	298
6.3.17.3.	Переход 3 - начало работы в неактивное состояние	298
6.3.17.4.	Переход 4 - Начало активной деятельности	298
6.3.17.5.	Переход 5 - Переход в режим ожидания	298
6.3.17.6.	Переход 6 - Неактивный в неконфигурированный	298
6.3.17.7.	Переход 7 - активный в неконфигурированный	298
6.3.17.8.	Переход 8 - активный в неактивный	299

6.3.17.9.	Переход 9 - активный режим в резервный.	299
6.3.17.10.	Переход 10 - из режима ожидания в режим неконфигурирования.	299
6.3.17.11.	Переход 11 - из режима ожидания в режим неактивности	299
6.3.17.12.	Переход 12 - из режима ожидания в активный.	299
6.3.18.	Первые моменты в активном состоянии.	299
6.3.19.	Распространенные отказы, вызывающие автоматическое переключение между полукластерами.	300
6.3.20.	Сбои, связанные с переключением между полукластерами, управляемыми пользователем.	300
6.3.21.	Отказоустойчивость	301
6.3.21.1.	Простой отказ с недоступностью.	302
6.3.21.2.	Простой отказ без недоступности, вызывающий переключение	302
6.3.21.3.	Двойной отказ без недоступности, вызывающий переключение	302
6.3.22.	Накладные расходы на резервирование.	303
6.4.	Программирование резервных процессоров	303
6.4.1.	Мастер создания нового резервного проекта.	303
6.4.2.	Конфигурация полукластеров.	308
6.4.2.1.	Фиксированная конфигурация в стойках от 0 до 5 позиций.	308
6.4.3.	Конфигурация портов Ethernet в процессоре NX3030 (NET 1 и NET 2).	308
6.4.3.1.	Конфигурация IP-адреса.	308
6.4.3.2.	Объединение сетевых карт между NET 1 и NET 2	309
6.4.3.3.	Настройка жизненно важных отказов в NET 1 и NET 2.	309
6.4.4.	Конфигурация модулей NX5001	310
6.4.4.1.	Установка или удаление модулей NX5001.	310
6.4.4.2.	Настройка параметров модулей NX5001.	310
6.4.4.3.	Конфигурация удаленных устройств PROFIBUS	311
6.4.5.	Конфигурация модулей NX5000	312
6.4.5.1.	Вставка или извлечение модулей NX5000	312
6.4.5.2.	Конфигурация модулей NX5000	312
6.4.5.3.	Группировка модулей NX5000 с резервированием сетевых карт	312
6.4.5.3.1.	Настройка жизнеспособности отказа.	312
6.4.6.	Конфигурация резервирования NX4010	312
6.4.7.	Конфигурация драйверов ввода/вывода.	313
6.4.8.	Конфигурация основной задачи.	313
6.4.8.1.	Программа ActivePrg	314
6.4.8.2.	Программа NonSkippedPrg.	314
6.4.9.	Объект конфигурации резервирования	315
6.4.10.	Модуль_диагностики GVL.	315
6.4.11.	GVL с избыточными символьными переменными.	315
6.4.12.	POU из типа программы с избыточными символьными переменными	316
6.4.13.	Использование точек останова в резервированных системах.	316
6.4.14.	Управление экземплярами MODBUS в резервированной системе.	316
6.4.15.	Ограничения на программирование резервного ПЛК	317
6.4.15.1.	Ограничения в резервированных GVL и POU	317
6.4.15.2.	Ограничения нередуцируемой программы (NonSkippedPrg)	317

6.4.16.	Получение состояния резервирования полукластера	317
6.4.17.	Чтение нерезервированных диагностических данных.....	317
6.5.	Загрузка программ резервного процессора.....	318
6.5.1.	Первоначальная загрузка резервного проекта	318
6.5.1.1.	Первый этап - обнаружение IP-адреса для подключения MasterTool.....	318
6.5.1.2.	Второй этап - проверка конфликта IP-адресов.....	319
6.5.1.3.	Третий этап - подготовка подключения MasterTool (установка активного пути).....	319
6.5.1.4.	Четвертый этап - идентификация процессора NX3030 и проверка дисплея процессора.....	319
6.5.1.5.	Пятый этап - Выгрузка резервных проектов	320
6.5.2.	Подключение MasterTool к процессору NX3030 от резервного ПЛК.....	321
6.5.3.	Загрузка модификаций в резервном проекте	321
6.5.4.	Оффлайн и онлайн загрузка модификации.....	321
6.5.4.1.	Модификации, требующие автономной загрузки и прерывания управления процессом.....	322
6.5.4.2.	Модификации, требующие автономной загрузки.....	322
6.5.4.3.	Модификации, разрешающие загрузку в режиме онлайн.....	322
6.5.5.	Онлайн-загрузка модификаций.....	323
6.5.6.	Автономная загрузка модификаций с прерыванием управления процессом.....	323
6.5.7.	Предыдущее Планирование модификаций в автономном режиме без прерывания управления процессом	324
6.5.7.1.	Предварительное планирование горячих модификаций в резервированных сетях PROFIBUS	324
6.5.7.1.1.	Шаг 1 - Планирование будущего расширения пультов, включенных в сеть PROFIBUS Первоначальная версия.....	324
6.5.7.1.2.	Шаг 2 - Вставка начальной версии резервированной сети PROFIBUS в проект.....	325
6.5.7.1.3.	Шаг 3 - Выделение областей переменных %I и %Q для сети PROFIBUS с учетом будущего удаленного расширения.....	325
6.5.7.2.	Предыдущее планирование других горячих модификаций	326
6.5.7.3.	Несовместимость приложений.....	327
6.5.7.4.	Обновление проекта в связи с обновлением MasterTool IEC XE.....	327
6.5.7.4.1.	Обновление проекта с версий, предшествующих 2.00, до версии 2.00 или выше.....	327
6.5.8.	Исследование резервирования для автономной загрузки модификаций без прерывания управления процессом.....	328
6.5.8.1.	Шаг 1 - Проверка основных требований присутствующих.....	328
6.5.8.2.	Шаг 2 - Не загружать в группу модификации, которые можно загрузить в режиме онлайн	329
6.5.8.3.	Шаг 3 - Резервное копирование предыдущего проекта.....	329
6.5.8.4.	Шаг 4. Редактирование загруженных в автономном режиме модификаций	329
6.5.8.5.	Шаг 5 - Отключение синхронизма неактивного проекта ПЛК.....	329
6.5.8.6.	Шаг 6 - Выполнение физических модификаций.....	330
6.5.8.7.	Шаг 7 - Загрузка автономных модификаций в неактивный ПЛК.....	330
6.5.8.8.	Шаг 8 - Переведите неактивный ПЛК обратно в режим работы, чтобы он перешел в состояние ожидания.....	330

6.5.8.9.	Шаг 9 - Выполнение переключения между активным и резервным ПЛК.....	330
6.5.8.10.	Шаг 10 - Включение синхронизма проектов в активном ПЛК	331
6.5.8.11.	Шаг 11 - Необязательная реорганизация сетей ПЛК и PROFIBUS в активном состоянии	331
6.6.	Обслуживание резервирования.....	331
6.6.1.	Горячая замена модулей в резервном ПЛК.....	331
6.6.2.	Предупреждающие сообщения MasterTool	331
6.6.2.1.	Блокировка загрузки резервного или нерезервного проекта.....	331
6.6.2.2.	Предупреждения перед командами, которые могут остановить активный ПЛК....	331
6.6.2.3.	Предупреждение перед входом в неактивный CP.....	332
6.6.3.	Диагностика резервирования на графическом дисплее процессора NX3030.....	332
6.6.3.1.	Состояние резервирования CP	332
6.6.3.2.	Экраны, расположенные ниже меню РЕЗЕРВ.....	332
6.6.4.	Структура диагностики избыточности	332
6.6.4.1.	Диагностика избыточности	333
6.6.4.2.	Команды резервирования	344
6.6.4.3.	Обмен пользовательской информацией между PLCA и PLCB.....	347
6.6.4.4.	Диагностика Modbus, используемая при резервировании	347
6.6.4.5.	Журнал событий резервирования.....	348
6.6.5.	Панельный тест PX2612.....	348
6.6.5.1.	Вход в тестовый режим.....	348
6.6.5.2.	Ручной и автоматический выходы тестового режима.....	348
6.6.5.3.	Тестирование светодиодов.....	348
6.6.5.4.	Тест кнопок.....	349
6.6.5.5.	Испытание реле.....	349
6.6.5.6.	Предлагаемая последовательность выполнения теста PX2612.....	349
7.	Техническое обслуживание	351
7.1.	Диагностика модуля.....	351
7.1.1.	Диагностика в одно касание.....	351
7.1.2.	Диагностика с помощью светодиода.....	353
7.1.2.1.	DG (Диагностика).....	353
7.1.2.2.	WD (Watchdog)	353
7.1.2.3.	Светодиодные индикаторы разъема RJ45	354
7.1.3.	Диагностика через WEB.....	354
7.1.4.	Диагностика через Explorer	356
7.1.5.	Диагностика через Variables	357
7.1.5.1.	Обобщенная диагностика.....	357
7.1.5.2.	Детальная диагностика	360
7.1.6.	Диагностика с помощью функциональных блоков	370
7.1.6.1.	Получить информацию о задаче.....	370
7.2.	Графический дисплей.....	372
7.3.	Системный журнал	374

СОДЕРЖИМОЕ

7.4.	Не загружается приложение при запуске.....	374
7.5.	Отказ источника питания	374
7.6.	Общие проблемы	375
7.7.	Поиск и устранение неисправностей.....	375
7.8.	Профилактическое обслуживание.....	376
8.	Приложение. Взаимодействие DNP3	377
8.1.	Профиль устройства DNP3.....	377
8.2.	Таблица реализации DNP V3.0.....	378

1. Введение

Программируемые контроллеры серии Nexto - это оптимальное решение для промышленной автоматизации и управления системами. Благодаря высоким встроенным технологиям продукты этого семейства способны распределенно и с резервированием управлять сложными промышленными системами, станками, высокопроизводительными производственными линиями и самыми передовыми процессами Индустрии 4.0. Современные и высокоскоростные устройства серии Nexto используют передовые технологии для обеспечения надежности и связи, помогая повысить производительность различных предприятий.

Компактные, надежные и с высокой степенью готовности, продукты серии обладают отличной производительностью и возможностями расширения стойки. Их архитектура обеспечивает легкую интеграцию с сетями контроля, управления и полевыми сетями, а также резервирование процессора и источника питания. Кроме того, оборудование семейства обладает расширенными возможностями диагностики и "горячей" замены, что позволяет минимизировать или исключить простои в обслуживании и обеспечить непрерывность производственного процесса.



Рисунок 1: NX3030

1.1. Nexto Series

Серия Nexto - это мощная и полная серия программируемых контроллеров (ПЛК) с эксклюзивными и инновационными характеристиками. Благодаря 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. Эти модули в сочетании с мощным 32-разрядным процессором и высокоскоростной шиной на базе 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. Кроме того, в серии Nexto имеются модули для управления движением, коммуникационные модули, охватывающие наиболее популярные полевые сети.

В шине серии Nexto используется передовая технология, основанная на высокоскоростном интерфейсе Ethernet, что позволяет обмениваться входной и выходной информацией и данными между несколькими контроллерами в одной системе. Система может быть легко разделена и распределена по всему полю, позволяя использовать

1. ВВЕДЕНИЕ

расширение шины с той же производительностью, что и локальный модуль, что делает возможным использование любого модуля в локальном каркасе или в каркасах расширения без каких-либо ограничений. Для связи между каркасами расширения используется простой стандартный кабель Ethernet.



Рисунок 2: Серия Nexto – обзор

1.2. Инновационные возможности

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



Работа без батарей: Для поддержания памяти и работы часов реального времени в серии Nexto не требуется батарея. Эта особенность чрезвычайно важна, поскольку снижает потребность в обслуживании системы и позволяет использовать ее в удаленных местах, где обслуживание может быть затруднено. Кроме того, данная особенность является экологически безопасной.



Система Easy Plug: В серии Nexto реализован эксклюзивный способ подключения и отключения клеммных блоков ввода/вывода. Клеммные блоки легко снимаются одним движением и без использования специальных инструментов. Для установки клеммного блока обратно в модуль передняя крышка помогает выполнить процедуру монтажа, устанавливая клеммный блок на модуль.



Многоблочное хранение данных: В процессорах серии Nexto пользователю доступно несколько видов памяти, что позволяет подобрать оптимальный вариант для любых нужд пользователя. Эти виды памяти делятся на энергонезависимые и энергонезависимые. Для энергонезависимой памяти в процессорах серии Nexto предусмотрены адресуемый вход (%I), адресуемый выход (%Q), адресуемая память (%M), память данных и резервная память данных. Для приложений, требующих энергонезависимой функциональности, процессоры серии Nexto предлагают сохраняемую адресную память (%Q), сохраняемую память данных, постоянную адресную память (%Q), постоянную память данных, память программ, память исходного кода, файловую систему процессора (doc, PDF, data) и интерфейс для карт памяти.



One Touch Diag: One Touch Diag - это эксклюзивная функция, которую серия Nexto привносит в ПЛК. Благодаря этой новой концепции пользователь может проверить диагностическую информацию любого модуля, присутствующего в системе, непосредственно на графическом дисплее процессора одним нажатием на диагностический переключатель соответствующего модуля. OTD - это мощный диагностический инструмент, который можно использовать в автономном режиме (без супервизора или программатора), что сокращает время обслуживания и ввода в эксплуатацию.

OFD - On Board Full Documentation: Процессоры серии Nexto способны хранить полную проектную документацию в собственной памяти. Эта функция может быть очень удобна для целей резервного копирования и технического обслуживания, поскольку вся информация хранится в одном надежном месте.

1. ВВЕДЕНИЕ

ETD - электронная метка на дисплее: Еще одной эксклюзивной функцией, которую серия Nexto привносит в ПЛК, является электронная метка на дисплее. Эта новая функциональная возможность позволяет проверять названия тегов для любого контакта ввода/вывода или модуля, используемого в системе, непосредственно на графическом дисплее процессора. Вместе с этой информацией пользователь может проверить и ее описание. Эта функция чрезвычайно полезна при обслуживании и поиске неисправностей.

DHW - Double Hardware Width: модули серии Nexto были разработаны с целью экономии места в пользовательских шкафах или машинах. По этой причине в серии Nexto предусмотрено два варианта ширины модулей: Double Width (требуется два слота для установки в стойку объединительной платы) и Single Width (требуется только один слот для установки в стойку объединительной платы). Эта концепция позволяет использовать компактные модули ввода/вывода с высокой плотностью точек ввода/вывода наряду со сложными модулями, такими как центральные процессоры, мастера полевых шин и модули питания.

Высокоскоростной центральный процессор: Все центральные процессоры серии Nexto были разработаны с целью обеспечения выдающейся производительности, позволяющей удовлетворить широкий спектр требований приложений.



iF Product Design Award 2012: Серия Nexto стала лауреатом премии iF Product Design Award 2012 в группе "Промышленность + квалифицированные рабочие". Эта награда признана во всем мире как знак качества и совершенства и считается "Оскаром" в области дизайна в Европе.

1.3. Документы, относящиеся к данному руководству

Для получения дополнительной информации о серии Nexto, кроме этой, можно обратиться к другим документам (руководствам и техническим характеристикам). Эти документы доступны в последней версии на сайте http://www.altus.com.br/site_en/.

Каждое изделие имеет документ, оформленный в виде Технических характеристик (CE), в котором описаны особенности изделия. Кроме того, к изделию могут прилагаться руководства по эксплуатации (коды руководств указаны в CE).

Например, для модуля NX2020 в CE указана информация об особенностях использования 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	Серия Nexto - Технические характеристики	Португальский
CS114000	Серия Nexto - Технические характеристики	Испанский
CE114102	Технические характеристики NX3030	Английский
CT114102	Технические характеристики NX3030	Португальский
CS114102	Технические характеристики и конфигурации NX3030	Испанский
CE114200	Технические характеристики модуля питания NX8000	Английский
CT114200	Fonte de Alimentação NX8000 Технические характеристики	Португальский
CS114200	Технические характеристики источника питания NX8000	Испанский
CE114700	Технические характеристики стоек объединительных панелей серии Nexto	Английский
CT114700	Технические характеристики стоек объединительных панелей серии Nexto	Португальский

1. ВВЕДЕНИЕ

CS114700	Технические характеристики стоек серии Nexto	Испанский
CE114810	Технические характеристики Аксессуары для объединительных стоек серии Nexto	Английский
CT114810	Аксессуары серии Nexto для стоек объединительных панелей	Португальский
CS114810	Боковые крышки для стоек	Испанский
CE114900	Технические характеристики модуля резервирования NX4010	Английский
CT114900	Технические характеристики модуля резервирования NX4010	Португальский
CS114900	Технические характеристики модуля резервирования NX4010	Испанский
CE114902	Технические характеристики Nexto Series PROFIBUS-DP Master	Английский
CT114902	Технические характеристики ведущего устройства PROFIBUS-DP серии Nexto	Португальский
CS114902	Технические характеристики ведущего модуля Profibus-DP серии Nexto	Испанский
CE114903	Технические характеристики Ethernet-модуля серии Nexto	Английский
CT114903	Технические характеристики Ethernet-модуля серии Nexto	Португальский
CS114903	Технические характеристики Ethernet-модуля серии Nexto	Испанский
CE114908	Технические характеристики PROFIBUS-DP головок NX5110 и NX5210	Английский
CT114908	Технические характеристики интерфейсов с головками NX5110 и NX5210 PROFIBUSDP	Португальский
CS114908	Технические характеристики и конфигурации Интерфейсные головки PROFIBUS-DP NX5110 и NX5210	Испанский
Код	Описание	Язык
CT112500	Технические характеристики панели управления резервированием PX2612	Португальский
MU214600	Руководство пользователя серии Nexto	Английский
MU214000	Руководство пользователя серии Nexto	Португальский
MU214615	Руководство пользователя процессора NX3030	Английский
MU214103	Руководство пользователя процессора NX3030	Португальский язык
MU299609	Руководство пользователя MasterTool IEC XE	Английский
MU299048	Руководство пользователя MasterTool IEC XE	Португальский
MP399609	Руководство по программированию MasterTool IEC XE	Английский
MP399048	Руководство по программированию MasterTool IEC XE	Португальский
MU214601	Руководство пользователя NX5001 PROFIBUS-DP Master	Английский
MU214001	Руководство пользователя мастера PROFIBUS-DP NX5001	Португальский
MU214608	Руководство по использованию головки Nexto PROFIBUS-DP	Английский
MU214108	Руководство по использованию головки Nexto PROFIBUS-DP	Португальский
MU219000	Руководство по эксплуатации серии Ponto	Английский
MU209000	Руководство по эксплуатации серии Ponto	Португальский

1. ВВЕДЕНИЕ

MU209508	Головка PROFIBUS PO5063V1 и резервная головка PROFIBUS PO5063V5 Руководство по эксплуатации	Португальский
MU219511	Руководство по эксплуатации головки PROFIBUS PO5064 и резервной головки PROFIBUS PO5065	Английский
MU209511	Руководство по использованию головки PROFIBUS PO5064 и резервной головки PROFIBUS PO5065	Португальский
MU209020	Руководство по использованию сети HART по PROFIBUS	Португальский язык
MU223603	IEC 60870-5-104 Документ профиля серверного устройства	Английский
MU214603	Руководство по HART для серии Nexto	Английский
MU214610	Расширенные функции управления Руководство пользователя	Английский
NAP151	Использование OPC Tunneller	Португальский
NAP165	Связь OPC UA с контроллерами ALTUS	Португальский
NAP165_ing	Связь OPC UA с контроллерами ALTUS	Английский

Таблица 1: Связанные документы

1.4. Визуальный осмотр

Прежде чем приступить к монтажу, рекомендуется внимательно осмотреть оборудование и убедиться в отсутствии транспортных повреждений. Убедитесь, что все требуемые детали находятся в идеальном состоянии. В случае обнаружения повреждений сообщите об этом транспортной компании или ближайшему дистрибьютору Altus.

ОСТОРОЖНО

Перед тем как снять модули с корпуса, необходимо разрядить возможную статическую энергию, накопившуюся в теле. Для этого перед обращением с модулями необходимо прикоснуться (голыми руками) к любой металлической заземленной поверхности. Такая процедура гарантирует, что предельные значения статической энергии модулей не будут превышены.

Важно регистрировать серийный номер каждого полученного оборудования, а также ревизии программного обеспечения, если они существуют. Эта информация необходима в случае обращения в службу технической поддержки Altus.

1.5. Техническая поддержка

Для получения информации о технической поддержке Altus в Сан-Леопольдо, РС, звоните по телефону +55 51 3589-9500. Для получения дополнительной информации о технической поддержке Altus, существующей в других местах, см. <https://www.altus.com.br/en/> или отправьте письмо по адресу altus@altus.com.br.

Если оборудование уже установлено, то на момент обращения в службу технической поддержки необходимо иметь следующую информацию:

1. ВВЕДЕНИЕ

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

1.6. Предупреждающие сообщения, используемые в данном руководстве

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

ОПАСНО

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

ОСТОРОЖНО

Сообщает о деталях конфигурации, приложения или установки, которые необходимо учесть, чтобы избежать ситуаций, способных привести к сбою системы и последующим последствиям.

ВНИМАНИЕ

Определяет детали конфигурации, применения и установки, направленные на достижение максимальных эксплуатационных характеристик системы.

2. Техническая характеристика

В этой главе представлены все технические характеристики NX3030.

2.1. Панели и соединения

На следующем рисунке показана передняя панель процессора.



Рисунок 3: NX3030

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

Сразу под графическим дисплеем расположены 2 светодиода, служащие для индикации аварийной диагностики и работы сторожевого таймера. Описание светодиодов приведено в таблице ниже. Более подробная информация о состоянии и значении светодиодов приведена в разделе "Диагностика с помощью светодиодов".

LED	Описание
DG	Диагностика LED
WD	Watchdog LED

Таблица 2: Описание светодиодов

В процессорах серии Nexto пользователю доступны два переключателя. Описание этих переключателей приведено в таблице ниже. Более подробную информацию о переключателе диагностики см. в разделах One Touch Diag и Информационно-конфигурационное меню процессора. Дополнительную информацию о переключателе MS см. в разделе Карта памяти.

Клавиша	Описание
Переключатель диагностики	Переключатель расположен на верхней части модуля. Используется для визуализации диагностики на графическом дисплее или для навигации по информационному меню и конфигурации процессора.

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

Клавиша	Описание
MS	Переключатель, расположенный на передней панели. Используется для безопасного извлечения карты памяти.

Таблица 3: Описание клавиш

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

Интерфейсы	Описание
NET 1	Коммуникационный разъем RJ45 стандарта 10/100Base-TX. Обеспечивает связь "точка-точка" или сетевую связь. Более подробная информация об использовании приведена в разделе "Конфигурация интерфейсов Ethernet" .
NET 2	Коммуникационный разъем RJ45 стандарта 10/100Base-TX. Позволяет организовать связь "точка-точка" или сеть. Дополнительную информацию по использованию см. в разделе Конфигурация интерфейсов Ethernet .
COM 1	Разъем DB9 для связи по стандарту RS-232. Позволяет работать в режиме "точка-точка" или в сети. Более подробная информация об использовании приведена в разделе Конфигурация последовательных интерфейсов .
COM 2	Дополнительную информацию по использованию см. в разделе Конфигурация последовательных интерфейсов .
ГНЕЗДО ПАМЯТИ	Слот для карты памяти. Позволяет использовать карту памяти для хранения различных типов данных, таких как: пользовательские журналы, Web-страницы, проектная документация и файлы. Более подробная информация об использовании приведена в разделе Карта памяти .

Таблица 4: Интерфейсы подключения

2.2. Общие характеристики

2.2.1. Общие характеристики

	NX3030
Возможность установки в стойку с задней панелью	2 sequential slots
Встроенный источник питания	Нет
Локальный интерфейс Ethernet TCP/IP	2
Последовательный интерфейс	2
CAN-интерфейс	Нет
USB-порт Хост	Нет
Интерфейс карты памяти	Да

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

Часы реального времени (RTC)	Да Разрешение 1 мс и максимальная дисперсия 2 с в сутки.
Сторожевой таймер	Да
Индикация состояния и диагностики	Графический дисплей Светодиоды Веб-страницы Внутренняя память процессора
Языки программирования	Структурированный текст (ST) Лестничная диаграмма (LD) Последовательная функциональная диаграмма (SFC) Функциональная блок-схема (FBD) Непрерывная функциональная диаграмма (CFC)
Задачи	Циклический (периодический) Событие (программное прерывание) Внешний (аппаратное прерывание) Свободный ход (непрерывный) Состояние (программное прерывание)
Изменения в режиме онлайн	Да
Максимальное количество задач	32
Максимальное количество шин расширения	24
Поддержка избыточности расширения шины	Да
Максимальное количество модулей ввода/вывода на шине	128
Максимальное количество дополнительных интерфейсных модулей Ethernet TCP/IP	6
Поддержка резервирования интерфейсов Ethernet TCP/IP	Да
Максимальное количество сетей PROFIBUS-DP (при использовании ведущих модулей PROFIBUS-DP)	4
Поддержка резервирования сети PROFIBUS-DP	Да
Поддержка резервирования (полукластеры)	Да
Поддержка горячей замены	Да
Событийно-ориентированное представление данных (SOE)	Да
Протокол	DNP3
Максимальный размер очереди событий	1000
Разработка веб-страниц (доступна по протоколу HTTP)	Нет
Диагностика одно-касание (OTD)	Да
	NX3030
Electronic Tag on Display (ETD)	Да

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА





Стандарты и сертификаты IEC 61131-3	Да
Одобрение типа DNV–DNV-CG-0339 (TAA000013D)	Да
IEC 61131-2	Да
 CE – 2014/35/EU (LVD) и 2014/30/EU (EMC)	Да
 RoHS – 2011/65/EU	Да
 UL Listed – UL61010-1 (файл E473496)	Да
 EAC – CU TR 004/2011 (LVD) и CU TR 020/2011 (EMC)	Да

Таблица 5: Общие характеристики

Примечания:

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

Максимальное количество модулей ввода/вывода на шине: Максимальное количество модулей ввода/вывода - это сумма всех модулей на локальной шине и расширениях.

Журнал событий (SOE): Типы данных приведены в профиле устройства DNP3.

2.2.2. Память

	NX3030
Адресная память входных переменных (%I)	96 Кбайт
Память адресуемых выходных переменных (%Q)	96 Кбайт
Память переменных прямого представления (%M)	64 Кбайт
Память символьных переменных	6 Мбайт
Максимальный объем памяти, конфигурируемый как ретенционная или персистентная	112 Кбайт
Память данных с полным резервированием	736 Кбайт
Память входных переменных прямого представления (%I)	80 Кбайт
Память переменных прямого представления выходных данных (%Q)	80 Кбайт
Память переменных прямого представления (%M)	64 Кбайт
Память символьных переменных	512 Кбайт
Память программ	8 Мбайт
Память исходного кода (резервная копия)	120 Мбайт
Память пользовательских файлов	32 Мбайт

Таблица 6: Память

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. Адресные входные переменные могут использоваться для сопоставления точек цифровых или аналоговых входов. В качестве примера можно привести 8 цифровых входов на один байт и одну точку аналогового входа на два байта.

Общая память адресуемых выходных переменных (%Q): Область, в которой хранятся адресуемые выходные переменные. Адресуемые переменные означают, что к ним можно обращаться непосредственно по нужному адресу. Например: %QB0, %QW100. Адресные выходные переменные могут использоваться для сопоставления точек цифрового или аналогового выхода. В качестве примера можно привести 8 цифровых выходов на один байт и одну точку аналогового выхода на два байта. Адресные выходные переменные могут быть сконфигурированы как сохраняемые, постоянные или избыточные, но общий размер при этом не изменяется.

Процессор Nexto серии NX3030 позволяет определить область избыточных переменных в область памяти %Q выходных переменных прямого представления. Подмножество типов памяти выходных переменных прямого представления является частью общей доступной памяти.

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

Память символьных переменных: Область, в которой размещаются символьные переменные. Символьные переменные - это переменные МЭК, созданные в POU и GVL в процессе разработки приложения, к которым нет прямого обращения в памяти. Символьные переменные могут быть определены как ретенционные или постоянные, в этом случае будут использоваться области памяти ретенционных символьных переменных или памяти постоянных символьных переменных соответственно. Система ПЛК выделяет переменные в этой области, поэтому 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 свободным пространством в сообщениях компиляции инструмента MasterTool IEC XE.

Постоянная и сохраняемая память символьных переменных: Область, в которой размещаются ретенционные символьные переменные. Удерживаемые данные сохраняют свои значения даже после циклов выключения и включения питания процессора. Постоянные данные сохраняют свои значения даже после загрузки нового приложения в процессор.

ВНИМАНИЕ

Объявление и использование символьных постоянных переменных должно осуществляться исключительно через объект Persistent Vars, который может быть включен в проект через древовидное представление в Приложении -> Добавить объект -> Постоянные переменные. Не следует использовать выражение VAR PERSISTENT при объявлении полевых переменных POU.

Полный список того, когда символьные постоянные переменные сохраняют свои значения и когда значение теряется, приведен в таблице ниже. Кроме объявленного в таблице размера персистентной области, для хранения информации о персистентных переменных зарезервированы 44 байта (не доступны для использования).

В таблице ниже показано поведение ретенционных и персистентных переменных для различных ситуаций, в которых "-" означает потерю значения, а "X" - сохранение значения.

Клмнда	VAR	VAR RETAIN	VAR PERSISTENT
Сброс теплого / Цикл включения/выключения питания	-	X	X
Сброс холодного	-	-	X
Сброс исходного состояния	-	-	-

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

Извлечение процессора или блока питания из стойки под напряжением	-	-	-
Загрузить	-	-	X
Изменение в режиме онлайн	X	X	X
Перезагрузить ПЛК	-	X	X
Очистить все	-	-	X
Процесс сброса (IEC 60870-5-104)	-	X	X

Таблица 7: Поведение посткомандных переменных

В версиях младше или равных 1.5.1.0 для NX3010, NX3020 и NX3030 ретенционная и персистентная символьная память и память адресуемых выходных переменных (%Q) имели фиксированный максимальный размер. В таблице ниже можно ознакомиться с максимальными размерами, допустимыми в этих старых версиях.

В версиях, выше указанных, процессоры допускают гибкие размеры ретентивной и персистентной памяти. Более подробная информация приведена в разделе ["Области ретенционной и персистентной памяти"](#).

	NX3030
Ретенционная адресная память выходных переменных (%Q)	16 Кбайт
Персистентная адресная память выходных переменных (%Q)	48 Кбайт
Ретенционная память символьных переменных	32 Кбайт
Постоянная память символьных переменных	16 Кбайт

Таблица 8: Ретентивная и персистентная память в старых версиях

В случае команды Clean All, если приложение было модифицировано таким образом, что постоянные переменные были удалены, вставлены в начало списка или иным образом изменили свой тип, значение этих переменных будет потеряно (при запросе на загрузку инструментом MasterTool). Поэтому рекомендуется, чтобы изменения в постоянных переменных GVL включали только добавление новых переменных в список.

Общий объем избыточной памяти данных: Резервная память данных - это максимальная область памяти, которая может использоваться в качестве резервной памяти между двумя резервируемыми процессорами. Эта величина не является различной памятью, обратите внимание, что сумма всех избыточных переменных (адресуемая входная переменная, адресуемая выходная переменная, адресуемая переменная, символьная переменная, сохраняемая символьная переменная, постоянная символьная переменная) должна быть меньше или равна доступной избыточной памяти данных.

Программная память: Память программ - это максимальный объем, который может быть использован для хранения пользовательского приложения. Эта область разделяется с памятью исходного кода, поэтому общая площадь равна сумме "памяти программ" и "памяти исходного кода".

Память исходного кода (резервная): Эта область памяти используется в качестве резервной копии проекта. Если пользователь захочет импортировать проект, то MasterTool IEC XE получит необходимую информацию в этой области. Во избежание потери важной информации необходимо следить за тем, чтобы проект, сохраненный в качестве резервной копии, был актуальным. Эта область является общей с памятью исходного кода, поэтому общая площадь равна сумме "памяти программы" и "памяти исходного кода".

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

2.2.3. Протоколы

	NX3030	Интерфейс
--	--------	-----------

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

Открытый протокол	Да	COM1 / COM2
Ведущее устройство MODBUS RTU	Да	COM1 / COM2
MODBUS RTU Slave	Да	COM1 / COM2
Клиент MODBUS TCP	Да	NET1 / NET2
MODBUS TCP Server	Да	NET1 / NET2
MODBUS RTU over TCP Client	Да	NET1 / NET2
MODBUS RTU через TCP-сервер	Да	NET1 / NET2
CANopen Ведущий	Нет	-
CANopen Slave	Нет	-
Низкий уровень CAN	Нет	-
SAE J-1939	Нет	-
Сервер OPC DA	Да	NET1 / NET2
OPC UA Server	Да	NET1 / NET2
EtherCAT Master	Да	NET1 / NET2
Агент SNMP	Да	NET1 / NET2
DNP3 Server (Event-oriented data)	Да	NET1 / NET2
Сервер IEC 60870-5-104	Да	NET1 / NET2
Сканер EtherNet/IP	Да	NET1 / NET2
EtherNet/IP-адаптер	Да	NET1 / NET2
Клиент MQTT	Да	NET1 / NET2
Клиент SNTP (для синхронизации часов)	Да	NET1 / NET2
PROFINET-контроллер	Да	NET1 / NET2
PROFINET-устройство	Нет	-

Таблица 9: Протоколы

Примечание:

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

2.2.4. Последовательные интерфейсы

2.2.4.1. COM 1

	COM 1
Разъем	Shielded female DB9
Физический интерфейс	RS-232C
Сигналы модема	RTS, CTS, DCD
Скорость передачи данных	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

Изоляция Логика к последовательному порту Последовательный порт - защитное заземление ⚡	Не изолирован 1000 В переменного тока / 1 минута
---	---

Таблица 10: Характеристики последовательного интерфейса COM 1

2.2.4.2. COM 2

	COM 2
Разъем	Экранированный гнездовой разъем DB9
Физический интерфейс	RS-422 или RS-485 (в зависимости от выбранного кабеля)
Направление связи	RS-422: полный дуплекс RS-485: полудуплекс
RS-422 макс. приемопередатчиков	11 (1 передатчик и 10 приемников)
RS-485 макс. приемопередатчики	32
Окончание	Да (опционально при выборе кабеля)
Скорость передачи данных	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Изоляция Логика к последовательному порту Последовательный порт - защитное заземление ⚡	1000 В переменного тока / 1 минута 1000 В переменного тока / 1 минута

Таблица 11: Характеристики последовательного интерфейса COM 2

Примечания:

Физический интерфейс: В зависимости от конфигурации используемого кабеля можно выбрать вид физического интерфейса: RS-422 или RS-485. Список кабелей приведен в разделе "Сопутствующие товары".

Максимальное количество приемопередатчиков RS-422: Это максимальное количество интерфейсов RS-422, которые могут быть использованы на одной шине. Максимальное количество приемопередатчиков RS-485: Это максимальное количество интерфейсов RS-485, которые можно использовать на одной шине.

2.2.5. Интерфейсы Ethernet

2.2.5.1. NET 1

	NET 1
Разъем	Экранированный гнездовой RJ45
Автоматический кроссовер	Да

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

Максимальная длина кабеля	100 м
Тип кабеля	UTP или ScTP, категория 5
Скорость передачи данных	10/100 Мбит/с
Физический уровень	10/100 BASE-TX (полный дуплекс)
Канальный уровень	LLC ("Управление логическим каналом")
Сетевой уровень	IP ("Интернет-протокол"))
Транспортный уровень	TCP ("Протокол управления передачей")
Диагностика	UDP ("Протокол пользовательских дейтаграмм")
Изоляция Интерфейс Ethernet для подключения к логике и заземлению	1500 В переменного тока / 1 минута

Таблица 12: Характеристики интерфейса Ethernet NET 1

Интерфейс NET 1 - это интерфейс, используемый для программирования с помощью инструмента MasterTool IEC XE.

2.2.5.2. NET 2

	NET 2
Разъем	Экранированный гнездовой RJ45
Автоматический кроссовер	Да
Максимальная длина кабеля	100 м
Тип кабеля	UTP или ScTP, категория 5
Скорость передачи данных	10/100 Мбит/с
Физический уровень	10/100 BASE-TX (полный дуплекс)
Канальный уровень	LLC ("Управление логическим каналом")
Сетевой уровень	IP ("Интернет-протокол")
Транспортный уровень	TCP ("Протокол управления передачей") UDP ("Протокол пользовательских дейтаграмм")
Диагностика	Светодиоды - зеленый (скорость), желтый (соединение/активность)
Изоляция Интерфейс Ethernet для подключения к логике и заземлению	1500 В переменного тока / 1 минута
Интерфейс Ethernet к интерфейсу Ethernet	1500 В переменного тока / 1 минута

Таблица 13: Особенности интерфейса Ethernet NET 2

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

2.2.6. Интерфейс карты памяти

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

	Карта памяти
Максимальная мощность	32 Гбайт
Минимальная вместимость	2 Гбайт
Тип	MiniSD
Файловая система	FAT32
Безопасное извлечение карты	Да, нажатием кнопки MS

Таблица 14: Особенности интерфейса карты памяти

Примечания:

Максимальная емкость: Для корректной работы процессора Nexto CPU емкость карты памяти должна быть меньше или равна этому пределу, иначе процессор Nexto CPU может не обнаружить карту памяти или даже возникнуть проблемы при передаче данных.

Минимальная емкость: Для корректной работы процессора Nexto емкость карты памяти должна быть больше или равна этому значению, иначе процессор Nexto может не обнаружить карту памяти или даже возникнуть проблемы при передаче данных.

Файловая система: Рекомендуется форматировать карту памяти с помощью процессора Nexto CPU, иначе это может привести к снижению производительности интерфейса карты памяти.

2.2.7. Экологические характеристики

	NX3030
Ток, потребляемый по шине питания	1000 мА
Рассеиваемая мощность	5 Вт
Рабочая температура	От 0 до 60 °С
Температура хранения	от -25 до 75 °С
Относительная влажность	От 5% до 96%, без конденсации
Конформное покрытие	Да
Уровень защиты IP	IP 20
Размеры модуля (Ш x В x Г)	36,00 x 114,63 x 115,30 мм
Размеры упаковки (Ш x В x Г)	44,00 x 122,00 x 147,00 мм
Масса	350 г
Масса с упаковкой	400 г

Таблица 15: Экологические характеристики

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

Примечания:

Конформное покрытие электронных схем: Покрытие электронных схем защищает внутренние детали изделия от влаги, пыли и других агрессивных для электронных схем элементов.

2.3. Совместимость с другими продуктами

Для разработки приложения для процессоров серии Nexto необходимо проверить версию MasterTool IEC XE. В следующей таблице указана минимально необходимая версия (где были представлены контроллеры) и соответствующая версия микропрограммы на тот момент:

Nexto Series CPUs	MasterTool IEC XE	Версия микропрограммы
NX3010, NX3020, NX3030	1.00 до 2.09	1.2.0.9 до 1.7.0.14
NX3010, NX3020, NX3030	3.00 или выше	1.8.3.0 или выше

Таблица 16: Совместимость с другими продуктами

Кроме того, в ходе разработки MasterTool IEC XE могут быть включены некоторые функции (например, специальные функциональные блоки и т.д.), которые могут ввести требование к минимальной версии микропрограммы. Во время загрузки приложения MasterTool IEC XE проверяет версию микропрограммы, установленную на контроллере, и, если она не соответствует минимальным требованиям, выдает сообщение с запросом на обновление. Последнюю версию микропрограммы можно загрузить с сайта компании Altus, и она полностью совместима с предыдущими версиями приложений.

2.4. Производительность

Производительность процессоров серии Nexto зависит от:

- Время применения пользователем
- Интервал приложения
- Время работы системы
- Количество модулей (данные процесса, вход/выход и т.д.)

2.4.1. Сроки подачи заявок

Время выполнения приложения Nexto процессоров зависит от следующих переменных:

- Время чтения входных данных (локальных и удаленных)
- Время выполнения заданий
- Время записи выходных данных (локальное и удаленное)

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

2.4.2. Время выполнения инструкций

В таблице ниже представлено необходимое время выполнения различных инструкций в процессорах серии Nexto.

Инструкция	Язык	Переменные	Время выполнения инструкции (мкс)
------------	------	------------	-----------------------------------

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

1000 Contacts	LD	BOOL			6 < 5000	
		1000 Divisions	ST	INT		43
				REAL		81
			LD	INT		43
				REAL		81
		1000 Multiplications	ST	INT		15
				REAL		23
			LD	INT		15
				REAL		23
		1000 Sums	ST	INT		15
				REAL		23
			LD	INT		15
REAL	23					
1000 PID	ST	REAL				

Таблица 17: Время выполнения инструкций

2.4.3. Время инициализации

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

2.4.4. Интервальное время

Время интервала выполнения основной задачи процессора может быть установлено в диапазоне от 5 до 750 мс.

2.5. Физические размеры

Размеры в мм.

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

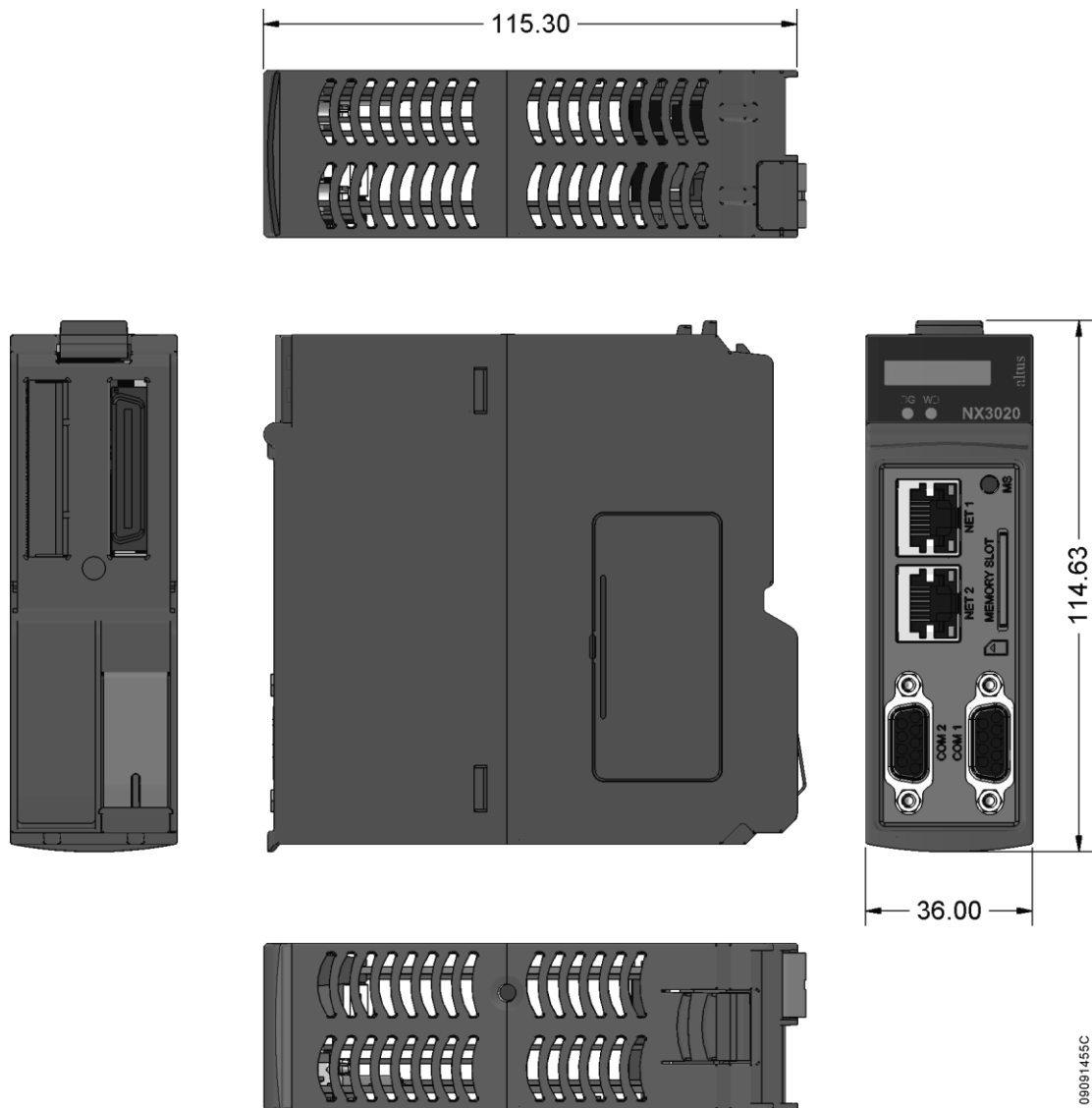


Рисунок 4: Физические размеры процессора

2.6. Данные о закупках

2.6.1. Включенные элементы

В комплект поставки входят следующие элементы:

- NX3030 модуль

2.6.2. Код товара

Для приобретения продукта следует использовать следующий код:

Код	Описание
NX3030	Высокоскоростной процессор, 2 порта Ethernet, 2 последовательных канала, интерфейс для карт памяти, поддержка удаленного расширения стойки и резервирования

Таблица 18: Код продукта

2.7. Сопутствующие продукты

Следующие продукты при необходимости приобретаются отдельно:

Код	Описание
MT8500	MasterTool IEC XE
AL-2600	Разветвитель и терминатор сети RS-485
AL-2306	Кабель RS-485 для сети MODBUS или CAN
AL-2319	Кабель RJ45-RJ45
AL-1729	Кабель RJ45-CMDB9
AL-1748	Кабель CMDB9-CFDB9
AL-1752	Кабель CMDB9-CMDB9
AL-1753	Кабель CMDB9-CMDB25
AL-1754	Кабель CMDB9-CFDB9
AL-1761	Кабель CMDB9-CMDB9
AL-1762	Кабель CMDB9-CMDB9
AL-1763	CMDB9-Кабель для клеммного блока
AL-1766	Кабель CFDB9-терминальный блок
NX9101	Карта памяти microSD 32 Гб с адаптерами miniSD и SD
NX9202	Кабель RJ45-RJ45 2 м
NX9205	Кабель RJ45-RJ45 5 м
NX9210	Кабель RJ45-RJ45 10 м
NX9000	8-слотовая стойка объединительной панели
NX9001	12-слотовая стойка объединительной панели
NX9002	16-слотовая стойка объединительной панели
NX9003	24-слотовая стойка объединительной панели
NX8000	Модуль питания 30 Вт 24 В пост. тока

Таблица 19: Сопутствующие товары

Примечания:

MT8500: MasterTool IEC XE выпускается в четырех различных версиях: LITE, BASIC, PROFESSIONAL и ADVANCED. Более подробная информация приведена в руководстве пользователя MasterTool IEC XE - MU299609.

AL-2600: Этот модуль предназначен 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, используется для сетей на базе RS-485 или CAN.

AL-2319: Два разъема RJ45 для программирования центральных процессоров серии Nexto и Ethernet-точки с другим устройством, имеющим интерфейс Ethernet.

2. ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА

AL-1729: Кабель стандарта RS-232C с одним разъемом RJ45 и одним разъемом DB9 male для связи между процессорами серии Nexto и другими устройствами Altus серий DUO, Piccolo и Ponto. AL-1748: Кабель стандарта RS-232C с одним гнездовым разъемом DB9 и одним гнездовым разъемом DB9 для связи между процессорами серии Nexto и изделиями Altus серии Cimrex.

AL-1752: Стандартный кабель RS-232C с двумя гнездами DB9 для связи между процессорами серии Nexto и продуктами Altus серий H и iX.

AL-1753: Стандартный кабель RS-232C с одним гнездовым разъемом DB9 и одним гнездовым разъемом DB25 для связи между процессорами серии Nexto и продуктами Altus серии H.

AL-1754: Кабель стандарта RS-232C с одним гнездовым разъемом DB9 и одним гнездовым разъемом DB9 для связи между процессорами серии Nexto и Altus серии Exter или последовательным портом стандарта RS-232C микрокомпьютера.

AL-1761: Кабель стандарта RS-232C с двумя гнездами DB9 для связи между процессорами серии Nexto и продуктами Altus серии AL.

AL-1762: Кабель стандарта RS-232C с двумя гнездами DB9 для связи между процессорами серии Nexto.

AL-1763: Кабель с одним гнездовым разъемом DB9 и клеммной колодкой для связи между центральными процессорами серии Nexto и изделиями со стандартной клеммной колодкой RS-485/RS-422.

AL-1766: Кабель с гнездовым разъемом DB9 и клеммами для связи между HMI P2 и контроллерами Nexto Xpress/NX3003.

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

3. Установка

В этой главе представлены необходимые процедуры для физической установки процессоров серии Nexto, а также меры предосторожности при других видах монтажа в панели, где установлен процессор.

ОСТОРОЖНО

Если оборудование используется не так, как указано в данном руководстве, это может привести к нарушению защиты, обеспечиваемой оборудованием.

3.1. Механический монтаж

Процессоры серии Nexto должны устанавливаться в стойку объединительной панели в позиции 2, непосредственно рядом с модулем питания. Вся информация по механическому монтажу и установке модулей приведена в документе MU214600 - Руководство пользователя серии Nexto.

3.2. Электромонтаж

ОПАСНО

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

Питание процессоров осуществляется от модуля питания, который подает питание на процессоры через стоечное соединение объединительной панели. Он не нуждается во внешнем подключении. Заземление модуля осуществляется через контакт между пружиной заземления модуля и стойкой объединительной панели.

На рисунке ниже показана электрическая схема процессоров серии Nexto, установленных в стойку объединительной панели серии Nexto.

Расположение разъемов на рисунке является лишь иллюстративным.

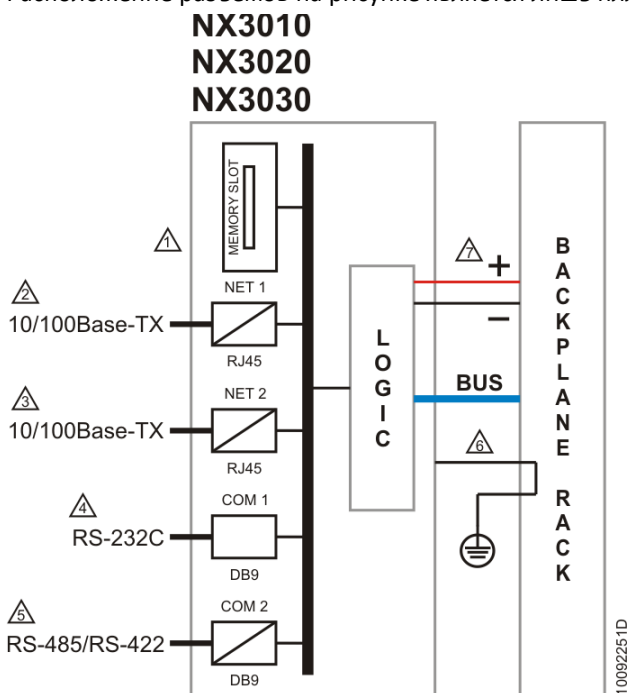










Рисунок 5: Электрическая схема процессоров NX3010, NX3020 и NX3030

Примечания к диаграмме:

3. УСТАНОВКА

-  Интерфейс карты памяти.
-  Интерфейс Ethernet стандарта 10/100Base-TX для программирования, отладки и сетевого подключения MODBUS TCP или других протоколов.
-  Интерфейс Ethernet 10/100Base-TX для сетевого соединения MODBUS TCP или других протоколов (только для NX3020 и NX3030).
-  Последовательный интерфейс RS-232C для сетевого соединения MODBUS RTU или других протоколов.
-  Последовательный интерфейс RS-485/RS-422 - стандарт для сетевого соединения MODBUS RTU или других протоколов. Выбор физического интерфейса зависит от используемого кабеля.
-  Модуль заземляется через стойку объединительной платы серии Nexto.
-  Питание осуществляется от разъема стойки объединительной панели. Внешние подключения не требуются.
-  Клемма защитного заземления.

3.3. Сетевое подключение Ethernet

Изолированные коммуникационные интерфейсы NET 1 и NET 2 позволяют подключаться к сети Ethernet, однако для связи с MasterTool IEC XE наиболее подходит интерфейс NET 1.

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

3.3.1. IP Адрес

Ethernet-интерфейс NET 1 используется для связи по сети Ethernet и для конфигурирования процессора, 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 20: Конфигурация параметров по умолчанию для интерфейса Ethernet NET 1

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

Первоначально интерфейс NET 1 должен быть подключен к сети ПК с такой 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 Ethernet](#).

Ethernet-интерфейс NET 2 используется только для связи по сети Ethernet и поставляется со следующей конфигурацией параметров по умолчанию:

	NET 2
IP Адрес	192.168.16.1
Маска подсети	255.255.255.0
Адрес шлюза	192.168.16.253

Таблица 21: Конфигурация параметров по умолчанию для интерфейса Ethernet NET 2

3. УСТАНОВКА

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

Параметры сети интерфейса NET 2 могут быть изменены с помощью программы MasterTool IEC XE. Более подробная информация о конфигурации и изменении параметров приведена в разделе [Конфигурация интерфейсов Ethernet](#).

3.3.2. Безвозмездный ARP

Ethernet-интерфейс NETx оперативно рассылает пакеты ARP типа broadcast, сообщая свой IP- и MAC-адрес всем устройствам, подключенным к сети. Эти пакеты отправляются во время загрузки нового приложения программой MasterTool IEC XE и при запуске процессора, когда приложение переходит в режим Run.

Пять ARP-команд запускаются в течение начального интервала 200 мс, с последующим удвоением интервала с каждой новой запущенной командой, что в сумме составляет 3 с. Пример: первый запуск происходит в момент времени 0, второй - через 200 мс, третий - через 600 мс и так далее до пятого запуска в момент времени 3 с.

3.3.3. Прокладка сетевого кабеля

Nexto Ethernet-порты процессоров серии Nexto, обозначенные на панели буквой NET, имеют стандартную распиновку, аналогичную используемой в персональных компьютерах. Тип разъема, тип кабеля, физический уровень, а также другие детали, касающиеся процессора и сетевого устройства Ethernet, определены в разделе "Интерфейсы Ethernet".

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

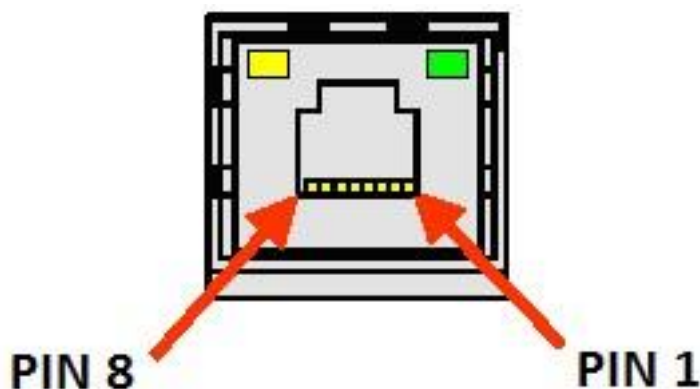


Рисунок 6: Разъем RJ45

Разъем	Сигнал	Описание
1	TXD +	Передача данных, позитив
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 22: Распиновка гнезда RJ45 - 10BASE-TE и 100BASE-TX

3. УСТАНОВКА

Интерфейс может быть подключен в коммуникационную сеть через концентратор или коммутатор, либо непосредственно от коммуникационного оборудования. В последнем случае, благодаря функции автоматического кроссовера процессоров Nexto, отпадает необходимость в использовании перекрестного сетевого кабеля, который используется для соединения двух ПК "точка-точка" через порт Ethernet.

Важно подчеркнуть, что под сетевым кабелем понимается пара гнезд RJ45, соединенных кабелем UTP или ScTP, категории 5, как прямым, так и перекрестным. Он используется для связи двух устройств через порт Ethernet.

Обычно такие кабели оснащены замком соединения, который гарантирует идеальное соединение между интерфейсным гнездом и гнездом кабеля. В момент установки необходимо вставить штекер в гнездо модуля до щелчка, обеспечивающего срабатывание замка. Чтобы отсоединить кабель от модуля, необходимо с помощью рычага замка отсоединить один от другого.

3.4. Последовательное сетевое подключение RS-232

Неизолированный коммуникационный интерфейс COM 1 позволяет подключаться к сети RS-232C. Ниже представлен разъем DB9 для подключения к процессору Nexto, с идентификацией и описанием признаков.

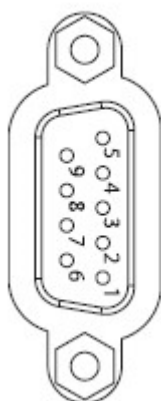


Рисунок 7: Разъем DB9

Разъем	Знак	Описание
1	DCD	Обнаружение несущей данных
2	TXD	Передача данных
3	RXD	Прием данных
4	-	Не используется
5	GND	Заземление
6	-	Не используется
7	CTS	Очистить для отправки
8	RTS	Запрос на отправку
9	-	Не используется

Таблица 23: Расположение контактов разъема COM 1 DB9 гнездо

3.4.1. Связь по интерфейсу RS-232C

Для подключения к устройству RS-232C используйте соответствующий кабель, как указано в разделе ["Сопутствующие товары"](#).

3.5. Последовательное сетевое соединение RS-485/422

Изолированный интерфейс связи COM 2 позволяет подключаться к сети RS-485/422. Ниже представлен разъем DB9 для подключения к процессору Nexto CPU, с идентификацией и описанием признаков.

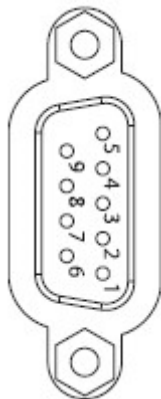


Рисунок 8: Разъем DB9

Разъем	Знак	Описание
1	-	Не используется
2	Term+	Внутренняя заделка, положительный
3	TXD+	Передача данных, положительный
4	RXD+	Прием данных, положительный
5	GND	Отрицательный опорный сигнал для внешней клеммы
6	+5V	Положительный опорный сигнал для внешней клеммы
7	Term-	Внутренняя клемма, отрицательная
8	TXD-	Передача данных, отрицательный
9	RXD-	Прием данных, отрицательный

Таблица 24: COM 1 (NX3004/NX3005) и COM 2 (NX3010/NX3020/NX3030)

3.5.1. RS-485 Связь без оконечного устройства

Для подключения в сеть RS-485 без оконечного устройства идентифицированные клеммы кабеля AL-1763 должны быть подключены к соответствующим клеммам устройства, как показано в таблице ниже.

AL-1763 клеммы	Сигналы терминала устройства
0	Экран
1	Не подключено
2	D+
3	D+
4	Не подключено
5	Не подключено
6	Не подключено

3. УСТАНОВКА

7	D-
8	D-

Таблица 25: Соединения RS-485 без терминации

На приведенной ниже схеме показано, как следует подключать соединительные клеммы AL-1763 к клеммам устройства.

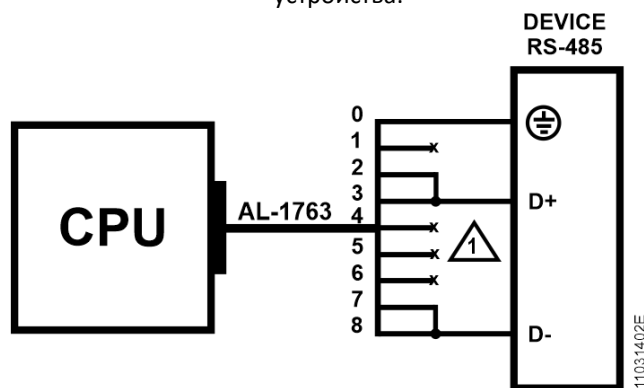


Рисунок 9: Схема подключения RS-485 без терминации

Примечание к схеме:

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

3.5.2. Связь RS-485 с внутренней заделкой

Для подключения в сеть RS-485 с использованием внутренней заделки необходимо подключить идентифицированные клеммы кабеля AL-1763 к соответствующим клеммам устройства, как показано в таблице ниже.

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

Таблица 26: Соединения RS-485 с внутренней заделкой

Примечание: Внутренняя заделка в открытом режиме имеет безопасный тип состояния.

На приведенной ниже схеме показано, как следует подключать соединительные клеммы AL-1763 к клеммам устройства.

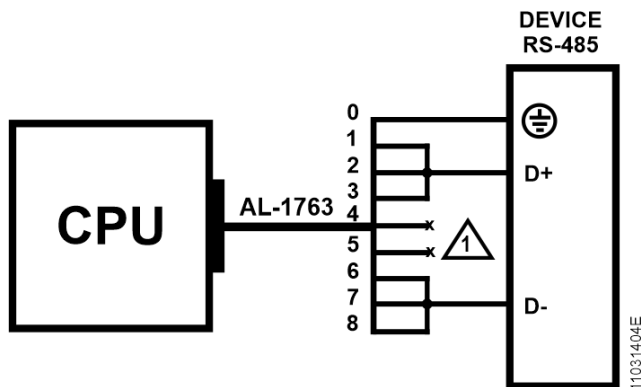


Рисунок 10: Подключения RS-485 с внутренней схемой заделки

Примечание к схеме:

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

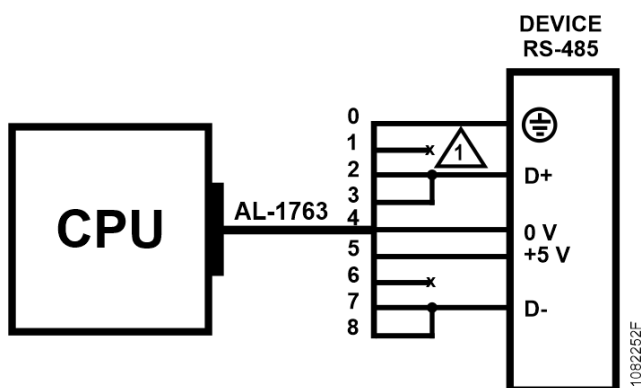
3.5.3. RS-485 Communication with External Termination

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

AL-1763 клеммы	Сигналы клемм процессора
0	Экран
1	Не подключен
2	D+
3	D+
4	0 V
5	+5 V
6	Не подключено
7	D-
8	D-

Таблица 27: Соединения RS-485 с внешней клеммой

На приведенной ниже схеме показано, как следует подключать соединительные клеммы AL-1763 к клеммам устройства.



3. УСТАНОВКА

Рисунок 11: Схема подключения RS-485 с внешней заделкой

Примечание к схеме:

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

3.5.4. Пример подключения сети RS-485 с внешней заделкой и резервированием ведущего устройства

На рисунке ниже показан пример сетевого соединения RS-485 с внешней заделкой, в котором в качестве ведущего устройства используются два процессора Nexto NX3030 с полукластерным резервированием.

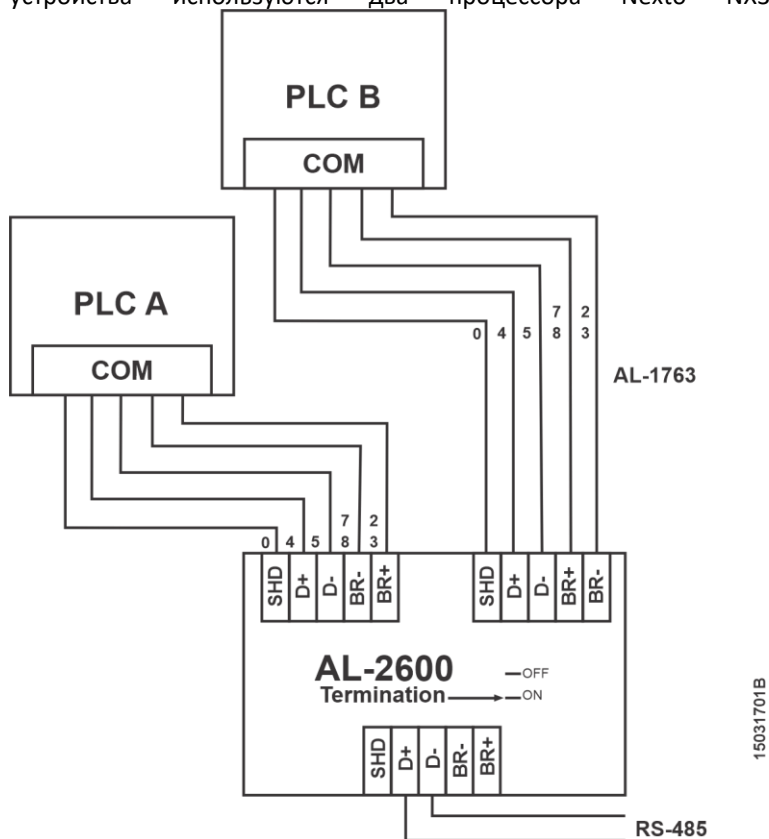


Рисунок 12: Схема подключения сети RS-485 с внешней заделкой и резервированием ведущего устройства

3.5.5. Связь по RS-422 без оконечного устройства

Для подключения в сеть RS-422 без оконечного устройства идентифицированные клеммы кабеля AL-1763 должны быть подключены к соответствующим клеммам устройства, как показано в таблице ниже.

AL-1763 клеммы	Сигналы клемм процессора
0	Экран
1	Не подключен
2	TX+
3	RX+
4	Не подключен
5	Не подключено

3. УСТАНОВКА

6	Не подключено
7	TX-
8	RX-

Таблица 28: Соединения RS-422 без оконечного устройства

На приведенной ниже схеме показано, как следует подключать соединительные клеммы AL-1763 к клеммам устройства.

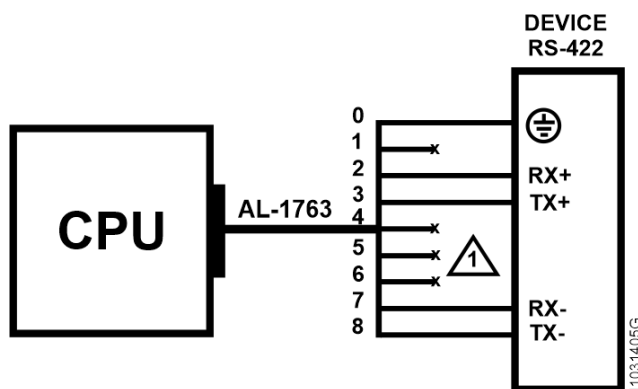


Рисунок 13: Соединения без схемы заделки

Примечание к схеме:

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

3.5.6. Связь RS-422 с внутренней заделкой

Для подключения в сеть RS-422 с использованием внутренней заделки необходимо подключить идентифицированные клеммы кабеля AL-1763 к соответствующим клеммам устройства, как показано в таблице ниже.

AL-1763 клеммы	Сигналы клемм процессора
0	Экран
1	TERM+
2	TX+
3	RX+
4	Не подключен
5	Не подключен
6	TERM-
7	TX-
8	RX-

Таблица 29: Соединения RS-422 с внутренней заделкой

Примечание: Имеющиеся внутренние клеммы имеют защищенное состояние в открытом режиме.

3. УСТАНОВКА

На приведенной ниже схеме показано, как следует подключать соединительные клеммы AL-1763 к клеммам устройства.

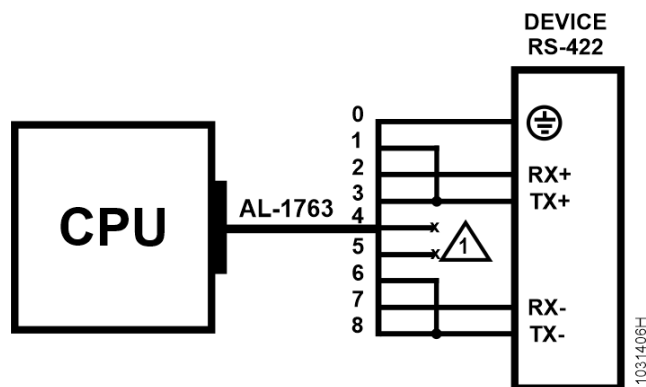


Рисунок 14: Соединения RS-422 со схемой заделки

Примечание к схеме:

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

3.5.7. Связь по интерфейсу RS-422 с внешней заделкой

Для подключения в сеть RS-422 с помощью интерфейсной внешней заделки необходимо подключить идентифицированные клеммы кабеля AL-1763 в соответствующие клеммы устройства, как показано в таблице ниже.

AL-1763 клеммы	Сигналы клемм процессора
0	Экран
1	Не подключен
2	TX+
3	RX+
4	0 V
5	+5 V
6	Не подключено
7	TX-
8	RX-

Таблица 30: Соединения RS-422 с внешней заделкой

На приведенной ниже схеме показано, как следует подключать соединительные клеммы AL-1763 к клеммам устройства.

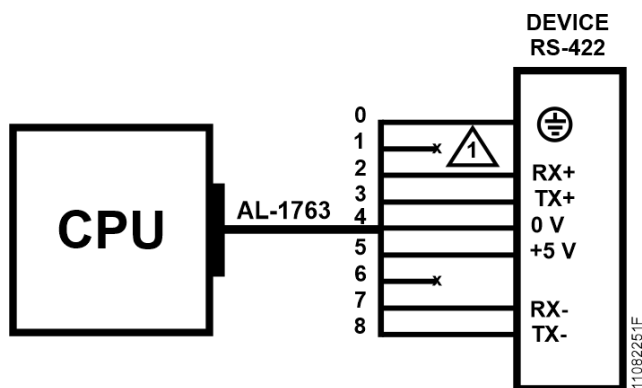


Рисунок 15: Схема подключения RS-422 с внешней заделкой

Примечание к схеме:

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

3.5.8. Пример сети RS-422

На рисунке ниже показан пример использования сети RS-422 с использованием процессора Nexto в качестве ведущего, ведомых устройств с интерфейсом RS-422 и решений Altus для терминаторов и соединений.

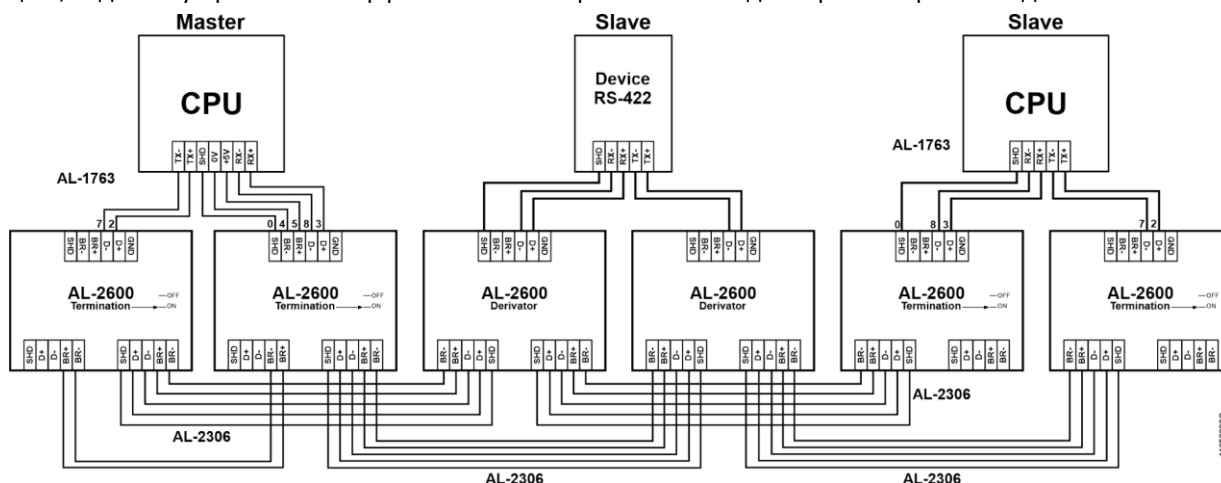


Рисунок 16: Пример сети RS-422

Примечание к схеме:

Модули AL-2600, находящиеся в конце сети, выполняют функцию терминаторов. В этом случае ключи AL-2600 должны быть сконфигурированы в режиме PROFIBUS Termination.

3.6. Установка карты памяти

В данном разделе описано, как вставить карту памяти в процессоры моделей серии Nexto. Более подробная информация приведена в разделе [Карта памяти](#).

Вначале необходимо обратить внимание на правильное положение карты памяти. Один ее угол отличается от трех других, и именно он должен служить ориентиром для правильной установки карты. Поэтому карту памяти следует вставлять в соответствии с рисунком на лицевой части процессора или так, как показано на рисунке ниже.

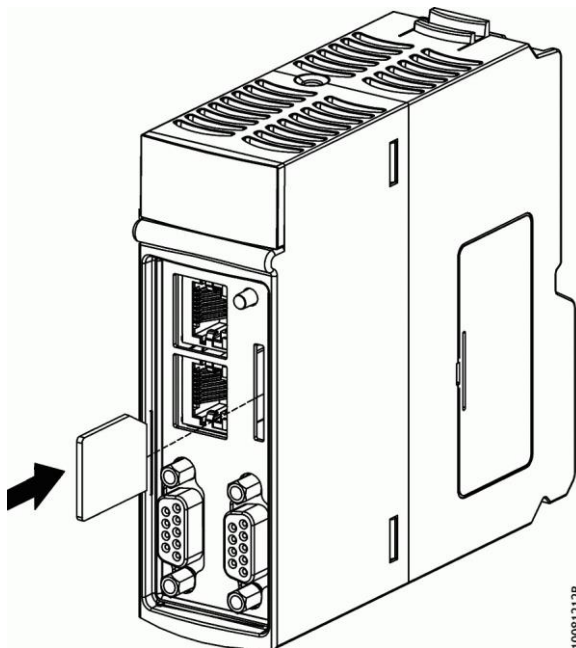


Рисунок 17: Установка карты памяти в процессор

Когда карта установлена правильно, на графическом дисплее процессора появляется символ. Для безопасного извлечения карты необходимо нажать клавишу MS, после чего произойдет небольшая задержка, и символ карты исчезнет с графического дисплея. Теперь карта готова к извлечению. Для этого необходимо прижать карту к процессору до щелчка, затем отпустить ее и извлечь из отсека, как показано на рисунке ниже. В этот момент карта будет свободна.

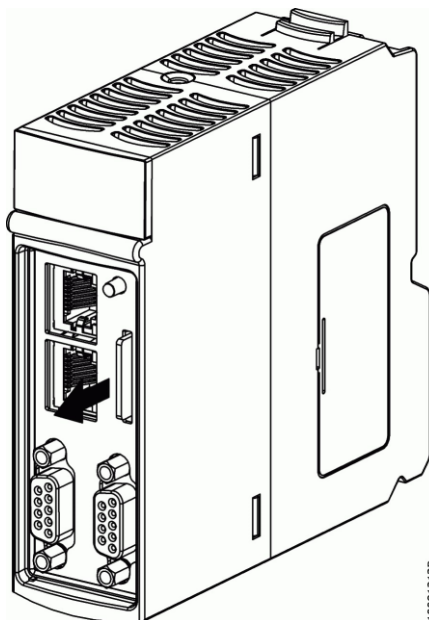


Рисунок 18: Извлечение карты памяти

3.7. Установка структуры

3.7.1. Установка модулей в стойку основной объединительной панели

3. УСТАНОВКА

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.

3.8. Установка программирующего устройства

Для выполнения установки ПО разработки MasterTool IEC XE необходимо иметь дистрибутивный CD-ROM или загрузить установочный файл с сайта http://www.altus.com.br/site_en/. Более подробная информация о пошаговой установке приведена в руководстве пользователя MasterTool IEC XE MT8500 - MU299609.

4. Начальное программирование

Основной целью данной главы является помощь в программировании и конфигурировании центральных процессоров серии Nexto, позволяющая пользователю сделать первые шаги перед началом программирования устройства.

В ЦПУ серии Nexto используется стандарт IEC 61131-3 для языков программирования, к которым относятся: IL, ST, LD, SFC и FBD, а также, кроме них, дополнительный язык - CFC. Эти языки можно разделить на текстовые и графические. IL и ST являются текстовыми языками и похожи на Ассемблер и Си соответственно. LD, SFC, FBD и CFC являются графическими языками. LD использует релейно-блочное представление и похож на релейные диаграммы. SFC использует представление диаграммы последовательности, что позволяет легко увидеть последовательность событий. FBD и CFC используют группу функциональных блоков, что позволяет наглядно увидеть функции, выполняемые каждым действием.

Программирование осуществляется через интерфейс разработки MasterTool IEC XE (IDE). MasterTool IEC XE позволяет использовать все шесть языков в одном проекте, таким образом, пользователь может применять лучшие возможности, предлагаемые каждым языком, что приводит к более эффективной разработке приложений, облегчает документирование и дальнейшее сопровождение.

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

4.1. Организация памяти и доступ к ней

В серии Nexto используется инновационная технология организации памяти и доступа к ней, называемая обратный порядок байтов, при которой старший байт хранится первым и всегда будет иметь наименьший адрес (например, %QB0 всегда будет более значимым, чем %QB1, как в таблице ниже, где для строки CPUNEXTO буква С - это байт 0, а буква О - байт 7).

Кроме того, обращение к памяти должно производиться аккуратно, так как переменные с большим количеством бит (WORD, DWORD, LONG) используют в качестве индекса старший байт, другими словами, в %QD4 старшим байтом всегда будет %QB4. Поэтому не нужно будет производить вычисления, чтобы узнать, какие DWORD соответствуют определенным байтам. В таблице, приведенной ниже, показана организация прямой и обратный порядок байтов.

старший бит ← прямой порядок байтов → младший бит								
БИТ	%QB7	%QB6	%QB5	%QB4	%QB3	%QB2	%QB1	%QB0
	С	Р	U	N	E	X	T	O
СЛОВО	%QW3		%QW2		%QW1		%QW0	
	CP		UN		EX		TO	
ДВОЙНОЕ	%QD1				%QD0			
	CPUN				EXTO			
ДЛИННОЕ	%QL0							
	CPUNEXTO							
старший бит ← обратный порядок байтов → младший бит								
БИТ	%QB0	%QB1	%QB2	%QB3	%QB4	%QB5	%QB6	%QB7
	С	Р	U	N	E	X	T	O
СЛОВО	%QW0		%QW2		%QW4		%QW6	
	CP		UN		EX		TO	
ДВОЙНОЕ	%QD0				%QD4			

	CPUN	EXTO
ДЛИННОЕ	%QL0	
	CPUNEXTO	

Таблица 31: 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						
↑		%QX1.0			%QD					%QD					
		%QX2.7			00				01		00				
		%QX2.6													
		%QX2.5													
		%QX2.4	%QB				%QB02								
		%QX2.3	02												
		%QX2.2													
		%QX2.1													
		%QX2.0		%QW				%QW			%QD				
		%QX3.7		02					02		01				
		%QX3.6													
		%QX3.5													
		%QX3.4	%QB				%QB03								
		%QX3.3	03												
		%QX3.2													
		%QX3.1													
		%QX3.0				%QL			%QW			%QD			
		%QX4.7				00			03			02			
		%QX4.6													
		%QX4.5													
		%QX4.4	%QB				%QB04								
		%QX4.3	04												
		%QX4.2													
		%QX4.1													
		%QX4.0		%QW				%QW				%QD			
		%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	
		%QX6.6													
		%QX6.5													
		%QX6.4	%QB				%QB06								
		%QX6.3	06												
		%QX6.2													
		%QX6.1													
		%QX6.0		%QW				%QW							
		%QX7.7		06					06						
		%QX7.6													
		%QX7.5													
		%QX7.4	%QB				%QB07								
		%QX7.3	07												
		%QX7.2													
		%QX7.1													
		%QX7.0													

Таблица 32: Организация памяти и доступ к ней

В приведенной таблице показана организация и доступ к памяти, иллюстрирующая значимость 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
- Базовый
- Нормальный
- Эксперт
- Пользовательский
- Профиль машины

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

В следующих разделах подробно описаны характеристики каждого профиля, которые имеют постепенный уклон в сторону усложнения. Исходя из этих определений, пользователю рекомендуется всегда использовать самый 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), однако разработчик должен сам 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 основной задачи. В том случае, если опция Включить обновление ввода/вывода для каждой задачи не может быть использована, присутствует, начиная с версии MasterTool IEC XE 2.01.

4.2.1. Одиночный

В профиле одиночного проекта приложение имеет только одну пользовательскую задачу - MainTask. Эта задача отвечает за выполнение одной программной единицы типа Program, называемой MainPrg. Эта единственная программа может вызывать другие программные единицы типа Program, Function или Function Block, но весь код будет выполняться только MainTask.

В данном профиле MainTask будет циклического типа (Cyclic) с приоритетом 13 (тринадцать) и будет выполнять исключительно программу MainPrg в непрерывном цикле. MainTask уже полностью определена, и разработчику необходимо создать программу MainPrg, используя любой из языков стандарта МЭК 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. Стандартный вариант MasterTool IEC XE - это использование стандартного проекта MasterTool Standard Project, связанного с профилем Одиночный, в который также входит MainPrg, созданный на языке, выбранном при создании проекта.

В этом случае не требуется рассматривать такие вопросы, как согласованность данных, совместное использование ресурсов или механизмы взаимного исключения.

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Задача	POU	Приоритет	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-

Таблица 33: Задача одного профиля

4.2.2. Базовый

В базовом профиле проекта приложение имеет одну пользовательскую задачу типа Непрерывный под названием MainTask, которая выполняет программу в непрерывном цикле (без определения времени цикла) с приоритетом, фиксированным в 13 (тринадцать). Эта задача отвечает за выполнение одного программного блока POU под названием MainPrg. Важно отметить, что время цикла может меняться в зависимости от количества используемых коммуникационных задач, так как в этом режиме основная задача прерывается коммуникационными задачами.

Данный профиль также допускает включение двух событийных задач с более высоким приоритетом, которые могут прервать (упредить) основную задачу в любой момент времени: задача с именем ExternInterruptTask00 - это событийная задача типа Внешний с приоритетом 02 (два); задача с именем TimeInterruptTask00 - это событийная задача типа Циклический с приоритетом 01 (один).

Модель шаблона проекта "Базовый" включает в себя уже полностью определенные три задачи, представленные в таблице ниже. Разработчику остается только создать соответствующие программы.

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

Таблица 34: Основные задачи профиля

4.2.3. Нормальный

В профиле проекта "Нормальный" приложение имеет одну пользовательскую задачу типа Циклический, называемую MainTask. Эта задача отвечает за выполнение единственной программной единицы POU под названием MainPrg. Эта программа и эта задача аналогичны единственной задаче и единственной программе профиля Single, но здесь в приложение могут быть интегрированы дополнительные пользовательские задачи. Эти дополнительные задачи, названные CyclicTask00 и CyclicTask01, отвечают за эксклюзивное выполнение 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 POU типа Program, Function и Function Block.

Кроме того, 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 с именем ExternInterruptTask00 - это событийная задача типа Внешний, выполнение которой инициируется каким-либо внешним событием, например, изменением управляющего сигнала на последовательном порту или изменением дискретного входа на шине NEXO. Приоритет этой задачи установлен в 02 (два), она отвечает исключительно за выполнение программы ExternInterruptPrg00. Задача под названием TimeInterruptTask00 является событийной задачей циклического типа с приоритетом 01 (один), отвечающей исключительно за выполнение программы TimeInterruptPrg00.

В модели "Нормальный проект" пять задач и их POU уже полностью определены, как показано в таблице ниже. Разработчику остается только реализовать содержание программ, выбрав на мастере любой из языков стандарта МЭК 61131-3. Интервал между задачами и триггерные события могут быть настроены разработчиком, а ненужные задачи могут быть исключены.

Tasks	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

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	20 мс	-
---------------------	--------------------	----	-------------	-------	---

Таблица 35: Задачи нормального профиля

4.2.4. Экспертный

Профиль проекта "Экспертный" включает 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. То есть 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 этом профиле, кроме стандартных программ, каждая задача может выполнять дополнительные программы.

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

Существует восемь уже полностью определенных задач, как показано в таблице ниже, а также связанные с ними программы на выбранном языке. Интервалы и триггеры любой задачи, а также приоритеты событийных задач могут быть настроены пользователем.

При разработке приложения с использованием профиля экспертного проекта необходимо особо внимательно отнестись к масштабированию событийных задач. Если между этими задачами или между ними и базовыми задачами происходит обмен информацией и ресурсами, то настоятельно рекомендуется использовать стратегии, обеспечивающие согласованность данных.

Задачи	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 36: Задачи экспертного профиля

4.2.5. Пользовательский

Профиль проекта "Пользовательский" позволяет разработчику использовать все возможности 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 с этим именем в данном профиле.

Помимо задач реального времени с приоритетом от 00 до 15, которые планируются по приоритету, в этом профиле можно определить задачи с более низкими приоритетами в диапазоне от 16 до 31. В этом диапазоне используется Completely Fair Scheduler (разделение времени), который необходим для выполнения кодов, которые могут быть заблокированы (например, использование сокетов).

Разработчик волен частично следовать или не следовать организации, определенной в других профилях проекта, в зависимости от особенностей приложения. С другой стороны, модель "Пользовательская", связанная с этим профилем, не нуждается в предопределении таких элементов, как задача, программа или параметр, оставляя разработчику возможность самому создавать все элементы, из которых состоит приложение. Однако пользователь может генерировать те же элементы, которые доступны для профиля Экспертный.

Задачи	POU	Приоритет	Тип	Интервал	Событие
--------	-----	-----------	-----	----------	---------

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

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 37: Задачи пользовательского профиля

4.2.6. Профиль машины

По умолчанию в машинном профиле приложение имеет пользовательскую задачу типа Cyclic под названием MainTask. Эта задача отвечает за реализацию единственного POU типа Program, называемого MainPrg. Эта программа может вызывать другие программные блоки типа Program, Function или Function Block, 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. Такая оптимизация возможна потому, что MainTask также выполняет обработку шины. Таким образом, в отличие от других профилей, машинный профиль не требует переключения контекста для обработки шины, что сокращает общее время обработки.

Кроме того, данный профиль может включать задачу прерывания, называемую TimeInterruptTask00, которая имеет более высокий приоритет, чем MainTask, и, следовательно, может прервать ее выполнение в любой момент.

Задачи	POU	Приоритет	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	4 мс	-

Таблица 38: Задачи профиля машины

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

4.2.7. Общая таблица

Проверки		Профили проектов					
		Одиночный	Машинный	Базовый	Нормальный	Экспертный	Пользовательский
Всего задач	Количество	01	04	[01..03]	[01..32]	[01..32]	[01..32]
Программы на одну задачу	Количество	01		01	01	<n>	<n>
Основная задача	Название						
	Тип	Циклический	Циклический	Внешний	Циклический	Циклический	Циклический
	Приоритет	13	13	13	13	13	13
	Количество	01	01	01	01	01	01
Задача временного прерывания	Название						
	Тип		Циклический	Циклический	Циклический	Циклический	Циклический

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

	Приоритет		01	01	01	01 or [08..12]	01 or [08..12]
	Количество		[00..01]	[00..01]	[00..01]	[00..31]	[00..31]
Внешняя задача прерывания	Название						
	Тип		Внешний	Внешний	Внешний	Внешний	Внешний
	Приоритет		02	02	02	02 or [08..12]	02 or [08..12]
	Количество		[00..01]	[00..01]	[00..01]	[00..31]	[00..31]
Циклическая задача	Название				CyclicTask<nn>	CyclicTask<nn>	CyclicTask<n>
	Тип				Циклический	Циклический	Циклический
	Приоритет				13	13	13
	Количество				[00..31]	[00..31]	[00..31]
Свободная задача	Название					FreeTask	FreeTask
	Тип					Свободный ход	Свободный ход
	Приоритет					31	31
	Количество					[00..01]	[00..01]
Событийная задача	Название						<n>
	Тип						Событие
	Приоритет						<n>
	Количество						[00..31]

Таблица 39: Общая таблица 'Профиль x Задачи'

ВНИМАНИЕ

Предлагаемые имена POU, связанные с задачами, не являются постоянными. Они могут быть изменены, если они также изменены в конфигурациях задач.

4.2.8. Максимальное количество задач

Максимальное количество задач, которые может создать пользователь, определено только для профиля "Пользовательский", единственного, который имеет такое право. В остальных профилях задачи уже созданы и настроены. Однако создаваемые задачи должны использовать следующие префиксы, в соответствии с типом каждой из них: CyclicTaskxx, TimeInterruptTaskxx, ExternInterruptTaskxx, где xx - номер создаваемой задачи.

В таблице ниже описано максимальное количество задач IEC для каждого процессора и профиля проекта, при этом экземпляры протоколов также рассматриваются процессором как коммуникационные задачи.

	Тип задачи	NX3030					
		S	B	N	E	P	M
Задача конфигурирования (Задача ВСБ)	Циклический	1	1	1	1	1	0
Пользовательские задачи	Циклическая	1	1	31	31	31	2

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

	Срабатывание по событию	0	0	0	0	31	0
	Дисп. Внешнее событие	0	1	0	30	31	0
	Свободное движение	0	1	0	1	31	0
	Срабатывание по состоянию	0	0	0	0	31	0
NET - клиентские или серверные экземпляры	Циклический	16					
COM (n) - ведущие или ведомые экземпляры	Циклический	1					
ВСЕГО		32					

Таблица 40: Задачи NX3030 IEC Максимальное количество

Примечания:

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

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

Задача WHSB: WHSB - это системная задача, которую необходимо учитывать, чтобы не превысить общее значение.

NETs - клиентские или серверные экземпляры: Определенное максимальное значение учитывает все системные Ethernet-интерфейсы, включая модули расширения, если они применяются. Например, экземпляры протокола MODBUS.

COM (n) - Master или Slave экземпляры: "n" означает номер последовательного интерфейса. Даже при использовании модулей расширения значение таблицы будет максимальным для каждого интерфейса. Например, экземпляры протокола MODBUS.

Total: Общее значение представляет собой не сумму всех задач профиля, а максимальное значение, допустимое для одного процессора. Таким образом, пользователь может создать несколько типов задач, при этом установленные числа для каждой из них и общее значение не будут превышены.

4.3. Конфигурация процессора

Конфигурация процессора Nexto находится в дереве устройств, как показано на рисунке ниже, и доступна двойным щелчком мыши на соответствующем объекте. На этой вкладке можно настроить область диагностики, область ретентивной и постоянной памяти, режим горячей замены, а также другие параметры, описанные в

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

разделе [Конфигурация процессора](#).

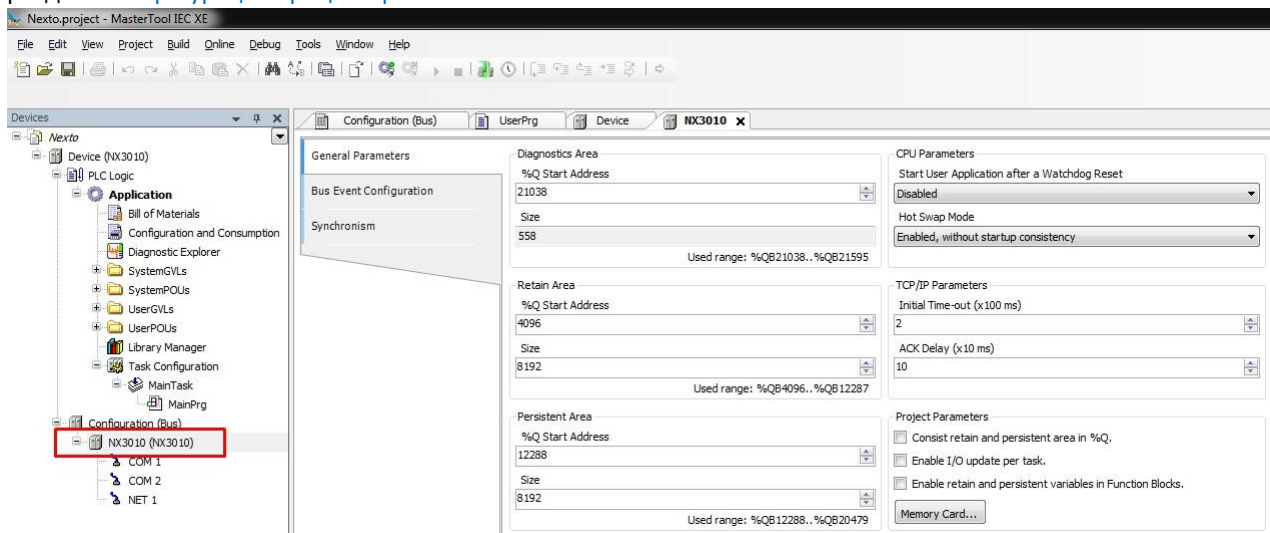


Рисунок 19: Конфигурация процессора

Кроме того, дважды щелкнув на пиктограмме NET 1 процессора, можно настроить интерфейс Ethernet, который будет использоваться для связи между контроллером и программой MasterTool IEC XE.

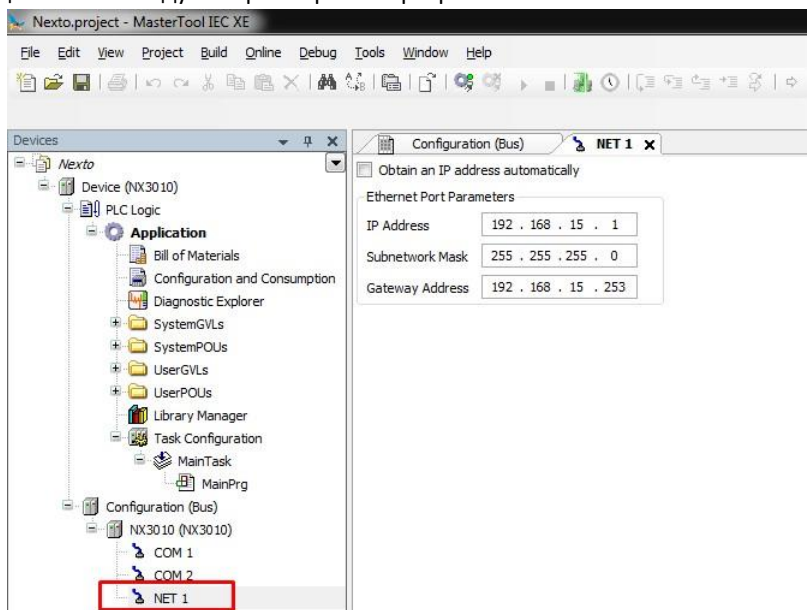


Рисунок 20: Настройка коммуникационного порта процессора

Конфигурация, заданная на этой 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 эти библиотеки должны быть вставлены в проект, чтобы их использование стало возможным. Порядок вставки и более подробную информацию о доступных библиотеках можно найти в руководстве по программированию MasterTool - MP399609.

4.5. Вставка экземпляра протокола

Процессоры серии Nexto, как описано в разделе "Протоколы", предлагают несколько протоколов связи. За исключением протоколов OPC DA и OPC UA, которые имеют другую процедуру конфигурирования, вставка протокола может быть выполнена простым щелчком правой кнопки мыши на нужном интерфейсе связи, выбором добавления устройства и последующим выполнением конфигурации, как показано в разделе "Конфигурация протоколов". Ниже приведены примеры.

4.5.1. MODBUS Ethernet

Первым шагом для конфигурирования MODBUS Ethernet (в данном примере - Client) является включение экземпляра в нужную NET (в данном случае NET 1, так как CPU NX3010 имеет только один интерфейс Ethernet). Щелкните на NET правой кнопкой мыши и выберите Add Device..., как показано на рисунке ниже.

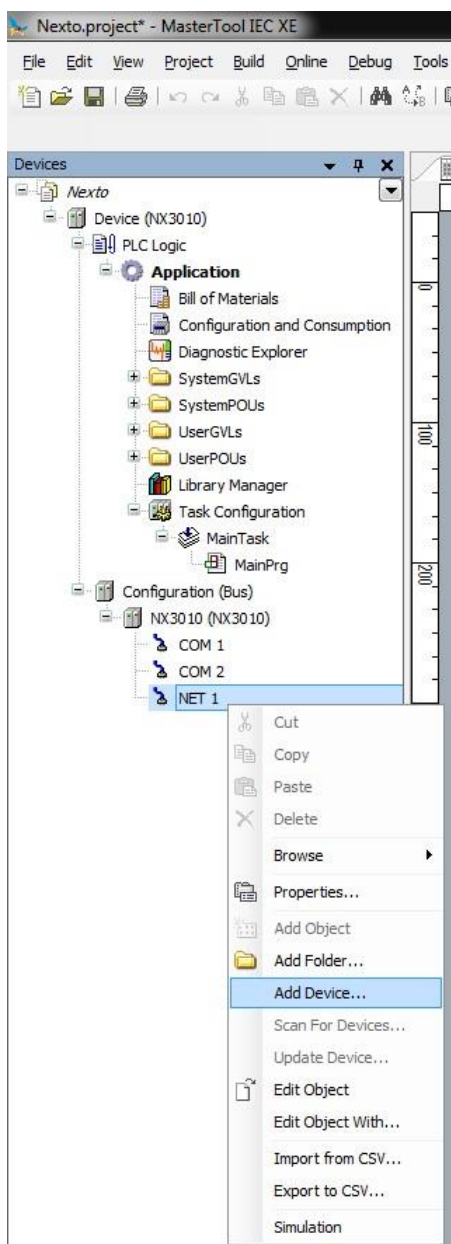


Рисунок 21: Добавление экземпляра

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

After После этого на экране появятся доступные пользователю протоколы. В этом меню определяется режим конфигурирования протокола. Выбирается опция MODBUS Symbol Client, для настройки символьного отображения, или MODBUS Client, для прямой адресации (%Q). Затем нажмите кнопку Add Device (Добавить устройство), как показано на рисунке ниже.

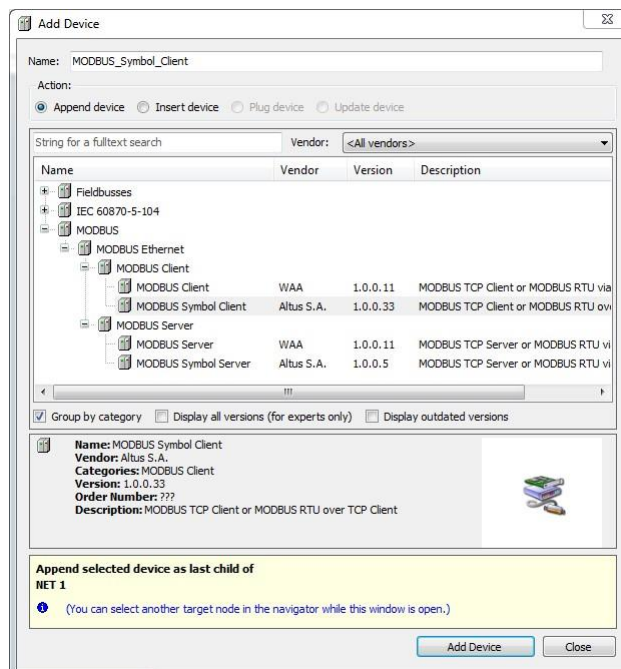


Рисунок 22: Выбор протокола

4.6. Поиск устройства

Для установления связи между процессором и MasterTool IEC XE сначала необходимо найти и выбрать нужное устройство. Конфигурация этой связи находится на объекте Device в дереве устройств, на вкладке Communication Settings. На этой вкладке, после выбора шлюза и нажатия на кнопку Scan network, программа MasterTool IEC XE выполняет поиск устройств и показывает ЦПУ, найденные в сети интерфейса Ethernet станции, на которой запущен инструмент.

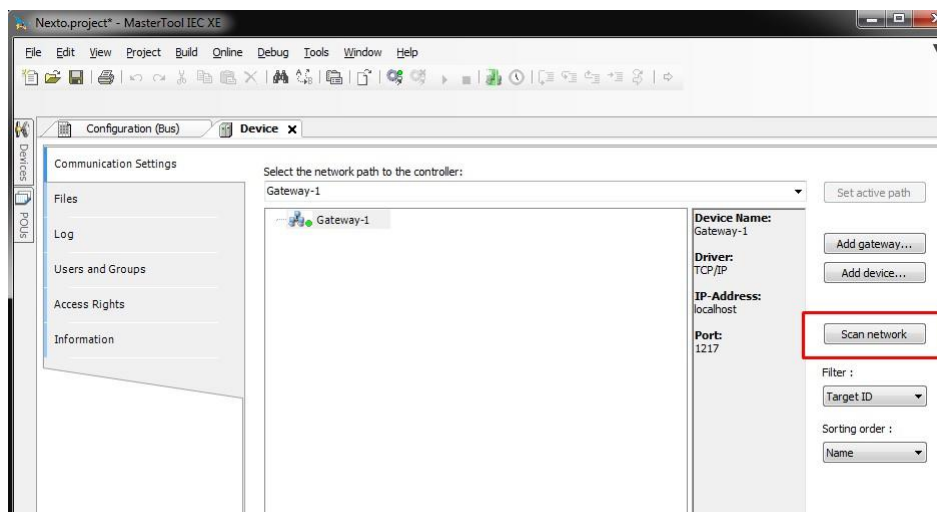


Рисунок 23: Поиск процессора

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Если шлюз ранее не был настроен, его можно включить с помощью кнопки 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 Set active path. Это действие позволяет выбрать контроллер и сообщить программе конфигурирования, какой контроллер будет использоваться для связи и отправки проекта.

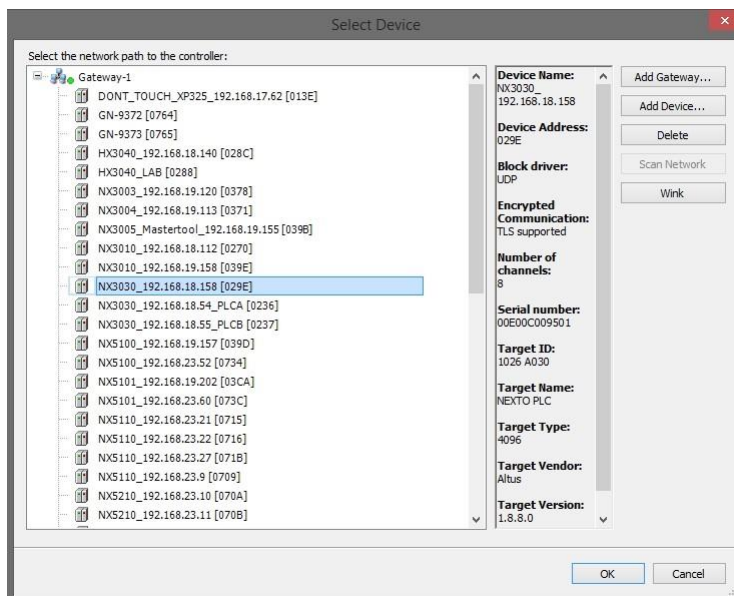


Рисунок 24: Выбор процессора

Кроме того, пользователь может изменить имя устройства, отображаемого по умолчанию. Для этого необходимо щелкнуть правой кнопкой мыши на нужном устройстве и выбрать пункт Change Device Name. После изменения имени устройство ни при каких обстоятельствах не вернется к имени по умолчанию.

В случае если Ethernet-конфигурация подключаемого процессора находится в сети, отличной от Ethernet-интерфейса станции, программа MasterTool IEC XE не сможет найти устройство. В этом случае рекомендуется использовать команду Easy Connection, расположенную в меню Online.

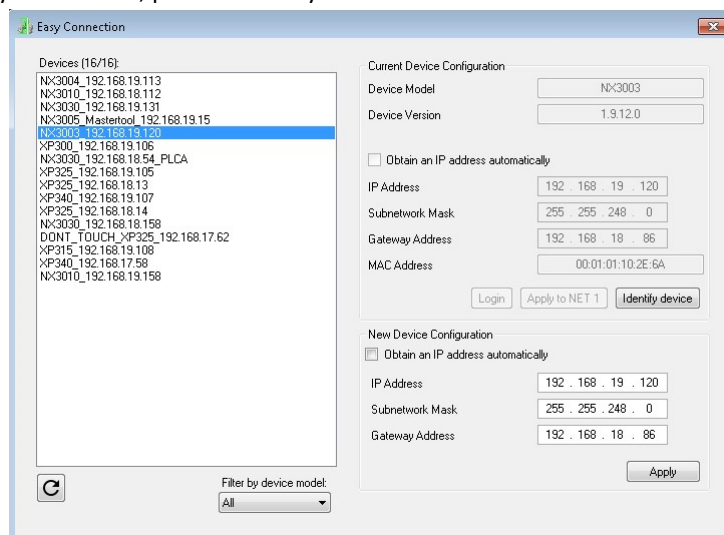


Рисунок 25: Простое подключение

Данная команда осуществляет обмен данными с устройством на уровне MAC, позволяя постоянно изменять конфигурацию Ethernet-интерфейса процессора независимо от IP-конфигурации станции и от той, которая была ранее настроена на устройстве. Таким образом, с помощью этой команды можно изменить конфигурацию устройства, поместив его в ту же сеть, что и Ethernet-интерфейс станции, на которой запущен MasterTool IEC XE, что позволяет

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

найти и выбрать устройство для связи. Полное описание команды 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. Для этого достаточно выбрать команду Login, расположенную в меню Online программы MasterTool IEC XE, как показано на следующем рисунке. Данная операция может занять несколько секунд, в зависимости от размера сгенерированного файла.

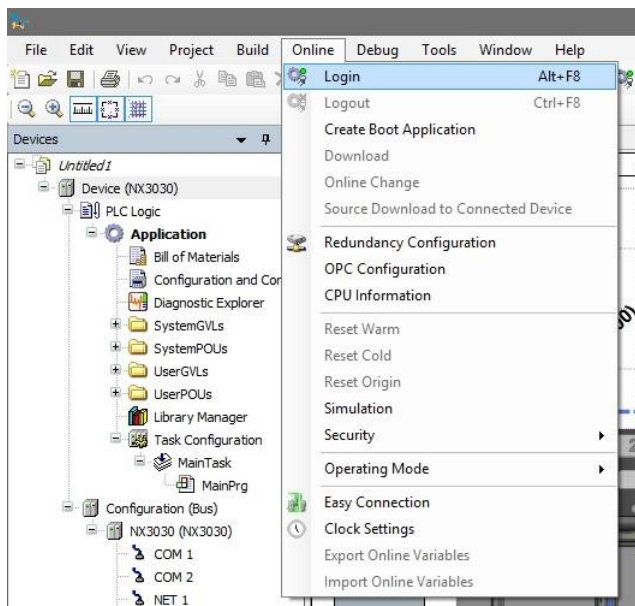


Рисунок 26: Отправка проекта на процессор

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

Если конфигурация Ethernet проекта отличается от конфигурации устройства, то связь может прерваться в конце процесса загрузки, когда новая конфигурация будет применена на устройстве. Поэтому будет выдано следующее предупреждающее сообщение, предлагающее пользователю продолжить или отказаться от этой операции.

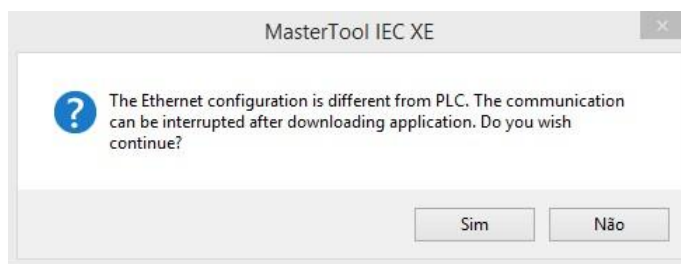


Рисунок 27: Предупреждение о конфигурации IP

Если на процессоре нет приложения, то будет выдано следующее сообщение.



Рисунок 28: Отсутствие приложения на устройстве

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

- Вход с изменением в режиме онлайн: выполнить вход и отправить новый проект без остановки текущего приложения ЦПУ (см. пункт Режим работы), обновляя изменения при выполнении нового цикла.
- Вход с загрузкой: выполнить вход и отправить новый проект при остановленном ЦПУ (см. пункт "Режим остановки"). При запуске приложения обновление уже будет выполнено.

Вход без изменений: выполняет вход в систему без отправки нового проекта.

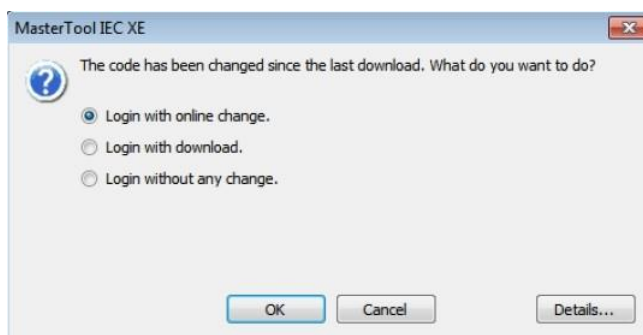


Рисунок 29: Обновление проекта на центральном процессоре

ВНИМАНИЕ

В интерактивных изменениях не разрешается связывать отображение символьных переменных из глобального списка переменных (GVL) и использовать эти переменные в другом глобальном списке переменных (GVL).

Если новое приложение содержит изменения в конфигурации, то изменение в режиме онлайн будет невозможно. В этом случае MasterTool IEC XE запрашивает, следует ли выполнить вход в систему как загрузку (остановив приложение) или отменить операцию.

Примечание: Кнопка Подробности... показывает изменения, внесенные в приложение.

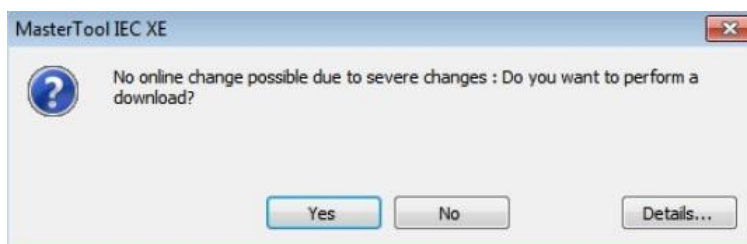


Рисунок 30: Изменения конфигурации

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

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

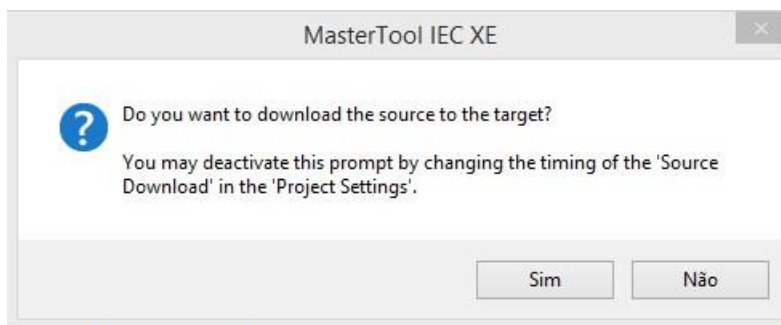


Рисунок 31: Загрузка исходного кода

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

4.8. Режим работы

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

Для выполнения этой команды достаточно зайти в меню Debug и выбрать опцию Start, как показано на рисунке ниже.

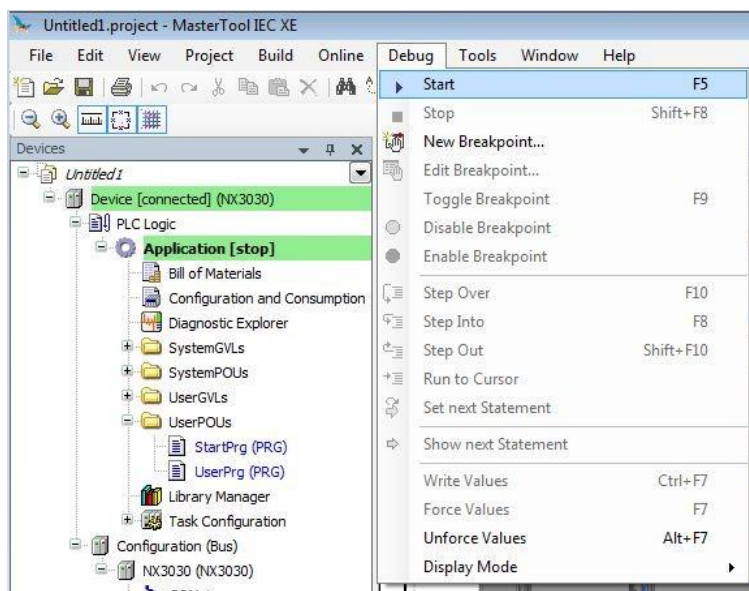


Рисунок 32: Запуск приложения

На рисунке ниже показано приложение в процессе выполнения. При выборе вкладки POU созданные переменные отображаются в окне мониторинга, в котором их значения могут быть визуализированы и принудительно изменены пользователем.

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

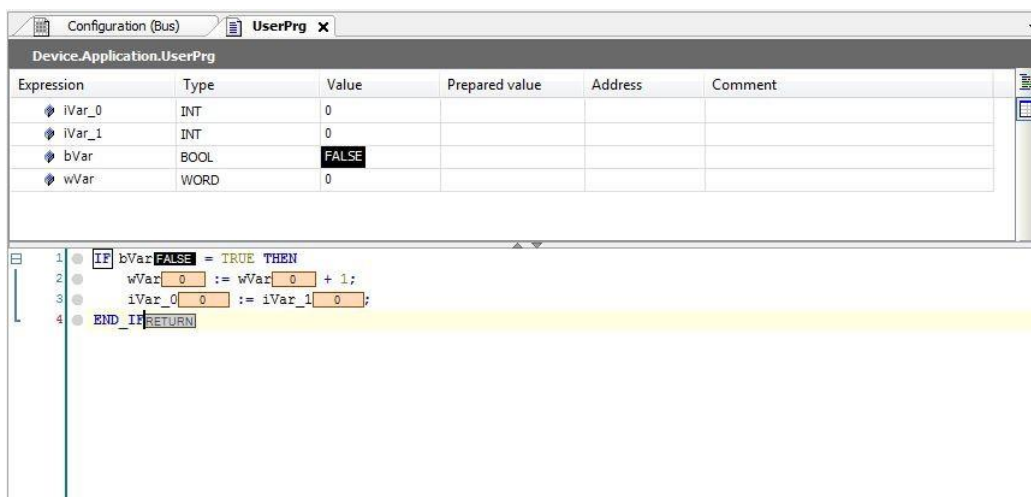


Рисунок 33: Выполнение программы

Если процессор уже имеет встроенное загрузочное приложение, то при включении питания он автоматически переходит в режим Run Mode, без необходимости подачи интерактивной команды через MasterTool IEC XE.

4.9.Режим остановки

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

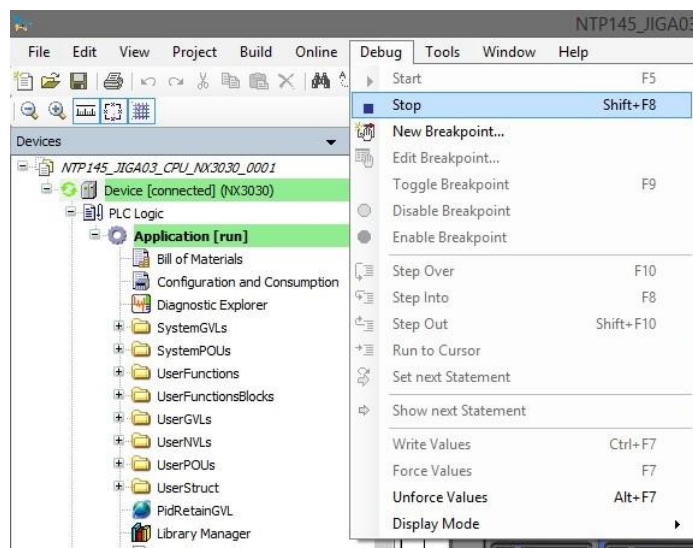


Рисунок 34: Остановка приложения

В случае инициализации процессора без сохраненного приложения он автоматически переходит в режим Stop Mode, как это происходит при возникновении программного исключения.

4.10.Запись и принудительное использование переменных

После входа в ПЛК пользователь может записать или принудительно ввести значения в переменную проекта.

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Команда записи (CTRL + F7) записывает значение в переменную, и это значение может быть перезаписано инструкциями, выполняемыми в приложении.

Кроме того, команда принудительной записи (F7) записывает значение в переменную, не позволяя изменить это значение до тех пор, пока не будет освобождена принудительная запись переменных.

Важно отметить, что при использовании MODBUS RTU Slave и MODBUS Ethernet Server, когда не выбрана опция Read-only из настроенных отношений, команда принудительной записи (F7) должна выполняться над доступными переменными в окне мониторинга, так как команда записи (CTRL + F7) оставляет переменные перезаписанными при новых считываниях.

ВНИМАНИЕ

Форсирование переменных может быть выполнено в режиме Online.
Диагностические переменные не могут быть принудительно введены, а только записаны, так как диагностика осуществляется процессором и будет им перезаписана.

При принудительной записи в резервную переменную активного ПЛК основная задача приложения будет испытывать влияние на время ее выполнения как в активном, так и в резервном ПЛК. Это связано с тем, что оба полукластера будут обмениваться информацией о принудительных переменных каждый цикл. Поэтому при форсировании переменных в резервированной системе необходимо учитывать возможное дополнение к выполнению задачи. В приведенной ниже таблице показано, насколько в среднем увеличится время выполнения MainTask в этом случае:

Время выполнения	Действующий СР			Резервный СР		
	50 мс	100 мс	200 мс	50 мс	100 мс	200 мс
Сложение с 10 силами	2,4 %	2,2 %	1,7 %	4,0 %	3,4 %	2,0 %
Добавление с 50 силами	12,0 %	9,2 %	6,0 %	18,0 %	12,0 %	8,0 %
Сложение со 128 силами	26,0 %	21,0 %	16,0 %	56,0 %	34,0 %	22,5 %

Таблица 41: Влияние форсирующих переменных на резервный СР

ВНИМАНИЕ

Если процессор имеет принудительные переменные и обесточен, то при следующей инициализации переменные потеряют принудительность.
Предельное число принудительных переменных для процессоров Nexto составляет 128 переменных, независимо от модели и конфигурации используемого процессора.

4.11. Выход из системы

Для завершения онлайн-связи с процессором необходимо выполнить команду Выйти, расположенную в меню Online, как показано на рисунке ниже.

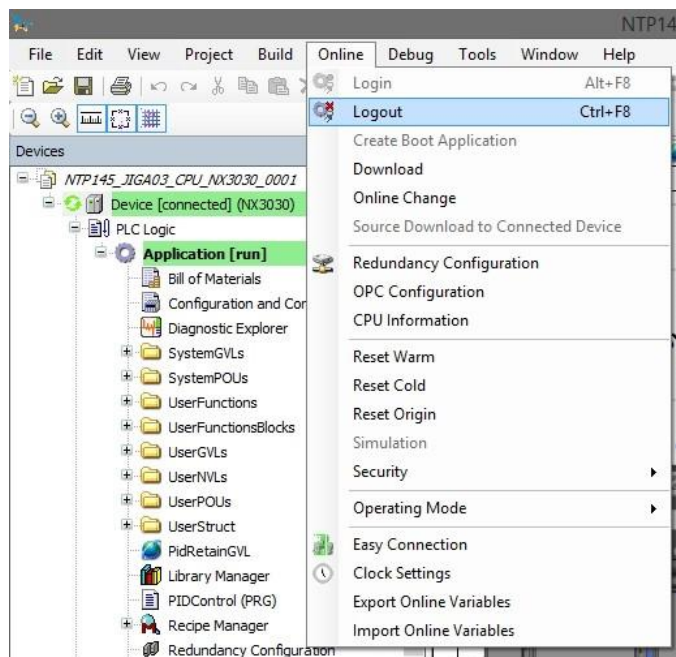


Рисунок 35: Завершение интерактивной связи с процессором

4.12. Загрузка проекта

Процессоры серии Nexto способны хранить исходный код приложения во внутренней памяти устройства, что позволяет в дальнейшем извлекать (загружать) полный проект и модифицировать приложение.

Для восстановления проекта, ранее сохраненного во внутренней памяти процессора, необходимо выполнить команду, расположенную в меню File, как показано на следующем рисунке.

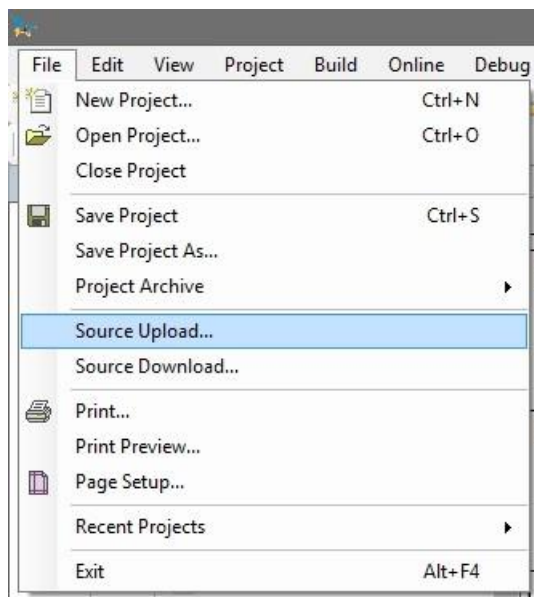


Рисунок 36: Опция загрузки проекта

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

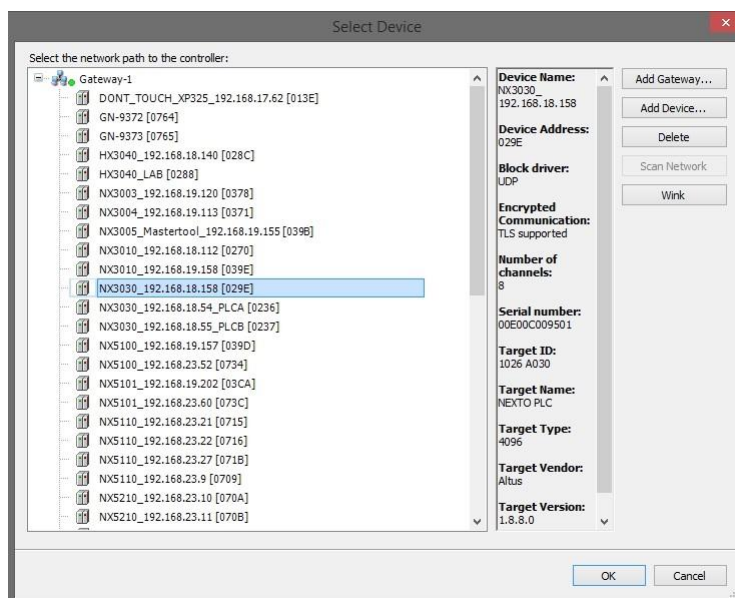


Рисунок 37: Выбор процессора

Чтобы убедиться, что проект, загруженный в процессор, идентичен и может быть доступен на других рабочих станциях, обратитесь к главе "Метод загрузки/входа в систему без различий проектов" в руководстве пользователя MasterTool IEC XE MT8500 - MU299609.

ВНИМАНИЕ

Область памяти для хранения проекта в процессорах Nexto определяется в разделе [Память](#).

ВНИМАНИЕ

При выгрузке 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 Выгрузка не удастся.

4.13. Рабочие состояния процессора

4.13.1. Работа

Когда процессор находится в режиме работы, выполняются все прикладные задачи.

4.13.2. Остановка

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

Когда процессор переходит в режим остановки из-за загрузки приложения, переменные в прикладных задачах будут потеряны, за исключением переменных типа "постоянный".

4.13.3. Точка останова

При достижении отладочной метки в задаче она прерывается. При этом все активные задачи в приложении не прерываются, продолжая свое выполнение. Благодаря этой возможности можно пошагово в режиме Online исследовать поток программы по позициям прерываний, включенных через редактор.

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Более подробную информацию об использовании точек останова можно найти в Руководстве по использованию MasterTool IEC XE - MU299609.

4.13.4. Исключение

Когда процессор находится в состоянии Исключения, это означает, что в одной из активных задач приложения произошла некорректная операция. Задача, вызвавшая Исключение, приостанавливается, а остальные задачи переходят в режим Стоп. Вывести задачи из этого состояния и вновь запустить их в выполнение можно только после нового запуска процессора. Таким образом, только сброс тепла, сброс холода, сброс исходного состояния или перезапуск процессора переводит приложение в режим работы.

4.13.5. Сброс тепла

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

4.13.6. Сброс холода

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

4.13.7. Сброс исходного состояния

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

Примечания:

Сброс: При выполнении команды Сброс точки останова, определенные в приложении, отключаются.

Команда: Для выполнения команд Сброс Тепла, Холода или Исходного состояния необходимо, чтобы MasterTool IEC XE находился в режиме Online с процессором.

4.13.8. Команда сброса процесса (IEC 60870-5-104)

Эта команда сброса процесса может запрашиваться клиентами МЭК 60870-5-104. После ответа клиента центральный процессор начинает процесс перезагрузки, как если бы выполнялся цикл подачи напряжения.

В случае резервирования ПЛК команда сброса процесса синхронизируется с неактивным ПЛК, что приводит к перезагрузке обоих ПЛК.

Стандарт IEC 60870-5-104 предусматривает передачу квалификационного значения (0..255) с командой сброса процесса, однако этот "параметр" не учитывается процессором.

4.14. Программный компонент (POU) и глобальные списки переменных (GVL)

Проект, созданный MasterTool IEC XE, содержит набор программных модулей (POU) и списки глобальных переменных, что призвано облегчить программирование и использование контроллера. В следующих разделах описаны основные элементы, входящие в эту стандартную структуру проекта.

4.14.1. MainPrg программа

Задача MainTask связана с одним уникальным POU программного типа с именем MainPrg. Программа MainPrg создается автоматически и не может быть отредактирована пользователем.

Код программы MainPrg на языке ST выглядит следующим образом:

```
(*Основной POU, связанный с MainTask, который вызывает StartPrg, UserPrg/ActivePrg и  
NonSkippedPrg.  
Этот POU заблокирован для редактирования.*)
```

```
PROGRAM MainPrg  
VAR isFirstCycle : BOOL := TRUE;  
END_VAR  
  
SpecialVariablesPrg();  
IF isFirstCycle THEN StartPrg();  
    isFirstCycle := FALSE;  
ELSE  
    UserPrg();  
END_IF;
```

MainPrg вызывает два других POU программного типа, названных StartPrg и UserPrg. В то время как UserPrg вызывается всегда, StartPrg вызывается только один раз при запуске приложения ПЛК.

В противоположность программе MainPrg, которая не подлежит модификации, пользователь может изменять 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 этом POU пользователь может создавать логики, циклы, стартовые переменные и т.д., которые будут выполняться только один раз в первом цикле ПЛК, перед первым выполнением POU UserPrg. И больше не будут вызываться в ходе выполнения проекта.

В случае загрузки нового приложения, выключения питания ПЛК, а также при сбросе исходного состояния, сбросе холодного состояния и сбросе теплого состояния этот POU будет выполнен повторно.

4.14.3. Программа UserPrg

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

Пользователь также может создавать дополнительные POU (программы, функции или функциональные блоки) и вызывать их или инстанцировать внутри POU UserPrg, в конце его программной инструкции. Также можно вызывать функции и инстанцировать функциональные блоки, определенные в библиотеках.

4.14.4. GVL System_Diagnostics

GVL System_Diagnostics содержит диагностические переменные процессора, коммуникационного интерфейса (Ethernet и PROFIBUS) и всех коммуникационных драйверов. Этот GVL не редактируется, и при добавлении в проект переменные объявляются автоматически с типом, определяемым типом устройства, к которому оно относится.

ВНИМАНИЕ

В GVL System_Diagnostics GVL также объявлены диагностические переменные прямого представления MODBUS запросов клиента/мастера.

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, как Server IEC 60870-5-104.

На следующем рисунке показан пример представления этого GVL в режиме Online.

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 38: System_Diagnostics GVL в режиме Online

4.14.5. GVL отклонения

В разделе отклонения GVL содержатся переменные отключения MODBUS Master/Client по символическому отображению реквизита. Не обязательно, но рекомендуется использовать автоматическую генерацию этих переменных, которая осуществляется нажатием кнопки Generate Disabling Variables на вкладке device requisition. Эти переменные объявляются как тип BOOL и имеют следующую структуру:

Запрос на отключение объявления переменных:

```
[Название устройства]_СНЯТЬ_[Номер заявки]: BOOL;
```

Где:

Имя устройства: имя, которое отображается в режиме Tree View для устройства MODBUS.

Requisition Number (Номер заявки): Номер заявки, который был объявлен в таблице заявок устройства MODBUS в последовательности сверху вниз, начиная с 0001.

Пример:

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Имя устройства: имя, отображаемое в режиме Tree View для устройства MODBUS.

Requisition Number (Номер заявки): Номер заявки, который был объявлен в таблице заявок устройства MODBUS в последовательности сверху вниз, начиная с 0001.

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 Generate Disabling Variables только создает переменные, но не удаляет их автоматически. Таким образом, в случае удаления какого-либо отношения соответствующая ему отключающая переменная должна быть удалена вручную.

Disables GVL является редактируемой, поэтому переменные отключения реквизитов могут быть созданы вручную без необходимости следовать модели, созданной автоматическим объявлением, и могут использоваться одновременно в двух вариантах, но всегда должны быть типа BOOL. При этом необходимо следить за тем, чтобы не удалять и не изменять автоматически объявленные переменные, так как они могут быть использованы для какого-либо устройства MODBUS. Если переменная будет удалена или изменена, то при компиляции проекта будет выдана ошибка. Для корректировки имени автоматически объявленной переменной необходимо следовать приведенному выше образцу в соответствии с устройством и реквизитом, к которому они относятся.

На следующем рисунке показан пример представления данной GVL в режиме Online. Если значения переменных TRUE, то это означает, что реквизит, к которому принадлежат переменные, отключен, и наоборот, если значение переменной FALSE.









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

Рисунок 39: Отключение GVL в режиме Online

4.14.6. GVL IOQualities

GVL IOQualities содержит переменные качества модулей ввода/вывода, объявленных на шине процессора. Этот 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: ARRAY[0..15] LibDataTypes.QUALITY;  
QUALITY_NX2020: ARRAY[0..15] LibDataTypes.QUALITY;  
QUALITY_NX6000: ARRAY[0..7] LibDataTypes.QUALITY;
```


4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

QUALITY_NX6100: ARRAY[0..3]
END_VAR

LibDataTypes.QUALITY;

После запуска приложения можно посмотреть значения переменных качества модулей ввода/вывода, которые были добавлены в проект через IOQualities GVL.

4.14.7. GVL Module_Diagnostics

GVL Module_Diagnostics содержит диагностические переменные модулей ввода/вывода, используемых в проекте, за исключением драйверов процессора и коммуникационных драйверов. Этот 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 этой GVL в режиме Online.

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 40: Модуль_диагностики GVL в режиме Online

4.14.8. GVL Qualities

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Переменные качества GVL содержат переменные качества внутренних переменных MODBUS Master/Client символического отображения. Не обязательно, но рекомендуется использовать автоматическую генерацию этих переменных, что делается нажатием кнопки Generate Quality Variables на вкладке device mapping. Эти переменные объявляются как тип LibDataTypes.QUALITY и имеют следующую структуру:

Объявление переменной отображения качества:

```
[Название устройства]_Качество_[Номер отображения]:LibDataTypes.QUALITY;
```

Где:

Название устройства: Имя, которое отображается в древовидном представлении для устройства.

Номер отображения: Номер отображения, объявленного в таблице отображения устройств, в последовательности "вверх-вниз", начиная с 0001.

ВНИМАНИЕ

Невозможно связать переменные качества с отображениями драйверов MODBUS Master/Client прямого представления. Поэтому рекомендуется использовать символическое отображение MODBUS-драйверов.

Пример: Device.Application.Qualities

VAR_GLOBAL

```
MODBUS_Device_QUALITY_0001: LibDataTypes.QUALITY;
```

```
MODBUS_Device_QUALITY_0002: LibDataTypes.QUALITY;
```

```
MODBUS_Device_QUALITY_0003: LibDataTypes.QUALITY;
```

END_VAR

Переменная качества GVL является редактируемой, поэтому переменные качества отображения могут быть созданы вручную, без необходимости следовать модели автоматического объявления, и могут быть использованы одновременно в двух вариантах. Однако они всегда должны иметь тип LibDataTypes.QUALITY, и необходимо следить за тем, чтобы не удалять и не изменять автоматически объявленные переменные, поскольку они могут быть использованы каким-либо устройством. 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 реквизитом, к которому оно относится.

Для устройств связи MODBUS переменные качества ведут себя так, как показано в табл. 54.

На следующем рисунке показан пример представления данного GVL в режиме Online.

ВНИМАНИЕ

Если переменная драйвера MODBUS Client/Master с символическим отображением будет отображена в драйвер IEC 60870-5-104 Server, то необходимо, чтобы переменные качества отображения MODBUS были созданы для генерации корректных событий качества для таких точек IEC 60870-5-104 Server. В обратном случае не будут генерироваться "плохие" события качества для клиентов сервера МЭК 60870-5-104 в ситуациях, когда ведущий/клиент MODBUS не может связаться со своими ведомыми/серверами, например.

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

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

Рисунок 41: Качества GVL в режиме онлайн

4.14.9. GVL ReqDiagnostics

В GVL ReqDiagnostics содержатся диагностические переменные запроса, имеющие символьную связку MODBUS Master/Client. Не обязательно, но рекомендуется использовать автоматическую генерацию этих переменных, которая осуществляется нажатием кнопки Generate Diagnostics Variables на вкладке Device Request. Объявление этих переменных имеет следующую структуру:

Объявление диагностической переменной запроса:

[Название устройства]_REQDG_[Номер заявки]: [Тип переменной];

Где:

Название устройства: Имя, которое отображается в древовидном представлении для устройства.

Номер отображения: Номер отображения, объявленного в таблице отображения устройств, в порядке возрастания и убывания, начиная с 0001.

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Тип переменной: NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS_RTU_MAPPING_1 для MODBUS Master и NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS_ETH_MAPPING_1 для MODBUS Client..

ВНИМАНИЕ

Диагностические переменные запроса при прямом отображении MODBUS Master/Client объявляются в System_Diagnostics GVL.

Пример:

Device.Application.ReqDiagnostics

VAR_GLOBAL

```

MODBUS_Device_REQDG_0001 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_RTU_MAPPING_1; MODBUS_Device_REQDG_0002 :
NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_RTU_MAPPING_1; MODBUS_Device_REQDG_0003 :
NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_RTU_MAPPING_1; MODBUS_Device_1_REQDG_0001
: NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_ETH_MAPPING_1; MODBUS_Device_1_REQDG_0002
: NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_ETH_MAPPING_1;

```

END_VAR

ReqDiagnostics GVL является редактируемым, поэтому диагностические переменные заявки 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 реквизитом, к которому оно относится.

На следующем рисунке показан пример представления данного GVL в режиме Online.

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 42: ReqDiagnostics GVL в режиме Online

4.14.10. Prepare_Start Function

4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

В данном POU определена функция системного события PrepareStart. Она относится к коммуникационной задаче и вызывается перед запуском приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке System Events объекта Task Configuration. Каждый раз, когда пользователь запускает приложение, счетчик увеличивается.

4.14.11. Prepare_Stop Function

В данном POU определена функция системного события PrepareStop. Она принадлежит задаче связи и вызывается перед остановкой приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке System Events объекта Task Configuration. Каждый раз, когда пользователь останавливает приложение, счетчик увеличивается.

4.14.12. Start_Done Function

В данном POU определена функция системного события StartDone. Она принадлежит коммуникационной задаче и вызывается при успешном запуске приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке System Events объекта Task Configuration. Каждый раз, когда пользователь успешно запускает приложение, счетчик увеличивается.

4.14.13. Stop_Done Function

В данном POU определена функция системного события StopDone. Она принадлежит задаче связи и вызывается при успешном завершении работы приложения. При активной связи с ПЛК можно наблюдать состояние события и счетчик вызовов на вкладке System Events объекта Task Configuration. Каждый раз, когда пользователь успешно останавливает приложение, счетчик увеличивается.

5. Конфигурация

Конфигурирование и программирование процессоров серии Nexto осуществляется с помощью программы MasterTool IEC XE. Конфигурация определяет поведение и режимы использования периферийных устройств, а также специальные возможности процессора. Программирование представляет собой приложение, разработанное пользователем.

5.1. Конфигурация процессора

5.1.1. Общие параметры

Приведенные ниже параметры являются частью конфигурации процессора, входящей в состав приложения. Для корректного выполнения проекта каждый из них должен быть проверен.

Кроме этих параметров, можно изменить имя каждого модуля, вставленного в приложение, нажав правую кнопку на модуле. В пункте Properties (Свойства) на общем листе изменить имя, которое ограничено 24 символами.

Настройки	Описание	Стандарт	Опции
%Q Начальный адрес	Область диагностики (%Q)		
	Начальный адрес диагностики UCP (%Q)	Автоматическое выделение проекта при создании.	0 до 97611
Размер	Размер области диагностики в байтах	693	Невозможно изменить размер области диагностики процессора
%Q Стартовый адрес	Подпорная площадь (%Q)		
	Начальный адрес памяти удерживаемых данных (%Q)	4096	0 до 98303
Размер	Размер памяти ретенционных данных в байтах	98304	0 до 98304
%Q Стартовый адрес	Постоянная область (%Q)		
	Начальный адрес памяти постоянных данных (%Q)	20480	0 до 98302
Размер	Размер памяти постоянных данных в байтах	98304	0 до 98304
Запуск пользовательского приложения после сброса по сторожевому таймеру	Параметры процессора		
	Если эта функция включена, то после сброса аппаратного сторожевого таймера или перезапуска режима Runtime запускается пользовательское приложение, но при этом сохраняется диагностическая	Отключено	Включено Отключено

5. КОНФИГУРАЦИЯ

	индикация через светодиод WD и через переменные.		
Настройки	Описание	Стандарт	Опции
Режим горячей замены	Режим горячей замены модулей	Включено, согласованность совпадений отсутствует. (может меняться в зависимости от модели процессора)	<ul style="list-style-type: none"> - Отключено, только для объявленных модулей - Отключено (с согласованностью) - Отключено, без согласования - Включено, с согласованием только для объявленных модулей - Включено, с согласованием - Включено, без согласования
Параметры TCP/IP			
Начальный тайм-аут (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 тайм-аут удваивается.	2	1 до 75
Задержка ACK (x10 мс)	Тайм-аут для отправки команды подтверждения.	10	0 до 100
Параметры проекта			
Последовательное сохранение и постоянная область в %Q	Конфигурация, состоящая из адресуемых постоянных и ретенционных запоминающих устройств.	без маркировки	<ul style="list-style-type: none"> - Проверено: Состоит из адресуемой постоянной и ретенционной памяти - Не отмечено: Не состоит из адресуемых постоянных и ретенционных запоминающих устройств.
Включить обновление ввода/вывода для каждой задачи	Настройка на обновление входов и выходов в задачах, в которых они используются.	без маркировки	<ul style="list-style-type: none"> - Проверено: Входы и выходы обновляются задачами, в которых они используются.

5. КОНФИГУРАЦИЯ

			- Не отмечено: Входы и выходы обновляются только в MainTask
Включить сохранение и постоянство переменных в функциональных блоках	Настройка, позволяющая использовать в функциональных блоках ретенционные и персистентные переменные	без маркировки	- Отмечено: разрешает использование ретенционных и персистентных переменных в функциональных блоках. - Не отмечено: При запуске может возникнуть ошибка исключения.

Таблица 42: Настройки процессора

Примечания:

Генерировать ошибку при согласовании задач сторожевого таймера: Этот параметр был упразднен начиная с версии MasterTool IEC XE 1.32. Включить обновление ввода/вывода для каждой задачи: Этот параметр был добавлен в MasterTool IEC XE версии 2.01.

Consist retain and persistent area in %Q: Если пользовательское приложение объявляет retain или persistent переменные, отображенные по адресам %Q, не входящим в конфигурируемые области, то эти переменные не будут сохраняться после выключения питания, сброса в теплом режиме или загрузки. При отмеченном параметре MasterTool будет проверять код и генерировать ошибку во время компиляции для данной ситуации. Настоятельно рекомендуется держать этот параметр отмеченным.

ВНИМАНИЕ

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

ВНИМАНИЕ

В ситуациях, когда модифицируется символьная постоянная область памяти, программист MasterTool IEC XE выводит сообщение о выборе поведения для этой области после зарядки модифицированной программы. Выбор этого поведения не влияет на постоянную область прямого представления, которая всегда чиста.

ВНИМАНИЕ

Опция Включить обновление входов/выходов по заданию не поддерживается для мастеров полевой шины, таких как модуль NX5001. Эта 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 в каждой задаче, она будет продолжать обновляться и в MainTask; за исключением случаев, 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 процессорах серии Nexto реализована возможность замены модулей ввода/вывода в шине без необходимости выключения системы и без потери информации. Эта возможность известна как горячая замена.

ОСТОРОЖНО

Процессоры серии Nexto не гарантируют сохранение постоянных и ретенционных переменных в случае, если блок питания или даже сам процессор будут извлечены из стойки объединительной платы, находящейся под напряжением.

При горячей замене соответствующее поведение системы изменяется в соответствии с заданной пользователем конфигурационной таблицей, в которой представлены следующие варианты:

- Отключено, только для объявленных модулей
- Отключено (с согласованностью при запуске)
- Отключено, без согласованности при вводе в эксплуатацию
- Включено, с согласованностью запуска только для объявленных модулей
- Включено, с согласованностью при вводе в эксплуатацию
- Включено, без согласованности при вводе в эксплуатацию

Таким образом, пользователь может выбрать поведение системы в нештатных ситуациях на шине и когда процессор находится в режиме работы. В таблице ниже представлены возможные нештатные ситуации на шине.

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

Таблица 43: Нештатные ситуации с автобусами

Более подробная информация о диагностике, соответствующей описанным выше ситуациям, приведена в разделе Диагностика по переменным.

Если модуль присутствует в определенной позиции, в которой он не должен находиться в соответствии с конфигурационными модулями, то этот модуль считается недеklarированным. Опции горячей замены Отключена, только для объявленных модулей и Включена, с согласованностью запуска только для объявленных модулей не учитывают модули, находящиеся в таком состоянии.

5.1.1.1.1. Горячая замена отключена, только для заявленных модулей

В этой конфигурации при возникновении нештатной ситуации на шине (как описано в табл. 43) процессор немедленно переходит в режим останова. Светодиод DG начинает мигать 4 раза (в соответствии с табл. 44). В этом случае для возвращения процессора в нормальное состояние Run, помимо отмены того, что вызвало нештатную ситуацию, необходимо выполнить сброс тепла или сброс холода.

5. КОНФИГУРАЦИЯ

Если выполняется команда *Reset Origin*, то необходимо выполнить загрузку, чтобы процессор мог вернуться в нормальное состояние (*Run*). Команды *Сброс Тепла*, *Сброс Холода* и *Сброс Исходного состояния* могут быть выполнены с помощью *MasterTool IEC XE* в меню *Online*.

CPU останется в нормальном состоянии *Работы*, даже если найдет модуль, не заявленный в шине.

5.1.1.1.2. Горячая замена отключена

При такой настройке запрещается любая нештатная ситуация в шине (как показано в табл. 43), включая недеklarированные и присутствующие на шине модули. При этом процессор переходит в режим *Stop*, а светодиод *DG* начинает мигать 4 раза (как в табл. 44). В этих случаях для возврата процессора в нормальный режим работы, помимо устранения причины нештатной ситуации, необходимо выполнить сброс тепла или холода (*Reset Warm* или *Reset Cold*). Если выполняется *Сброс исходного состояния* (*Reset Origin*), то необходимо загрузить проект, чтобы процессор мог вернуться в нормальный режим *Работы*. Команды *Reset Warm*, *Reset Cold* и *Reset Origin* могут быть выполнены в *MasterTool IEC XE* в меню *Online*.

5.1.1.1.3. Отключение горячей замены, отсутствие согласованности при запуске

Позволяет запустить систему даже в том случае, если какой-либо модуль находится в нештатной ситуации на шине (как показано в табл. 43). О нештатных ситуациях сообщается через диагностику.

Любая модификация шины приведет к тому, что процессор перейдет в режим *останова*, а светодиод *DG* начнет мигать 4 раза (как показано в табл. 44). Для возврата процессора в нормальное состояние *Run* в этих случаях необходимо выполнить *Reset Warm* или *Reset Cold*. Если выполняется *Reset Origin*, то необходимо загрузить процессор, чтобы он мог вернуться в нормальное состояние *Run*. Команды *Reset Warm*, *Reset Cold* и *Reset Origin* могут быть выполнены с помощью *MasterTool IEC XE* в меню *Online*.

5.1.1.1.4. Горячая замена включена, согласованность запуска только для объявленных модулей

"*Пуск*" - это интервал времени между подачей питания на процессор (или команды сброса, или загрузки приложения) и первым переходом процессора в режим *работы* после включения. В данной конфигурации проверяется, не возникла ли во время запуска какая-либо нештатная ситуация на шине (как описано в табл. 43). В утвердительном случае процессор переходит в режим *останова*, а светодиод *DG* начинает мигать 4 раза (в соответствии с табл. 44). После этого, чтобы перевести процессор в режим *Run*, а затем устранить причину нештатной ситуации, необходимо выполнить команду *Reset Warm* или *Reset Cold*, которая может быть выполнена с помощью *MasterTool IEC XE* (меню *Online*). Если выполняется команда *Reset Origin*, то необходимо выполнить загрузку, чтобы процессор мог вернуться в нормальное состояние (*Run*).

После запуска, если в каком-либо модуле возникнет ситуация, описанная в предыдущей таблице, система продолжит работать в нормальном режиме и просигнализирует о проблеме посредством диагностики.

Если никаких других отклонений в работе заявленных модулей нет, процессор перейдет в нормальное состояние (*Run*), даже если на шине присутствует не заявленный модуль.

ВНИМАНИЕ

В этой конфигурации при возникновении (даже временного) сбоя питания, выполнении команд *Reset Warm Command*, *Reset Cold Command* или новой загрузки приложения, а также при возникновении нештатной ситуации на шине любого модуля, процессор переходит в режим *останова*, а светодиод *DG* начинает мигать 4x (в соответствии с табл. 44). Это считается ситуацией запуска. Этот 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 с поддержкой последовательности запуска

Эта настройка проверяет, не возникла ли во время запуска какая-либо нештатная ситуация в шине (см. табл. 43), даже если на шине нет заявленных и присутствующих модулей; если да, то процессор переходит в режим *Stop*, а светодиод *DG* начинает мигать 4x (см. табл. 44). В этих случаях для возврата процессора в нормальный режим работы,

5. КОНФИГУРАЦИЯ

помимо устранения причины нештатной ситуации, необходимо выполнить Reset Warm или Reset Cold. Если выполняется Reset Origin, то необходимо загрузить проект, чтобы процессор мог вернуться в нормальный режим Run. Команды Reset Warm, Reset Cold и Reset Origin могут быть выполнены в MasterTool IEC XE в меню Online.

5.1.1.1.6. Горячая замена включена без согласованности запуска

Позволяет начать работу системы, даже если модуль находится в нештатной ситуации на шине (как описано в табл. 43). О нештатных ситуациях сообщается через диагностику во время и после ввода в эксплуатацию.

ВНИМАНИЕ

Этот вариант рекомендуется использовать на этапе внедрения системы, так как 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 возможную статическую энергию, накопившуюся в теле. Для этого перед обращением с модулями 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 использованием, рекомендуется осуществлять контроль диагностики "горячей замены" в прикладном контроле, разработанном пользователем.

Процедура "горячей замены" описана ниже:

- Отсоедините модуль от стойки объединительной платы с помощью предохранителя.
- Снимите модуль, сильно потянув за него.
- Вставьте новый модуль в стойку объединительной платы.
- Убедитесь, что предохранительный замок 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 процессор переходит в режим останова и светодиод DG начинает мигать 4 раза (в соответствии с табл. 44, из-за любой нештатной ситуации на шине (как описано в табл. 43), при переходе процессора из режима работы в режим останова выходные модули имеют работу своих точек в соответствии с конфигурацией модуля. В случае запуска приложения, когда процессор переходит в режим останова, не перейдя в режим работы, выходные модули переводят свои точки в режим защиты от сбоев, другими словами, выключают их (0 В пост. 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 демонтажа разъема состояние логической точки будет переведено в безопасное состояние, что означает нулевой или высокий уровень импеданса.

5. КОНФИГУРАЦИЯ

ВНИМАНИЕ

Для обновления состояния модулей процессор всегда заменяет по одному модулю.

Ниже, в табл. 44, представлены состояния шины и состояние светодиода Nexto CPU DG. Более подробная информация о состояниях светодиодов диагностики приведена в разделе Диагностика с помощью светодиодов.

Состояние	Включено, с согласованностью запуска	Включено, с согласованностью запуска только для объявленных модулей	Включено, без согласованности запуска	Отключено	Отключено, только для объявленных модулей	Отключено, без согласованности запуска
Недекларируемый модуль	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 4 раза Применение: Остановка
Недекларированный модуль (состояние ввода в эксплуатацию)	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить
Отсутствующий модуль	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка
Отсутствующий модуль (состояние ввода в эксплуатацию)	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 2 раза Применение: Запустить
Несовместимая конфигурация	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 2 раза Применение: Запустить	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка

5. КОНФИГУРАЦИЯ

Несовместимая конфигурация (состояние запуска)	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 2 раза Применение: Запустить или Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	LED DG: Blinks 2x Application: Run
Дублирование адреса слота	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка
Нерабочий модуль	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка	Светодиод DG: мигает 4 раза Применение: Остановка

Таблица 44: Горячая замена и соотношение 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 Stop в режим Run. Однако если этот модуль сконфигурирован как NX5000 или NX5001, а на его месте стоит другой модуль, то приложение останется в состоянии остановки..

5.1.1.2. Сохраняемые и постоянные области памяти

Процессор Nexto позволяет использовать символьные переменные и выходные переменные прямого представления в качестве ретенционных или персистентных переменных.

Выходные переменные прямого представления, которые будут ретенционными или персистентными, должны быть объявлены в Общих параметрах процессора, как описано в разделе Конфигурация процессора. Символические имена также могут быть присвоены этим выходным переменным прямого представления с помощью директивы AT, а также с использованием ключевого слова RETAIN или PERSISTENT при их объявлении. Например, %QB4096 и %QB20480 в ретенционной и персистентной памяти, соответственно:

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

Если символьные переменные, объявленные с помощью директивы AT, не находятся в соответствующей ретенционной и/или постоянной памяти, то при генерации кода в MasterTool могут возникнуть ошибки (как описано

5. КОНФИГУРАЦИЯ

в разделе Общие параметры, конфигурация Consist retain and persistent area in %Q), информирующие о том, что в ретенционной или постоянной области памяти определены неретенционные или непостоянные переменные.

Что касается символьных переменных, которые будут ретенционными или персистентными, то только ретенционные переменные могут быть локальными или глобальными, так как персистентные символьные переменные всегда должны быть глобальными. Для объявления ретенционных символьных переменных необходимо использовать ключевое слово RETAIN. Например, для локальных переменных:

```
PROGRAM 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 символьные переменные должны быть объявлены в объекте Persistent Variables, добавляемом в приложение. Эти переменные будут глобальными и будут объявлены в объекте следующим образом:

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

Начиная с версии 1.5.1.1 процессоры серии Nexto позволяют гибко подходить к использованию ретентивной и персистентной памяти. Это означает, что пользователь может самостоятельно выбирать объем памяти каждого типа, если сумма ретенционной и постоянной памяти не превышает общего лимита, доступного в каждой модели процессора. Сумма доступной ретентивной и постоянной памяти описана в таблице 6 в разделе "Память".

Если сумма ретентивной символьной, постоянной символьной, ретентивной %Q и постоянной %Q памяти превысит общее количество доступной, MasterTool выдаст ошибку при генерации кода.

```
VAR_GLOBAL PERSISTENT RETAIN wGlobalSymbolicPersistentVariable_01 : WORD;
END_VAR

VAR_GLOBAL RETAIN wGlobalSymbolicRetentiveVariable_01 : WORD;
END_VAR
```

ВНИМАНИЕ

Для гибкого использования ретенционной и персистентной памяти необходимо использовать MasterTool IEC XE 2.03 или выше.

5.1.1.3. Конфигурации TCP

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

- Начальный тайм-аут
- Задержка ACK

5. КОНФИГУРАЦИЯ

CPU Nexto, прежде чем ответить на любой запрос, как и любое другое оборудование 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 вызвать проблемы для установленных соединений в режиме сервера, так как закрытие соединений зависит от другого оборудования - клиента.

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

Initial Time-out: указывает, через какое время после первой передачи сообщения оно должно быть повторно передано, если предположить, что оно не было получено конечным устройством. При каждой повторной передаче тайм-аут удваивается. Количество попыток передачи связано с таймаутом связи, настроенным в протоколе. Это будет максимальное время до отказа от доставки сообщения, когда конкретизируется сбой передачи. Кроме того, важно подчеркнуть, что для процессоров 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 тайм-аута 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 нового тайм-аута используется статистика последних соединений.

Пример начального тайм-аута и параметров тайм-аута связи внутри MODBUS TCP сервера, учитывая неполученное подтверждение, для следующих значений: 300 для начального тайм-аута (=300 мс) и 3000 для тайм-аута связи (=3000 мс):

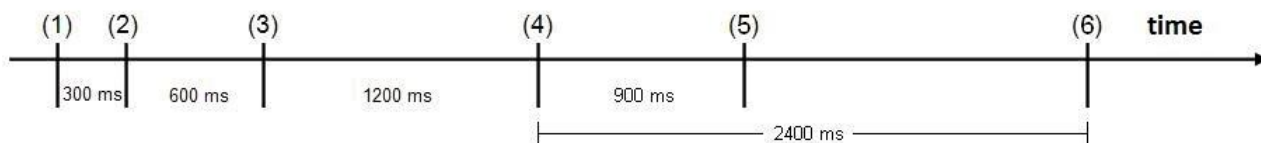


Рисунок 43: Начальный тайм-аут и тайм-аут связи

Комментарий к графику:

1. Мгновение передачи сообщения.
2. Первая попытка передачи сообщения после начального тайм-аута.
3. Вторая попытка передачи сообщения, после двукратного превышения начального тайм-аута.
4. Третья попытка передачи сообщения, после двукратного превышения последнего тайм-аута.
5. Прекращение передачи сообщения и индикация отказа по превышению тайм-аута связи (общее время до сдачи: $300 + 600 + 1200 + 900 = 3000$ мс).
6. Это была бы четвертая попытка передачи сообщения, после двукратного превышения последнего тайм-аута, но тайм-аут связи, настроенный в протоколе, был превышен, и произошла индикация отказа.

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

- Все сообщения запроса, отправляемые клиентом на сервер, должны быть подтверждены сервером посредством ACK-сообщения, передаваемого клиенту.
- Все ответные сообщения, посылаемые клиентом серверу, должны быть подтверждены сервером посредством ACK-сообщения, передаваемого клиенту.
- Если одна из частей не получит ACK-сообщение в течение времени, определяемого тайм-аутом TCP, то сообщение будет повторно передано по адресу отправителя (см. параметр TCP number of tries).
- Сообщение ACK не обязательно должно быть эксклюзивным. ACK, который сервер должен послать клиенту в момент получения запроса, может быть включен в то же сообщение ответа, а

5. КОНФИГУРАЦИЯ

ACK, который клиент должен послать серверу в момент запроса, может быть включен в то же сообщение, включающее следующий запрос.

На следующих рисунках показана разница между отправкой немедленного и калиброванного ACK:

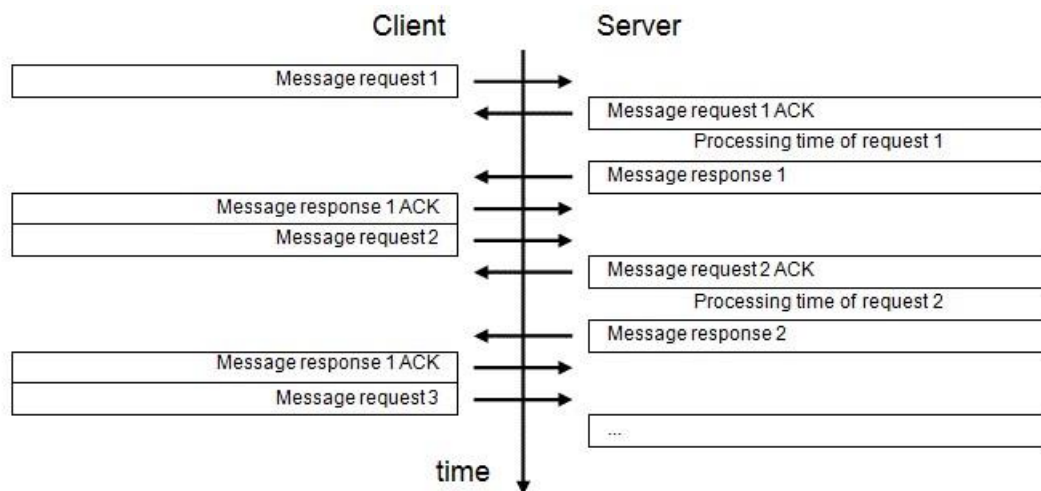


Рисунок 44: Пример немедленной отправки ACK (=0)

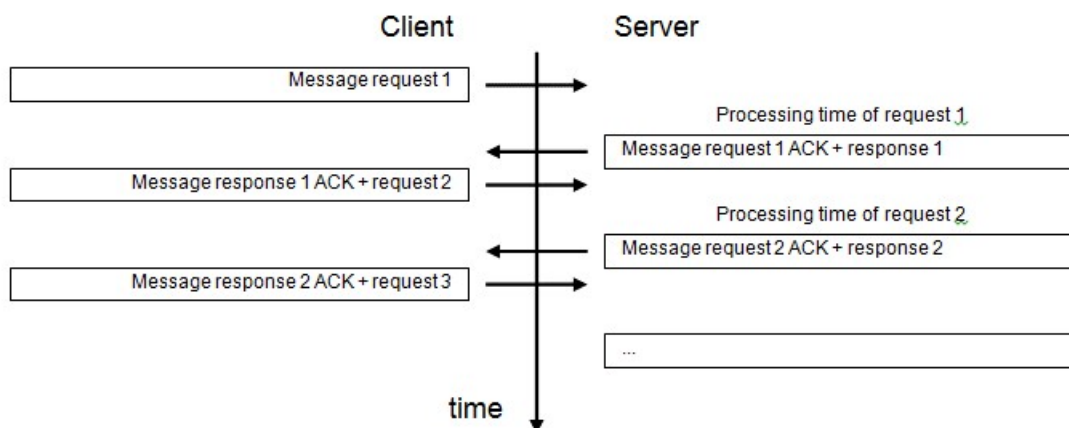


Рисунок 45: Пример калиброванной отправки ACK

5.1.1.4. Параметры проекта

Параметры проекта CPU связаны с конфигурацией для обновления ввода/вывода при использовании задач проекта и согласованности ретенционной и персистентной области в %Q, а также с опциями для чтения и записи на карту памяти.

Конфигурация	Описание	По умолчанию	Опции
Сохранение и постоянная область в %Q	Выполняет согласование ретенционных и персистентных областей в %Q	Без маркировки	- Маркированный - Без маркировки
Включить обновление ввода/вывода для каждой задачи	Обновляет входные и выходные данные в задачах, где они используются	Без маркировки	- Маркированный - Без маркировки

5. КОНФИГУРАЦИЯ

Включить сохранение и постоянство переменных в функциональных блоках	Настройка, разрешающая использование ретенционных и персистентных переменных в функциональных блоках	Без маркировки	- Маркированный - Без маркировки
Копирование проекта с процессора на карту памяти	Карта памяти		
	Копирование проекта из внутренней памяти процессора на карту памяти	Отключено	- Включено: Конфигурация включена - Отключено: Конфигурация отключена
Пароль для копирования проекта с процессора на карту памяти	Пароль для копирования проекта из внутренней памяти процессора на карту памяти	-	6-значный пароль (от 0 до 999999)
Копирование проекта с карты памяти на процессор	Копирование проекта с карты памяти во внутреннюю память процессора	Отключено	- Включено: Конфигурация включена - Отключено: Конфигурация отключена
Пароль для копирования проекта с карты памяти на процессор	Пароль для копирования проекта с карты памяти на внутреннюю память процессора	-	6-значный пароль (от 0 до 999999)

Таблица 45: Параметры проекта ЦПУ

ВНИМАНИЕ

После установки возможностей копирования проекта и создания загрузочного приложения необходимо найти файл "Application.crc", чтобы конфигурации, касающиеся карты памяти, имели силу. Поиск может быть выполнен при выборе Application.crc через клавишу Find File..., как показано на рис. 136.

5.1.2. Конфигурация внешних событий

Внешнее событие - это функция, имеющаяся в процессоре, которая позволяет сконфигурированному пользователем цифровому входу при активации запускать выполнение определенной задачи с заданным пользователем кодом. Таким образом, можно через этот вход при его срабатывании прервать выполнение основного приложения и запустить установленное приложение в задаче ExternalInterruptTask00, 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 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 заблокировало работу процессора, так как задача имеет более высокий приоритет по сравнению с другими.

5. КОНФИГУРАЦИЯ

Для конфигурирования внешнего события необходимо вставить модуль дискретного ввода и выполнить описанные ниже конфигурации в центральном процессоре с помощью программного обеспечения MT8500.

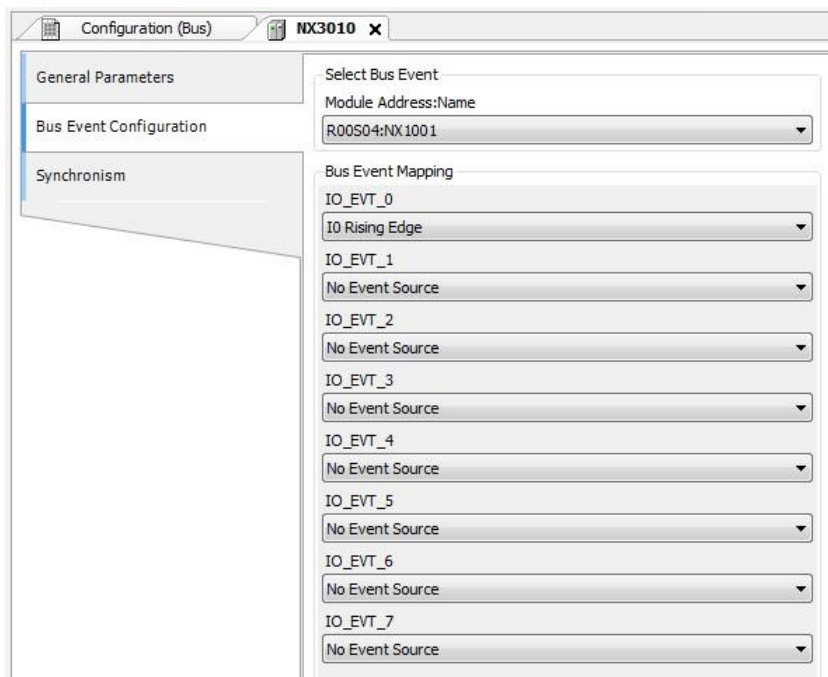


Рисунок 46: Экран конфигурации для внешнего события в процессоре

На вкладке конфигурации Внешнее событие в настройках процессора необходимо выбрать, какой модуль будет являться источником прерывания, в поле Адрес модуля: Название. Затем необходимо выбрать, какой вход этого модуля будет отвечать за генерацию события (IO_EVT_0). В этом выборе могут быть выбраны варианты, описанные на рисунке ниже.

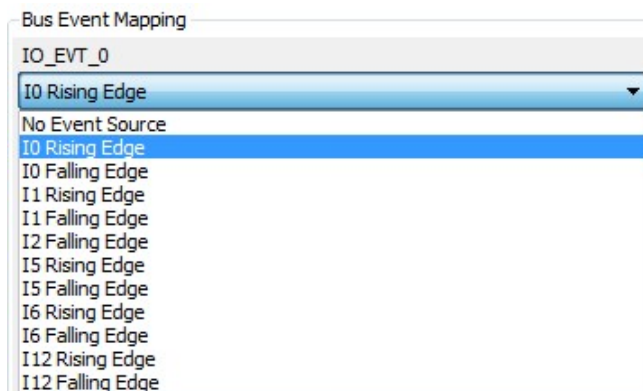


Рисунок 47: Параметры внешнего источника событий модуля NX1001

Помимо настройки процессора 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 (рис. ниже) необходимо выбрать источник события в соответствующем поле. В данном случае следует выбрать IO_EVT_0, поскольку другие источники событий (IO_EVT_1 - IO_EVT_7) недоступны. В последовательности действий в поле POU следует проверить, правильно ли выбрано POU, поскольку оно будет использоваться пользователем для определения действий, которые необходимо выполнить при наступлении внешнего события.

5. КОНФИГУРАЦИЯ

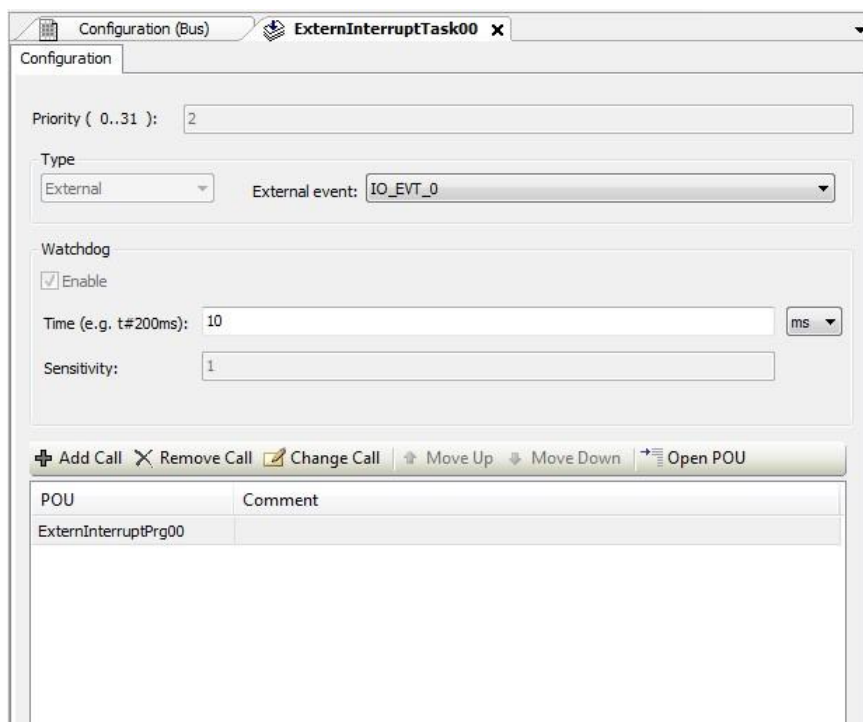


Рисунок 48: Экран конфигурации ExternInterruptTask00

5.1.3. Конфигурация SOE

SOE ("Последовательность событий") отвечает за генерацию последовательности цифровых событий. С помощью SOE можно анализировать историческое поведение системных переменных, отображаемых в зоне мониторинга. SOE является эксклюзивной услугой, доступной для моделей NX3020 и NX3030.

После включения сервиса SOE процессор начинает работать как DNP3-сервер, поэтому для использования данного ресурса необходима поддержка протокола DNP3 клиентом. Поддерживаемые типы объектов, а также коды функций и квалификаторы приведены в Приложении. [Взаимодействие с DNP3](#).

Сервис SOE использует адреса %Q для формирования своей базы статических данных. Для нее должна быть выделена непрерывная область памяти %Q, в которой пользователь будет сообщать ее начало и размер. Для резервных проектов область %Q также должна быть резервной, чтобы в момент переключения сохранилась база данных сервера DNP3.

Адрес первого объекта DNP3 всегда будет равен 0, что соответствует биту 0 в %QBxxx, где xxxx - начальный адрес %Q.

Таким образом, определив статическую базу данных, пользователь должен скопировать каждую цифровую точку, которая должна генерировать события в непрерывной области %Q. Максимальное количество точек, которые могут быть скопированы, составляет 8000.

Для настройки событий необходимо сообщить только размер очереди событий. В SOE используется специальная очередь (не та, что описана в разделе "Конфигурация протоколов"), которая является постоянной и резервной, поэтому события не будут потеряны ни в момент переключения, ни в случае сбоя питания. В случае переполнения очереди событий самые старые события будут перезаписаны. Если за один цикл генерируется больше событий, чем поддерживает очередь, то ее генерация прерывается и включается диагностика переполнения (SOE[x].bOverflowStatus). Например, если в конфигурации из 100 событий меняется 100+n битов, что приводит к утилизации n событий.

SOE будет выполняться в контексте MainTask, начиная уже с первого цикла. В конце каждого цикла MainTask SOE будет сравнивать отображенные биты, чтобы обнаружить переходы, произошедшие в этом цикле. Таким образом, в каждом цикле, в котором генерируются события, будет происходить увеличение времени в этом цикле MainTask. В наихудшем случае (1000 событий, при этом генерируется только 1000, а остальные отбрасываются) это влияние составит примерно 5 мс. Поэтому для приложения с включенным ГП пользователь должен будет учитывать это время при установке параметров времени сторожевого таймера и интервала основной задачи.

5. КОНФИГУРАЦИЯ

Для его использования пользователю необходимо установить следующие параметры на вкладке SOE Конфигурация:

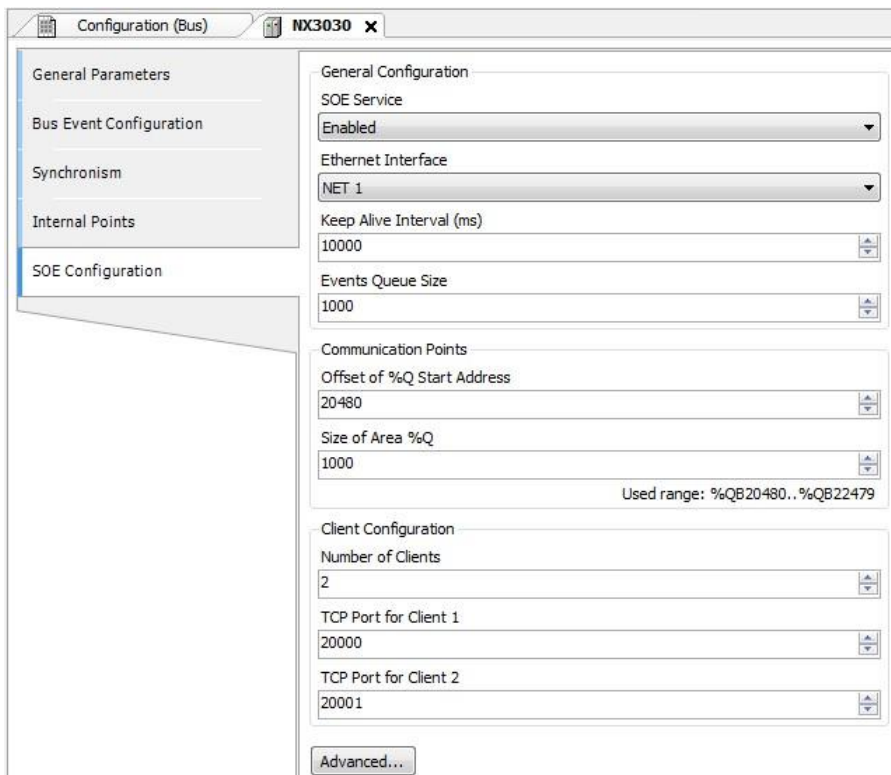


Рисунок 49: Конфигурация последовательности событий

Конфигурация	Описание	Значение по умолчанию	Опции
Сервис SOE	Общая конфигурация		
	Включение ГП	Отключено	Включено Отключено
Интерфейс Ethernet	Выбор используемого интерфейса	NET 1	NET 1 NET 2
Интервал поддержания в рабочем состоянии (мс)	Сообщения с интервалом поддержания жизни (мс)	10000	0 до 4294967295
Размер очереди событий	Размер очереди событий	1000	100 до 1000
Смещение начального адреса %Q	Точки коммуникации		
	Начальный адрес для статических данных	20480	Может быть использован любой адрес области %Q
Размер области %Q	Объем памяти, используемый статическими данными (%Q)	1000	1 до 1000
Конфигурация клиента			

5. КОНФИГУРАЦИЯ

Количество клиентов	Определяет количество клиентов	2	1, 2
TCP-порт для клиента 1	Выбор коммуникационного порта для первого клиента	20000	1 до 65535
TCP-порт для клиента 2	Выбор коммуникационного порта для второго клиента	20001	1 до 65535

Таблица 46: Конфигурация SOE

Примечания:

Размер памяти данных: Размер памяти данных, зарезервированный для использования статических данных, всегда будет в два раза больше установленного значения, так как вторая половина области памяти используется для хранения предыдущих значений переменных первой половины.

Keep Alive: Во время соединения с клиентом через заданные интервалы времени будут отправляться сообщения keep alive. Если клиент не отвечает на эти сообщения, то соединение закрывается. То есть соединение между клиентом и сервером может длиться время, равное установленному интервалу, чтобы в случае ошибки быть закрытым.

В дополнительных опциях (клавиша Advanced...) можно задать адреса связи по протоколу DNP3.

Конфигурация	Описание	Значение по умолчанию	Опции
Адрес источника DNP3	Адрес первого клиента (PLC)	4	0 до 65519
DNP3 Адрес назначения клиента 1	Адрес первого клиента	3	0 до 65519
DNP3 Адрес назначения клиента 2	Адрес второго клиента	3	0 до 65519

Таблица 47: Расширенные конфигурации SOE

Примечание:

Адрес DNP3: Адреса DNP3 из диапазона 65520 - 65535 не могут быть заданы ни в начале, ни в конце пути, поскольку они используются для широковещательных сообщений.

ВНИМАНИЕ

Сообщения DNP3 Data Link не используются процессорами серии Nexto, так как стандарт не рекомендует использовать их в TCP/IP-коммуникациях.

5.1.4. Синхронизация времени

Для синхронизации времени процессоры серии Nexto используют протокол SNTP (Simple Network Time Protocol) или синхронизацию по стандарту IEC 60870-5-104.

Для использования протоколов синхронизации времени пользователю необходимо установить следующие параметры на вкладке Synchronism, доступ к которой осуществляется через CPU в дереве устройств:

5. КОНФИГУРАЦИЯ

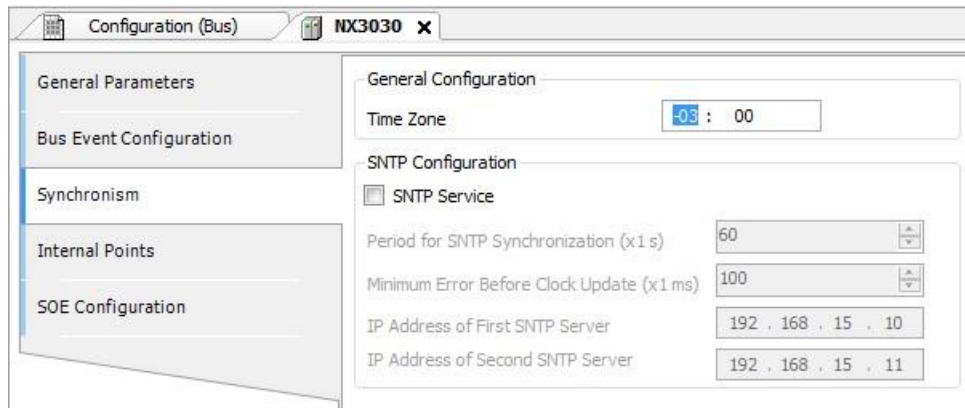


Рисунок 50: Конфигурация SNTP

Конфигурация	Описание	По умолчанию	Опции
Часовой пояс (чч:мм)	Часовой пояс местонахождения пользователя. Можно вставить часы и минуты.	-3:00	12:59 до +13:59
Служба SNTP	Включает службу SNTP.	Отключено	Отключено Включено
Период синхронизации SNTP (x1 с)	Временной интервал запросов синхронизации (секунды).	60	1 до 255
Минимальная ошибка перед обновлением часов (x1 мс)	Допустимое значение смещения между сервером и клиентом (миллисекунды).	100	1 до 65519
IP-адрес первого SNTP-сервера	IP-адрес первичного SNTP-сервера.	192.168.15.10	1.0.0.1 до 223.255.255.254
IP-адрес второго SNTP-сервера	IP-адрес вторичного SNTP-сервера.	192.168.15.11	1.0.0.1 до 223.255.255.254

Таблица 48: Конфигурации SNTP

Примечания:

Сервер SNTP: Можно задать преимущественный и вторичный адрес для доступа к SNTP-серверу и, соответственно, для получения синхронизации времени. Если оба поля пусты, то служба SNTP будет отключена.

Часовой пояс: Настройка часового пояса используется для преобразования местного времени в UTC и наоборот. Если одни источники синхронизации используют местное время (протокол IEC 60870-5-104, функция SetDateAndTime), то другие - время UTC (SNTP). Время UTC обычно используется для штамповки событий (DNP3, IEC 60870-5-104 и MasterTool Device Log), а местное время - для функций других процессоров (функция GetDateAndTime, OTD date and time info).

В проекте допускается включение более одного источника синхронизации, однако в процессе работы устройство не поддерживает синхронизацию от нескольких источников синхронизации. Поэтому неявно определен механизм приоритета. Синхронизация по протоколу SNTP является более приоритетной, чем по протоколу IEC 60870-5-104. Так,

5. КОНФИГУРАЦИЯ

если оба источника включены и присутствует SNTP-сервер, то он будет отвечать за синхронизацию часов процессора, а любая команда синхронизации от IEC 60870-5-104 будет отклонена.

5.1.4.1. IEC 60870-5-104

В случае если синхронизация осуществляется по протоколу IEC 60870-5-104, то для получения синхронизации часов пользователю необходимо включить синхронизацию времени на экране конфигурации протокола. Для установки этой опции на устройстве необходимо отметить параметр Enable Time Synchronization, доступный в разделе Application Layer.

ВНИМАНИЕ

Если ПЛК получает команду синхронизации времени от центра управления, а эта опция отключена, то на эту команду будет получен ответ об ошибке. Если же эта опция включена, то в центр управления будет возвращено сообщение об успехе, даже о том, что команда синхронизации отклонена, поскольку активен другой метод синхронизации с более высоким приоритетом.

Данный метод синхронизации следует использовать только как вспомогательный, поскольку точность процесса синхронизации часов сильно зависит от задержек и трафика в сети, а также от загрузки процессора CPU, так как данный механизм относится к низкоприоритетным задачам.

ВНИМАНИЕ

В архитектурах с резервированными ПЛК драйвер сервера IEC 60870-5-104 отключается на неактивном ПЛК. Таким образом, не рекомендуется использовать данный метод синхронизации в резервированных системах. Так как после переключения неактивному ПЛК может потребоваться несколько секунд для синхронизации часов. Для резервированных систем рекомендуется использовать SNTP.

5.1.4.2. SNTP

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

Процессор посылает циклические запросы на синхронизацию в соответствии со временем, установленным в поле Period for SNTP Synchronization. При первой попытке синхронизации, сразу после запуска сервиса, запрос идет на первый сервер, заданный в IP-адресе первого сервера. Если он не отвечает, то запросы направляются на второй сервер, заданный в IP-адресе второго сервера, что обеспечивает резервирование SNTP-серверов. 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 процессор пытается подключиться к каждому серверу по одному разу, при этом он пытается подключиться ко второму серверу в случае, если первый не отвечает. Время ожидания ответа от SNTP-сервера по умолчанию определено в 5 с и не может быть изменено.

Если после синхронизации разница между текущим временем процессора и временем, полученным сервером, превышает значение, заданное в параметре Minimum Error Before Clock Update, то происходит обновление времени процессора. SNTP использует время в формате UTC (Universal Time Coordinated), поэтому необходимо правильно установить параметр Time Zone, чтобы время, считанное SNTP, было правильно переведено в местное время.

Процесс выполнения SNTP-клиента можно проиллюстрировать следующими шагами:

1. Попытка синхронизации через первый сервер. В случае успешной синхронизации процессор выжидает время для новой синхронизации (Period for SNTP Synchronization) и снова синхронизируется с этим сервером, используя его в качестве основного. В случае неудачи (сервер не отвечает менее чем за 5 с) выполняется шаг 2.

5. КОНФИГУРАЦИЯ

2. Попытка синхронизации через второй сервер. В случае успешной синхронизации процессор выжидает время для новой синхронизации (Period for SNTP Synchronization) и пытается синхронизироваться с этим 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 ответа SNTP-сервера составляет 5 с, пользователь должен обращать внимание на значения Synchronization Period меньше 10 с. В случае если первичный сервер не отвечает, время синхронизации составит минимум 5 с (ожидание ответа первичного сервера и попытка синхронизации со вторичным сервером). В случае если ни первичный, ни вторичный сервер не отвечают, время синхронизации составит минимум 10 с (ожидание ответа обоих серверов и попытка нового соединения с первым сервером).

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

ВНИМАНИЕ

Служба SNTP зависит в своей конфигурации только от пользовательского приложения. Поэтому эта служба будет выполняться, даже когда процессор находится в режимах остановки или отбоя, если в процессоре есть приложение с включенным и правильно сконфигурированным SNTP-клиентом.

ОСТОРОЖНО

Очень важно настроить хотя бы один SNTP-сервер. Рекомендуется настраивать два SNTP-сервера (первичный и вторичный). Синхронизация по протоколу SNTP необходима для формирования событий с согласованной временной меткой между CPA и CPB и с мировым временем. Другой полезной функцией является предотвращение сбоев при переключении в приложениях, использующих дату и время, учитывая отсутствие синхронизации даты и времени между ПЛК по каналам синхронизации NETA и NETB..

5.1.4.3. Переход на летнее время (DST)

Настройка DST должна осуществляться косвенно через функцию SetTimeZone, которая изменяет часовой пояс, применяемый к RTC. В начале DST необходимо использовать функцию для увеличения часового пояса на один час. В конце DST она используется для уменьшения часового пояса на один час.

Более подробная информация приведена в разделе [Часы RTC](#).

5.1.5. Внутренние пункты

Точка связи хранится в памяти процессора в виде двух различных переменных. Одна из них представляет собой значение точки (тип BOOL, BYTE, WORD и т.д.), а другая - ее качество (тип QUALITY). Внутренние точки - это точки, значение и качество которых вычисляются внутри приложения пользователя, т.е. не имеют внешнего происхождения, как это происходит с точками, связанными с IED (коммуникационными драйверами типа Master/Client) или с локальными модулями ввода/вывода.

ВНИМАНИЕ

Different В отличие от модулей ввода/вывода, объявленных на локальной шине, которые имеют свои собственные переменные качества, созданные MasterTool (IOQualities GVL) и автоматически обновляемые CPU, модули ввода/вывода, объявленные на удаленных точках PROFIBUS, таких переменных не имеют. Объявление переменных качества точки PROFIBUS, связь этих переменных качества с переменными значения на закладке Internal Points, а также генерация и обновление значений переменных качества из имеющейся диагностики PROFIBUS является обязанностью пользователя: модулей ввода/вывода PROFIBUS, головки PROFIBUS и PROFIBUS Master.

5. КОНФИГУРАЦИЯ

Эта вкладка конфигурации "Внутренние точки" предназначена для связи переменной, представляющей значение точки, с переменной, представляющей ее качество. Она должна использоваться для связи переменных значения и качества, созданных внутри программы ПЛК (как в GVL), которые, как правило, впоследствии будут отображены на коммуникационный драйвер типа Сервер для связи с центром управления.

ВНИМАНИЕ

Если переменная значения не имеет связанной с ней переменной качества, то при сообщении переменной значения клиенту или центру управления она будет представлена как постоянное хорошее качество по умолчанию (без существенных признаков).

Таким образом, цель этой вкладки - не создание и не объявление внутренних точек. Для этого достаточно объявить переменные значения и/или качества в GVL и отобразить их на коммуникационный драйвер.

Конфигурация внутренних точек, представленная на рисунке ниже, соответствует параметрам, описанным в таблице. В таблице внутренних точек можно сконфигурировать до 5120 записей.

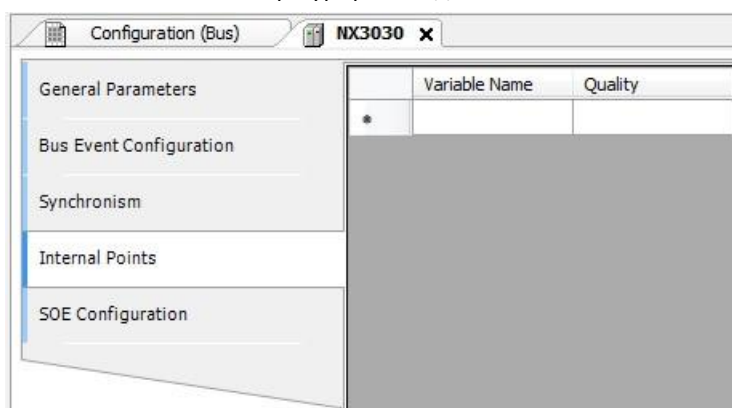


Рисунок 51: Экран конфигурации внутренних точек

Конфигурация	Описание	По умолчанию	Опции
Наименование переменной	Символьная переменная, в которой хранится значение внутренней точки	-	Принимаются переменные типа BOOL, WORD, DWORD, LWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL или DBP. Переменная может быть простой, массивом или элементом массива, а также может быть частью структуры.
Качество	Символьная переменная, хранящая качество внутренней точки	-	Переменные типа КАЧЕСТВО (LibRtuStandard), которые могут быть простыми, массивами или элементами массива и

			могут входить в состав структуры.
--	--	--	-----------------------------------

Таблица 49: Конфигурация внутренних точек

На рисунке ниже показан пример конфигурации двух внутренних точек:

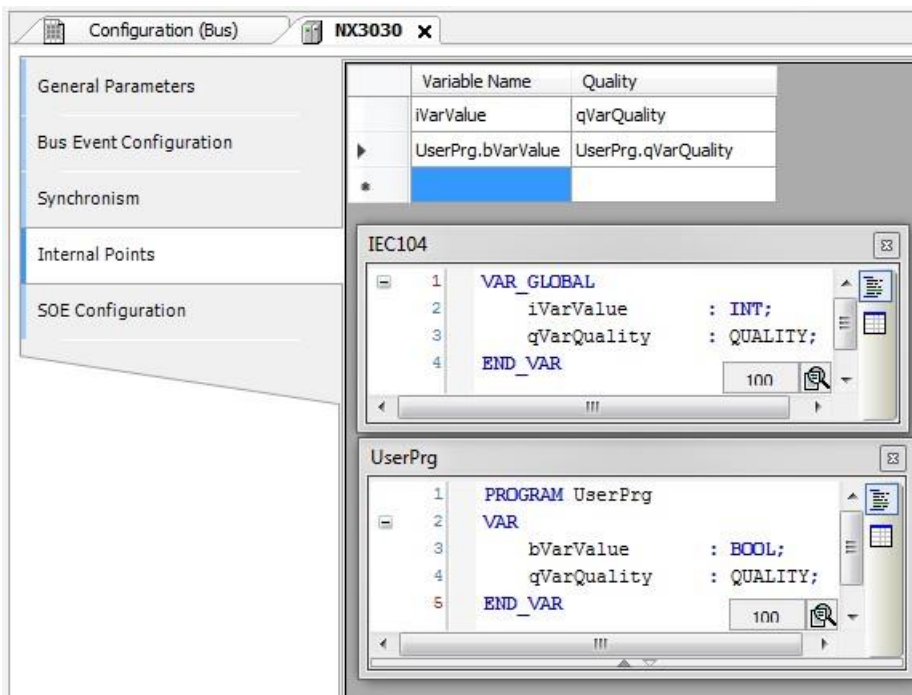


Рисунок 52: Пример конфигурации внутренних точек

5.1.5.1. Качественные конверсии

Качество внутренней точки - это информация об уровне доверия к значению, хранящемуся в этой точке. Качество может информировать, например, о том, что хранимое значение находится вне диапазона, или о том, что оно действительно, но не заслуживает доверия.

Стандарты IEC 61850, DNP3 и IEC104 имеют свои форматы представления информации о качестве точки. Серия Nexto, в свою очередь, имеет свой собственный формат качества (впрочем, весьма схожий с IEC 61850) под названием Internal Quality. Этот формат определяется типом QUALITY (библиотека LibRtuStandard) и используется для внутреннего хранения информации о качестве, позволяя осуществлять преобразование между протоколами без потери информации.

При отображении одной и той же точки связи между двумя драйверами преобразование качества автоматически выполняется в два этапа. Например: при отображении точки связи из драйвера DNP3 Client в драйвер IEC104 Server сначала качество будет преобразовано из формата DNP3 во внутренний формат (и сохранено внутри процессора), а затем из внутреннего формата в формат IEC104.

В следующих таблицах приведены преобразования собственных форматов протоколов во внутренний формат. В случае, если необходимо проконсультироваться по поводу преобразования между протоколами, необходимо провести анализ в два этапа, приведя каждую из таблиц к внутреннему формату и после соотнеся их.

ВНИМАНИЕ

В случае внутренних точек, сопоставленных с драйверами связи, не рекомендуется изменять значение флагов качества, не имеющих соответствия в данном протоколе (т.е. флагов, не описанных в следующих таблицах). Это приведет к генерации событий, равных предыдущему (но с более свежей временной меткой), и, таким образом, в зависимости от конфигурации, выбранной для режима передачи событий аналоговых входов, может перезаписать предыдущее событие, если оно еще не было доставлено в центр управления.

5. КОНФИГУРАЦИЯ

5.1.5.1.1. Внутреннее качество

Это и есть структура КАЧЕСТВА. В таблице подробно показан каждый из ее компонентов.

Бит	Наименование	Тип	Описание
0	ФЛАГ_ПЕРЕЗАПУСК	Логический	Флаг RESTART указывает на то, что данные не обновлялись полем с момента сброса устройства.
1	ФЛАГ_КОММ_ОТКАЗ	Логический	Указывает на наличие сбоя связи на пути между устройством-источником данных и устройством отчетов.
2	ФЛАГ_УДАЛЕННЫЙ_ЗАМЕЩЕННЫЙ	Логический	Если TRUE, то значения данных перезаписываются в удаленных устройствах связи.
3	ФЛАГ_ЛОКАЛЬНЫЙ_ЗАМЕЩЕННЫЙ	Логический	Если TRUE, то значение данных перезаписывается устройством, которое сгенерировало этот флаг. Такое поведение может быть вызвано работой диагностики или временным вмешательством человека.
4	ФЛАГ_ФИЛЬТР	Логический	Флаг, используемый для сигнализации и предотвращения события перегрузки канала связи в виде осцилляций (быстрых изменений) на дискретных входах.
5	ФЛАГ_ПЕРЕПОЛНЕНИЕ	Логический	Этот флаг должен указывать на проблему качества, на то, что значение атрибута, с которым связано качество, находится за пределами представления.
6	ФЛАГ_ОШИБКА_ССЫЛКИ	Логический	Этот флаг должен указывать на то, что значение не может быть корректным из-за отсутствия калибровочного эталона.
7	ФЛАГ_НЕСООТВЕТСТВУЮЩИЙ	Логический	Этот флаг должен указывать на то, что функция оценки обнаружила несоответствие.
8	ФЛАГ_ВНЕ_ДИАПАЗОНА	Логический	Этот флаг должен указывать на проблему качества, связанную с тем, что атрибут, с которым было связано качество, выходит за пределы заданных возможностей значений.
9	ФЛАГ_НЕТОЧНОСТИ	Логический	Этот флаг должен указывать на то, что значение не соответствует заявленной точности источника.
10	ФЛАГ_СТАРЫЕ_ДАННЫЕ	Логический	Значение кажется устаревшим. В случае если обновление не происходит в течение определенного периода времени.

5. КОНФИГУРАЦИЯ

11	ФЛАГ_НЕУДАЧИ	Логический	Этот флаг должен указывать на то, что функция наблюдения обнаружила внутренний или внешний сбой.
12	ФЛАГ_ОПЕРАТОР_ЗАБЛОКИРОВАН	Логический	Обновление заблокировано оператором.
Бит	Наименование	Тип	Описание
13	ФЛАГ_ТЕСТ	Логический	Это должен быть дополнительный идентификатор, по которому можно классифицировать значение как тестовое, которое не будет использоваться в оперативных целях.
14-15	РЕЗЕРВ	-	Reserved
16-17	ПРИГОДНОСТЬ	КАЧЕСТВО_ПРИГОДНОСТЬ	0 - Хорошо (достоверное значение, означает отсутствие аномальных условий) 1 - Недопустимо (Значение не соответствует значению IED) 2 - Резервный (зарезервировано) 3 - Под вопросом (Приведенное значение может не совпадать со значением IED)

Таблица 50: Структура КАЧЕСТВА

5.1.5.1.2. Преобразование IEC 60870-5-104

В приведенных ниже таблицах представлены соответственно цифровые, аналоговые преобразования и преобразования внутренних точек счетчиков по IEC 60870-5-104 серии Nexto, доступные для MT8500.

Внутренний-> IEC 60870-5-104 Цифровой		
Внутреннее качество		
Флаги	ПРИГОДНОСТЬ	IEC 60870-5-104 качество
ФЛАГ_ПЕРЕЗАПУСК	ЛЮБАЯ	НЕ ТЕМАТИЧЕСКИЙ
ФЛАГ_КОММ_ОТКАЗ	ЛЮБАЯ	НЕ ТЕМАТИЧЕСКИЙ
ФЛАГ_УДАЛЕННЫЙ_ЗАМЕЩЕННЫЙ	ЛЮБАЯ	ЗАМЕНА
ФЛАГ_ЛОКАЛЬНЫЙ_ЗАМЕЩЕННЫЙ	ЛЮБАЯ	ЗАМЕНА
ФЛАГ_ФИЛЬТР	ЛЮБАЯ	-
ФЛАГ_ПЕРЕПОЛНЕНИЕ	ЛЮБАЯ	-
ФЛАГ_ОШИБКА_ССЫЛКИ	ЛЮБАЯ	-
ФЛАГ_НЕСООТВЕТСТВУЮЩИЙ	ЛЮБАЯ	-
ФЛАГ_ВНЕ_ДИАПАЗОНА	ЛЮБАЯ	-
ФЛАГ_НЕТОЧНОСТИ	ЛЮБАЯ	-
ФЛАГ_СТАРЫЕ_ДАННЫЕ	ЛЮБАЯ	НЕ ТЕМАТИЧЕСКИЙ

5. КОНФИГУРАЦИЯ

ФЛАГ_НЕУДАЧИ	ЛЮБАЯ	НЕПРИГОДНАЯ
ФЛАГ_ОПЕРАТОР_ЗАБЛОКИРОВАН	ЛЮБАЯ	БЛОКИРОВАН
ФЛАГ_ТЕСТ	ЛЮБАЯ	-
ЛЮБАЯ	ПРИГОДНАЯ_НЕПРИГОДНАЯ	НЕПРИГОДНАЯ

Таблица 51: Преобразование цифровых точек по внутреннему стандарту IEC 60870-5-104

Внутренние -> IEC 60870-5-104 Аналог		
Внутреннее качество		
Флаги	ПРИГОДНОСТЬ	IEC 60870-5-104 качество
ФЛАГ_ПЕРЕЗАПУСК	ЛЮБАЯ	НЕ ТЕМАТИЧЕСКИЙ
ФЛАГ_КОММ_ОТКАЗ	ЛЮБАЯ	НЕ ТЕМАТИЧЕСКИЙ
ФЛАГ_УДАЛЕННЫЙ_ЗАМЕЩЕННЫЙ	ЛЮБАЯ	ЗАМЕНА
ФЛАГ_ЛОКАЛЬНЫЙ_ЗАМЕЩЕННЫЙ	ЛЮБАЯ	ЗАМЕНА
Внутренние -> IEC 60870-5-104 Аналог		
Внутреннее качество		
Флаги	ПРИГОДНОСТЬ	IEC 60870-5-104 качество
ФЛАГ_ФИЛЬТР	ЛЮБАЯ	-
ФЛАГ_ПЕРЕПОЛНЕНИЯ	ЛЮБАЯ	ПЕРЕПОЛНЕНИЕ
ФЛАГ_ОШИБКА_ССЫЛКИ	ЛЮБАЯ	НЕПРИГОДНАЯ
ФЛАГ_НЕСООТВЕТСТВУЮЩИЙ	ЛЮБАЯ	НЕПРИГОДНАЯ
ФЛАГ_ВНЕ_ДИАПАЗОНА	ЛЮБАЯ	ПЕРЕПОЛНЕНИЕ
ФЛАГ_НЕТОЧНОСТИ	ЛЮБАЯ	НЕПРИГОДНАЯ
ФЛАГ_СТАРЫЕ_ДААННЫЕ	ЛЮБАЯ	НЕ ТЕМАТИЧЕСКИЙ
ФЛАГ_НЕУДАЧИ	ЛЮБАЯ	НЕПРИГОДНАЯ
ФЛАГ_ОПЕРАТОР_ЗАБЛОКИРОВАН	ЛЮБАЯ	БЛОКИРОВАН
ФЛАГ_ТЕСТ	ЛЮБАЯ	-
ЛЮБАЯ	ПРИГОДНАЯ_НЕПРИГОДНАЯ	НЕПРИГОДНАЯ

Таблица 52: Преобразование аналоговых точек по внутреннему стандарту IEC 60870-5-104

Внутренние -> IEC 60870-5-104 Счетчики		
Внутреннее качество		
Флаги	ПРИГОДНОСТЬ	IEC 60870-5-104 качество
ФЛАГ_ПЕРЕЗАПУСК	ЛЮБАЯ	-
ФЛАГ_КОММ_ОТКАЗ	ЛЮБАЯ	-

5. КОНФИГУРАЦИЯ

ФЛАГ_УДАЛЕННЫЙ_ЗАМЕЩЕННЫЙ	ЛЮБАЯ	-
ФЛАГ_ЛОКАЛЬНЫЙ_ЗАМЕЩЕННЫЙ	ЛЮБАЯ	-
ФЛАГ_ФИЛЬТР	ЛЮБАЯ	-
ФЛАГ_ПЕРЕПОЛНЕНИЕ	ЛЮБАЯ	ПЕРЕПОЛНЕНИЕ
ФЛАГ_ОШИБКА_ССЫЛКИ	ЛЮБАЯ	-
ФЛАГ_НЕСООТВЕТСТВУЮЩИЙ	ЛЮБАЯ	-
ФЛАГ_ВНЕ_ДИАПАЗОНА	ЛЮБАЯ	-
ФЛАГ_НЕТОЧНОСТИ	ЛЮБАЯ	-
ФЛАГ_СТАРЫЕ_ДАННЫЕ	ЛЮБАЯ	-
ФЛАГ_НЕУДАЧИ	ЛЮБАЯ	НЕПРИГОДНАЯ
ФЛАГ_ОПЕРАТОР_ЗАБЛОКИРОВАН	ЛЮБАЯ	-
ФЛАГ_ТЕСТ	ЛЮБАЯ	-
ЛЮБАЯ	ПРИГОДНАЯ_НЕПРИГОДНАЯ	НЕПРИГОДНАЯ

Таблица 53: Преобразование счетчиков по внутреннему стандарту IEC 60870-5-104

5.1.5.1.3. Внутреннее качество MODBUS

Поскольку стандарт MODBUS не определяет типы качества для каждой точки, а служит для помощи в использовании коммуникационной диагностики каждой точки, MasterTool позволяет отображать переменные качества через внутреннюю собственную структуру для каждой точки MODBUS. В таблице ниже описаны типы качества, которые может принимать каждая точка MODBUS..

Итоговое качество	Итоговая ПРИГОДНОСТЬ	Описание
ФЛАГОВЫЙ_ПЕРЕЗАПУСК	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Начальное значение. Точка никогда не обновлялась.
-	ПРИГОДНОСТЬ_ХОРОШАЯ	Связь в порядке. Точка обновлена.
Итоговое качество	Итоговая ПРИГОДНОСТЬ	Описание
ФЛАГ_КОММ_ОТКАЗ И ФЛАГ_ПЕРЕЗАПУСК	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Ошибка связи. Точка так и не была обновлена.
ФЛАГ_КОММ_ОТКАЗ И ФЛАГ_СТАРЫЕ_ДАННЫЕ	ПРИГОДНОСТЬ_ПОД ВОПРОСОМ	Произошла ошибка, но точка была обновлена и теперь имеет старое значение.
ФЛАГ_ОТКАЗ И ФЛАГ_ПЕРЕЗАПУСК	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Получен ответ исключения, и точка сохранила свое начальное значение.
ФЛАГ_ОТКАЗ И ФЛАГ_СТАРЫЕ_ДАННЫЕ	ПРИГОДНОСТЬ_ПОД ВОПРОСОМ	Получен ответ исключения, но точка имеет действительное старое значение.
ФЛАГ_ПЕРЕЗАПУСК И ФЛАГ_СТАРЫЕ_ДАННЫЕ	ПРИГОДНОСТЬ_ПОД ВОПРОСОМ	Устройство остановлено. Точка имеет старое значение.

Таблица 54: MODBUS Качество

5.1.5.1.4. Качество модулей ввода/вывода локальной шины

Чтобы помочь в использовании диагностики каждой точки ввода/вывода, MasterTool автоматически создает структуру качества для каждого модуля локальной шины, используемого в проекте ПЛК, через собственную внутреннюю структуру, доступную по структуре QUALITY, доступной в GVL IOQualities.

В таблице ниже описаны типы качества для каждой точки входа и выхода.

Дополнительную информацию можно найти на сайте [GVL IOQualities](#).

Диагностика	Итоговое качество	Итоговая ПРИГОДНОСТЬ	Описание
Безразлично	ФЛАГ_ПЕРЕЗАПУСК	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Качество имеет это значение до того, как было прочитано или записано в первый раз.
Нет	-	ПРИГОДНОСТЬ_ХОРОШАЯ	Связь ОК. Точка обновлена.
Нет	ФЛАГ_СТАРЫЕ_ДАННЫЕ И ФЛАГ_НЕИСПРАВНОСТЬ	ПРИГОДНОСТЬ_ПОД ВОПРОСОМ	Нерабочий модуль. Однако данные были считаны или записаны хотя бы один раз.
bOverRange или bUnder-Range	ФЛАГ_ИЗ_ДИАПАЗОНА	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Значение выше или ниже допустимого диапазона входных сигналов модуля.
bInputNotEnable или bOutputNotEnable	ФЛАГ_ОПЕРАТОРА_ЗАБЛОКИРОВАНО	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Вход/выход не разрешен.
bOpenLoop	ФЛАГ_ОТКАЗА	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Открытый цикл в модуле ввода.
bFatalError	ФЛАГ_ОТКАЗА	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Аппаратный фатальный сбой.
bNoExternalSupply	ФЛАГ_ОТКАЗА	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Внешний источник питания находится ниже минимального рабочего предела.
bShortCircuit или bOutputShortCircuit	ФЛАГ_НЕИСПРАВНОСТИ	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Короткое замыкание на выходе.
bCalibrationError	ФЛАГ_НЕТОЧНЫЙ	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Ошибка калибровки.
bColdJunctionSensorError	ФЛАГ_НЕТОЧНЫЙ	ПРИГОДНОСТЬ_НЕПРИГОДНАЯ	Ошибка датчика холодного спая.

Таблица 55: Качество модулей ввода/вывода

5.1.5.1.5. Качество модулей ввода/вывода PROFIBUS

В отличие от локальной шины, MasterTool не создает автоматически структуры качества модулей PROFIBUS, а также не обновляет эти структуры в ПЛК. Поэтому создание и циклическое обновление качества модулей PROFIBUS является обязанностью пользователя.

5. КОНФИГУРАЦИЯ

Для помощи в разработке таких приложений приведены практические примеры на языке ST для основных модулей PROFIBUS (DI, DO, AI, AO) на базе PROFIBUS-ведомых устройств Nexto Serie (NX5110). Пользователю предлагается внести любые необходимые изменения и адаптации для своего применения.

ВНИМАНИЕ

Для корректной работы представленных в последовательности процедур необходимо включить функцию Diagnose на ведомых устройствах PROFIBUS.

Разработка процедуры обновления точек качества модулей ввода/вывода PROFIBUS должна начинаться с объявления и инициализации переменных качества, с GVL:

```
VAR_GLOBAL
QUALITY_PB_NX1005_I: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
                                FLAGS:= (FLAG_RESTART:= TRUE)); ( ПРИГОДНОСТЬ:=
                                ПРИГОДНОСТЬ_НЕПРИГОДНАЯ,
                                ФЛАГИ:= ((ФЛАГ_ПЕРЕЗАПУСК:= TRUE));

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

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

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

END_VAR
```

5.1.5.1.6. Качество цифровых входов PROFIBUS

```
// Обновление качества цифровых входов PROFIBUS, модуль NX1005

// В случае успешной связи с ведомым устройством PROFIBUS (адрес = 99) ...
ЕСЛИ DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE ТОГДА
    // Ждет, пока ведомое устройство PROFIBUS станет готовым к обмену данными и диагностике
    // (Необходимо подождать, чтобы избежать генерации недопустимого качества)
    ЕСЛИ DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
        DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND
        DG_NX5110.tPbusHeadA.wIdentNumber > 0 ТОГДА
        QUALITY_PB_NX1005_I.FLAGS.FLAG_COMM_FAIL:= FALSE;
        // Если на шине присутствует модуль (слот = 2) и
        // если нет проблем с конфигурацией модулей (общая) и
        // если нет проблем с конфигурацией этого модуля (специфических) и // если нет фатальной ошибки,
        идентифицированной модулем ...
        ЕСЛИ (DG_NX5110.tPbusHeadA.dwModuleNotPresent И SHL(1, 2)) = 0 И
            DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE И
            DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bConfigMismatch = FALSE И
            DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bFatalError = FALSE ТОГДА
            QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_GOOD;
            QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART:= FALSE;
            QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= FALSE;
            QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= FALSE;
        ТАКЖЕ
            QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_INVALID; QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= TRUE;
```

5. КОНФИГУРАЦИЯ

```
// Если точка хоть раз обновлялась ...
ЕСЛИ НЕ QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART ТОГДА
    QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= TRUE; END_IF
END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS ...
ELSE
    QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_INVALID;
    QUALITY_PB_NX1005_I.FLAGS.FLAG_COMM_FAIL:= TRUE;
    QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= FALSE; // Если точка хоть раз
    обновлялась ...
    ЕСЛИ НЕ QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART ТОГДА
        QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= TRUE; END_IF
END_IF
```

5.1.5.1.7. Качество цифрового выхода PROFIBUS

```
// Обновление качества цифрового выхода PROFIBUS, модуль NX1005

// В случае успешной связи с ведомым устройством PROFIBUS (адрес = 99) ...
ЕСЛИ DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE ТОГДА
    // Ждет, пока ведомое устройство PROFIBUS станет готовым к обмену данными и диагностике
    // (Необходимо подождать, чтобы избежать генерации недопустимого качества)
    ЕСЛИ DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE И
        DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE И
        DG_NX5110.tPbusHeadA.wIdentNumber > 0 ТОГДА
            QUALITY_PB_NX1005_O.FLAGS.FLAG_COMM_FAIL:= FALSE;
            // Если на шине присутствует модуль (слот = 2) и
            // если нет проблем с конфигурацией модуля (общее) и
            // если нет проблем с конфигурацией данного модуля (специфические) и
            // если модуль не идентифицировал фатальную ошибку и
            // если нет индикации короткого замыкания выходов и
            // если нет индикации отсутствия внешнего питания ...
            ЕСЛИ (DG_NX5110.tPbusHeadA.dwModuleNotPresent И SHL(1, 2)) = 0 И
                DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE И
                DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bConfigMismatch = FALSE И
                DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tGeneral.bFatalError = FALSE И
                DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tDetailed.bOutputShortCircuit = FALSE И
                DG_NX1005_24_Vdc_8_DO_Trans_8_DI.tDetailed.bNoExternalSupply = FALSE ТОГДА
                    QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_GOOD;
                    QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART:= FALSE;
                    QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= FALSE;
                    QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= FALSE;
            ТАКЖЕ
                QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_INVALID;
                QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= TRUE;
            // Если точка хоть раз обновлялась ...
```

5. КОНФИГУРАЦИЯ

```
    ЕСЛИ НЕ QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART ТОГДА
        QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= TRUE;
    END_IF
END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS ...
ТАКЖЕ
QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX1005_O.FLAGS.FLAG_COMM_FAIL:= TRUE;
QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= FALSE; // Если точка
хоть раз обновлялась ...
ЕСЛИ НЕ QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART ТОГДА QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:=
    TRUE;
END_IF
END_IF
```

5.1.5.1.8. Качество аналоговых входов PROFIBUS

```
// Обновление качества аналоговых входов PROFIBUS, модуль NX6000
// В случае успешной связи с ведомым устройством PROFIBUS (адрес = 99) ...
ЕСЛИ DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE ТОГДА
// Ждет, пока ведомое устройство PROFIBUS станет готовым к обмену данными и диагностике
// (Необходимо подождать, чтобы избежать генерации недопустимого качества)
ЕСЛИ DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE И
    DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE И
    DG_NX5110.tPbusHeadA.wIdentNumber > 0 ТОГДА
QUALITY_PB_NX6000.FLAGS.FLAG_COMM_FAIL:= FALSE;
// Если на шине присутствует модуль (слот = 3) и
// если нет проблем с конфигурацией модуля (общее) и
// если нет проблем с конфигурацией этого модуля (специфических) и
// если модуль не идентифицирует фатальную ошибку и
// если нет индикации ошибки калибровки и // если нет индикации ошибки превышения/превышения
диапазона и
// если нет индикации ошибки входа в разомкнутом контуре ...
ЕСЛИ (DG_NX5110.tPbusHeadA.dwModuleNotPresent И SHL(1, 3)) = 0 И
    DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE И
    DG_NX6000_8_AI_Voltage_Current.tGeneral.bConfigMismatch = FALSE И
    DG_NX6000_8_AI_Voltage_Current.tGeneral.bFatalError = FALSE И
    DG_NX6000_8_AI_Voltage_Current.tGeneral.bCalibrationError = FALSE И
    DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOverRange = FALSE И
    DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bUnderRange = FALSE И
    DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop = FALSE
    ТОГДА
    QUALITY_PB_NX6000.VALIDITY:= VALIDITY_GOOD;
    QUALITY_PB_NX6000.FLAGS.FLAG_RESTART:= FALSE;
    QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
    QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= FALSE;
    QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= FALSE;
```


5. КОНФИГУРАЦИЯ

```
QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= FALSE;
ТАКЖЕ
// Условие включения указания на неточность
// (сначала проверяется, так как недействительная достоверность должна преобладать)
ЕСЛИ DG_NX6000_8_AI_Voltage_Current.tGeneral.bCalibrationError = TRUE ТОГДА
    QUALITY_PB_NX6000.VALIDITY:= VALIDITY_QUESTIONABLE;
    QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= TRUE;
ТАКЖЕ
    QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= FALSE;
END_IF
// Условие включения индикации выхода за пределы диапазона
// (сначала проверяется, так как должна преобладать недействительная действительность)
ЕСЛИ DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOverRange =
TRUE ИЛИ
DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bUnderRange =
TRUE ТОГДА
    QUALITY_PB_NX6000.VALIDITY:= VALIDITY_QUESTIONABLE;
    QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= TRUE;
ТАКЖЕ
    QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= FALSE;
END_IF
// Условие включения общей индикации отказа (приоритет)
ЕСЛИ (DG_NX5110.tPbusHeadA.dwModuleNotPresent И SHL(1, 3)) > 0 ИЛИ
DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = TRUE ИЛИ
DG_NX6000_8_AI_Voltage_Current.tGeneral.bConfigMismatch = TRUE ИЛИ
DG_NX6000_8_AI_Voltage_Current.tGeneral.bFatalError = TRUE ИЛИ
DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop = TRUE
ТОГДА
    QUALITY_PB_NX6000.VALIDITY:= VALIDITY_INVALID;
    QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= TRUE;
    // Если точка хоть раз обновлялась ...
    ЕСЛИ НЕ QUALITY_PB_NX6000.FLAGS.FLAG_RESTART И
    НЕ DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop
ТОГДА
        QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= TRUE; END_IF
ТАКЖЕ
    QUALITY_PB_NX6000.FLAGS.FLAG_RESTART:= FALSE;
    QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
    QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= FALSE; END_IF
END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS ...
ТАКЖЕ
QUALITY_PB_NX6000.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX6000.FLAGS.FLAG_COMM_FAIL:= TRUE;
QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
// Если точка хоть раз обновлялась ...
```

5. КОНФИГУРАЦИЯ

```
ЕСЛИ НЕ QUALITY_PB_NX6000.FLAGS.FLAG_RESTART И  
    HE DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop ТОГДА  
    QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= TRUE;  
END_IF  
END_IF
```

5.1.5.1.9.PROFIBUS Качество аналогового выхода

```
// Обновление качества аналогового выхода по сети PROFIBUS, модуль NX6100
```

5. КОНФИГУРАЦИЯ

```
// В случае успешной связи с ведомым устройством PROFIBUS (адрес = 99) ...
ЕСЛИ DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE ТОГДА
// Ждет, пока ведомое устройство PROFIBUS станет готовым к обмену данными и диагностике
// (Необходимо подождать, чтобы избежать генерации недопустимого качества)
ЕСЛИ DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE И
    DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE И
    DG_NX5110.tPbusHeadA.wIdentNumber > 0 ТОГДА
QUALITY_PB_NX6100.FLAGS.FLAG_COMM_FAIL:= FALSE;
// Если на шине присутствует модуль (слот = 4) и
// если нет проблем с конфигурацией модуля (общее) и
// если нет проблем с конфигурацией этого модуля (специфических) и
// если модуль не идентифицирует фатальную ошибку и
// если нет индикации ошибки калибровки и
// если нет индикации отсутствия внешнего источника питания и
// если нет индикации ошибки выхода в разомкнутом контуре и // если нет индикации короткого замыкания
выходов ...
ЕСЛИ (DG_NX5110.tPbusHeadA.dwModuleNotPresent И SHL(1, 4)) = 0 И
    DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE И
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bConfigMismatch = FALSE И
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bFatalError = FALSE И
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bCalibrationError = FALSE И
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bNoExternalSupply = FALSE И
    DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop = FALSE И
    DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bShortCircuit =
    FALSE ТОГДА
    QUALITY_PB_NX6100.VALIDITY:= VALIDITY_GOOD;
    QUALITY_PB_NX6100.FLAGS.FLAG_RESTART:= FALSE;
    QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
    QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= FALSE;
    QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= FALSE;
ТАКЖЕ
// Условие включения указания на неточность
// (сначала проверяется, так как недействительная достоверность должна преобладать)
ЕСЛИ DG_NX6100_4_AO_Voltage_Current.tGeneral.bCalibrationError = TRUE ТОГДА
    QUALITY_PB_NX6100.VALIDITY:= VALIDITY_QUESTIONABLE;
    QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= TRUE;
ELSE
    QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= FALSE;
END_IF
// Условие включения общей индикации отказа (приоритет)
ЕСЛИ (DG_NX5110.tPbusHeadA.dwModuleNotPresent И SHL(1, 4)) > 0 ИЛИ
    DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = TRUE ИЛИ
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bConfigMismatch = TRUE ИЛИ
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bFatalError = TRUE ИЛИ
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bNoExternalSupply = TRUE ИЛИ
    DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop = TRUE
```

5. КОНФИГУРАЦИЯ

```
ИЛИ
  DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bShortCircuit =
  TRUE ТОГДА
    QUALITY_PB_NX6100.VALIDITY:= VALIDITY_INVALID;
    QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= TRUE;
    // Если точка хоть раз обновлялась ...
    ЕСЛИ НЕ QUALITY_PB_NX6100.FLAGS.FLAG_RESTART И НЕ
      DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop
    ТОГДА
      QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= TRUE; END_IF
    ТАКЖЕ
      QUALITY_PB_NX6100.FLAGS.FLAG_RESTART:= FALSE;
      QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
      QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= FALSE; END_IF
  END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS ...
ТАКЖЕ
QUALITY_PB_NX6100.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX6100.FLAGS.FLAG_COMM_FAIL:= TRUE;
QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
// Если точка хоть раз обновлялась...
ЕСЛИ НЕ QUALITY_PB_NX6100.FLAGS.FLAG_RESTART И
  НЕ DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop ТОГДА
  QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= TRUE;
END_IF
END_IF
```

5.2. Конфигурация последовательных интерфейсов

5.2.1. COM 1

Интерфейс связи COM 1 состоит из гнезда DB9 для стандарта RS-232C. Он позволяет осуществлять связь "точка-точка" (или в сети с помощью конвертера) в открытых протоколах MODBUS RTU slave или MODBUS RTU master. Ниже описаны параметры, которые должны быть сконфигурированы для правильного функционирования приложения.

При использовании протокола MODBUS master/slave некоторые из этих параметров (например, Serial Mode, Биты данных, RX Threshold и Serial Events) автоматически настраиваются программой MasterTool для корректной работы этого протокола.

Конфигурация	Описание	По умолчанию	Опции
Тип последовательного интерфейса	Конфигурация типа последовательного канала.	RS-232C	RS-232C
Скорость передачи данных	Конфигурация скорости порта последовательной связи.	115200	200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Четность	Конфигурация четности последовательного порта.	None	Нечетные Четные Пространство Отметка Нет

5. КОНФИГУРАЦИЯ

Биты данных	Установка количества символьных битов последовательного порта.	8	5, 6, 7 и 8
Стоп-биты	Устанавливает количество стоповых битов последовательного порта.	1	1, 1.5 и 2
Конфигурация	Описание	По умолчанию	Опции
Последовательный режим	Устанавливает конфигурацию режима работы последовательного порта.	Нормальный режим	<ul style="list-style-type: none"> - Расширенный режим: Расширенный режим работы, в котором передается информация о полученном кадре данных. - Нормальный режим: Нормальный режим работы последовательной связи.

Таблица 56: Стандартная конфигурация последовательного интерфейса RS-232

Примечания:

Расширенный режим: В этом режиме работы последовательной связи предоставляется информация о полученном кадре данных. Доступна следующая информация:

- Один байт для принятых данных (RX_CHAR : BYTE): Сохраняет пять, шесть, семь или восемь битов из принятых данных в зависимости от конфигурации последовательной связи.
- Один байт для ошибок сигнала (RX_ERROR : BYTE): Имеет формат, описанный ниже:
 - Бит 0: 0 - символ в битах 0-7 действителен. 1 - символ в битах 0-7 не действителен (или не может быть действительным) из-за проблем, указанных в битах 10-15.
 - Бит 1: не используется.
 - Бит 2: не используется.
 - Бит 3: Ошибка прерывания UART. Последовательный вход оставался в логическом 0 (пробел) в течение времени, превышающего символ (стартовый бит + биты данных + бит четности + стоповые биты).
 - Бит 4: Ошибка кадра UART. Логический 0 (пробел) был считан, когда ожидался первый стоп-бит, а должен быть логический 1 (метка).
 - Бит 5: Ошибка четности UART. Считанный бит четности не соответствует вычисленному.
 - Бит 6: UART overrun error. Данные были потеряны во время чтения FIFO UART. Новые символы были получены до того, как были удалены более поздние. Эта ошибка будет указана только в первом считанном символе после индикации ошибки превышения. Это означает, что были потеряны некоторые старые данные.
 - Бит 7: Ошибка выхода за пределы линии RX. Этот символ был записан при завершении линии RX, перезаписав непрочитанные символы.
- Два байта для сигнала временной метки (RX_TIMESTAMP : WORD): Указывает время молчания в интервале от 0 до 65535, используя в качестве базы 10 us. Насыщение происходит через 655,35 мс, если время молчания превышает 65535 единиц. RX_TIMESTAMP символа измеряет время от эталона, в качестве которого может выступать любой из трех приведенных ниже вариантов:
 - В большинстве случаев конец последующего символа.
 - Конфигурация последовательного порта.
 - Окончание последовательного обмена данными с использованием ФБ SERIAL_TX, другими словами, когда по линии передается последний символ.

Кроме измерения тишины между символами, важен также параметр RX_TIMESTAMP, который измеряет время тишины последнего символа на линии RX. Измерение времени молчания важно для корректной реализации протокола, например, MODBUS RTU. Этот протокол определяет межкадровый интервал более 3,5 символов и межбайтовый интервал менее 1,5 символов.

5. КОНФИГУРАЦИЯ

Биты данных: Конфигурация Биты данных последовательных интерфейсов ограничивает поля Стоп-биты и Четность связи. Поэтому количество стоповых битов и метод контроля четности будут зависеть от количества бит данных.

Биты данных	Стоп-биты	Четность
5	1, 1.5	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТЬ, ЧЕТНОСТЬ, ЧЕТНОСТЬ ВСЕГДА ОДИН, ЧЕТНОСТЬ ВСЕГДА НОЛЬ
6	1, 2	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТИ, ЧЕТНОСТИ, ЧЕТНОСТЬ ВСЕГДА ЕДИНИЦА, ЧЕТНОСТЬ ВСЕГДА НОЛЬ
7	1, 2	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТИ, ЧЕТНОСТИ, ЧЕТНОСТИ ВСЕГДА ОДИН, ЧЕТНОСТИ ВСЕГДА НОЛЬ
Биты данных	Стоп-биты	Четность
8	1, 2	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТИ, ЧЕТНОСТИ, ЧЕТНОСТИ ВСЕГДА ОДИН, ЧЕТНОСТИ ВСЕГДА НОЛЬ

Таблица 57: Специфические 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
Дополнительные параметры порта			
Автоответ	Выполняет управление запросом на передачу команды по интерфейсу RS-232C.	RTS выкл	<ul style="list-style-type: none"> - RTS: Включается в начале передачи и перезапускается как можно быстрее после ее окончания. Например, управление внешним преобразователем RS-232/RS-485. - RTS Off: Всегда отключен. - RTS On: Всегда включен. - RTS/CTS: В случае отключения CTS включается RTS. Поэтому необходимо дождаться включения CTS, чтобы снова начать передачу, и перезапустить RTS, как можно быстрее, в конце передачи. Например, управление радиомодемами

5. КОНФИГУРАЦИЯ

			осуществляется с помощью одного и того же сигнала модема. - Ручной RTS: пользователь отвечает за все управляющие сигналы.
Пороговое значение UART RX	Количество байтов, которое должно быть получено для генерации нового прерывания UART. Низкие значения делают TIMESTAMP более точным при использовании режима EXTENDED и минимизируют ошибки превышения. Однако слишком низкие значения могут привести к нескольким прерываниям, задерживающим работу процессора.	8	1, 4, 8 и 14
Серийные события			
RX на TX	Если значение true, то все принятые байты во время передачи будут разряжаться, а не поступать на линию RX. Используется для отключения полнодуплексной работы интерфейса RS-232C.	Отключено	- Включено: Конфигурация включена - Отключено: Конфигурация отключена
Конфигурация	Описание	По умолчанию	Опции
Событие RX DCD	Если true, генерирует внешнее событие, связанное с изменением сигнала DCD.	Включено	- Включено: Конфигурация включена - Отключено: Конфигурация отключена
Событие RX CTS	Если true, то генерируется внешнее событие, связанное с изменением сигнала CTS.	Включено	- Включено: Конфигурация включена - Отключено: Конфигурация отключена

Таблица 58: Расширенные конфигурации стандартного последовательного интерфейса RS-232

Примечания:

RX на TX: Этот расширенный параметр действителен для установок RS-232C и RS-422.

5. КОНФИГУРАЦИЯ

Событие RX DCD: Внешние события, такие как сигнал DCD COM 1 процессоров NX3010, NX3020, NX3030, могут быть связаны только с задачами пользовательского профиля проекта, более подробная информация приведена в руководстве пользователя MasterTool IEC XE - MU299609.

RX CTS Event: Внешние события, такие как сигнал CTS COM 1 процессоров NX3010, NX3020, NX3030, могут быть связаны только с задачами пользовательского профиля проекта, более подробная информация приведена в руководстве пользователя MasterTool IEC XE - MU299609.

5.2.2. COM 2

Конфигурация Биты данных последовательных интерфейсов ограничивает поля Стоп-биты и Четность связи. Поэтому количество стоповых битов и метод контроля четности будут зависеть от количества бит данных.

В таблице ниже приведены допустимые конфигурации интерфейсов.

Биты данных	Стоп-биты	Четность
5	1, 1.5	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТИ, ЧЕТНОСТИ, ЧЕТНОСТИ ВСЕГДА ОДИН, ЧЕТНОСТИ ВСЕГДА НОЛЬ
6	1, 2	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТИ, ЧЕТНОСТИ, ЧЕТНОСТИ ВСЕГДА ОДИН, ЧЕТНОСТИ ВСЕГДА НОЛЬ
7	1, 2	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТИ, ЧЕТНОСТИ, ЧЕТНОСТИ ВСЕГДА ОДИН, ЧЕТНОСТИ ВСЕГДА НОЛЬ
8	1, 2	НЕТ ЧЕТНОСТИ, НЕЧЕТНОСТИ, ЧЕТНОСТИ, ЧЕТНОСТИ ВСЕГДА ОДИН, ЧЕТНОСТИ ВСЕГДА НОЛЬ

Таблица 59: Специфические конфигурации

5.2.2.1. 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 UART. Низкие значения делают TIMESTAMP более точным при использовании режима EXTENDED и минимизируют ошибки превышения. Однако слишком низкие значения могут вызвать несколько прерываний,	8	1, 4, 8 и 14

5. КОНФИГУРАЦИЯ

	задерживающих работу процессора.		
--	----------------------------------	--	--

Таблица 60: Расширенные конфигурации стандартного последовательного интерфейса RS-485/RS-

4225.3. Конфигурация интерфейсов Ethernet

Процессоры Nexto могут предоставлять больше локальных интерфейсов Ethernet. Процессор NX3030 имеет интерфейсы NET 1 и NET 2. Помимо локальных интерфейсов Ethernet, в серии Nexto предусмотрены и удаленные интерфейсы Ethernet за счет включения модуля NX5000. Модули NX5000 имеют только интерфейс NET 1.

5.3.1. Локальные интерфейсы 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: MODBUS TCP Client, MODBUS RTU через TCP Client, MODBUS TCP Server и MODBUS RTU через TCP Server.

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

Конфигурация	Описание	По умолчанию	Опции
Получение IP-адреса в автоматическом режиме	Включение функции DHCP-клиента на устройстве для автоматического назначения IP-адреса	Немаркированные	Маркированные или немаркированные
IP-адрес	IP-адрес контроллера в шине Ethernet	192.168.15.1	1.0.0.1 до 223.255.255.254
Маска подсети	Маска подсети контроллера в шине Ethernet	255.255.255.0	128.0.0.0 до 255.255.255.252
Адрес шлюза	Адрес шлюза контроллера в шине Ethernet	192.168.15.253	1.0.0.1 до 223.255.255.254

Таблица 61: Конфигурация локальной сети 1

5.3.1.2. NET 2

Интерфейс NET 2 состоит из коммуникационного разъема RJ45 стандарта 10/100Base-TX. Он позволяет осуществлять связь "точка-точка" или сетевую связь по следующим открытым протоколам: MODBUS TCP Client, MODBUS RTU через TCP Client, MODBUS TCP Server и MODBUS RTU через TCP Server.

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

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

Таблица 62: Конфигурация NET 2

5. КОНФИГУРАЦИЯ

ВНИМАНИЕ

Невозможно сконфигурировать более одного Ethernet-интерфейса процессора в одной подсети, и такой тип конфигурации блокируется инструментом MasterTool. Поэтому каждый Ethernet-интерфейс должен быть сконфигурирован в отдельной подсети.

5.3.2. Удаленные интерфейсы Ethernet

5.3.2.1. 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: MODBUS TCP Client, MODBUS RTU через TCP Client, MODBUS TCP Server и MODBUS RTU через TCP Server.

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

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

Таблица 63: Конфигурация удаленной сети NET 1

5.3.3. Зарезервированные TCP/UDP-порты

Следующие TCP/UDP-порты интерфейсов Ethernet, как локальные, так и удаленные, используются службами ЦПУ (в зависимости от доступности в соответствии с таблицей "Протоколы") и, следовательно, зарезервированы и не должны использоваться пользователем.

Сервис	TCP	UDP
Веб-страница диагностики	80	-
Простой сетевой протокол времени	-	123
Простой управляющий протокол	-	161
MODBUS TCP	502*	-
Сервис	TCP	UDP
MasterTool MT8500	1217*	1740:1743
SQL Сервер	1433	-
Лёгкий сетевой протокол	1883* / 8883*	-
EtherNet/IP	44818	2222
IEC 60870-5-104	2404*	-

OPC UA	4840	-
WEBVISU	8080	-
CODESYS ARTI	11740	-
PROFINET	-	34964

Таблица 64: Зарезервированные порты TCP/UDP

* Порт по умолчанию, но может быть изменен пользователем.

5.4. Конфигурация модулей NX5000

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

Каналы Ethernet модулей NX5000 могут использоваться по отдельности или объединяться в пары NIC Teaming. Пары NIC Teaming используются для обеспечения резервирования каналов Ethernet. См. раздел "Общие характеристики".

Примером типичного применения модуля NX5000 является организация резервированной сети HSDN (High Speed Deterministic Network) для связи между различными ПЛК. Через эту сеть несколько ПЛК могут обмениваться сообщениями в полностью изолированной сети, что обеспечивает детерминизм и быстроту обмена данными. Кроме того, резервирование этой сети с помощью пар NIC Teaming обеспечивает отличную доступность. Для построения резервированной сети HSDN необходимо установить два модуля NX5000. На рисунке ниже показан пример резервированной сети HSDN с использованием двух модулей NX5000.

На рисунке также показан пример с одним изолированным модулем NX5000 (без резервирования NIC Teaming), установленным справа от других модулей.

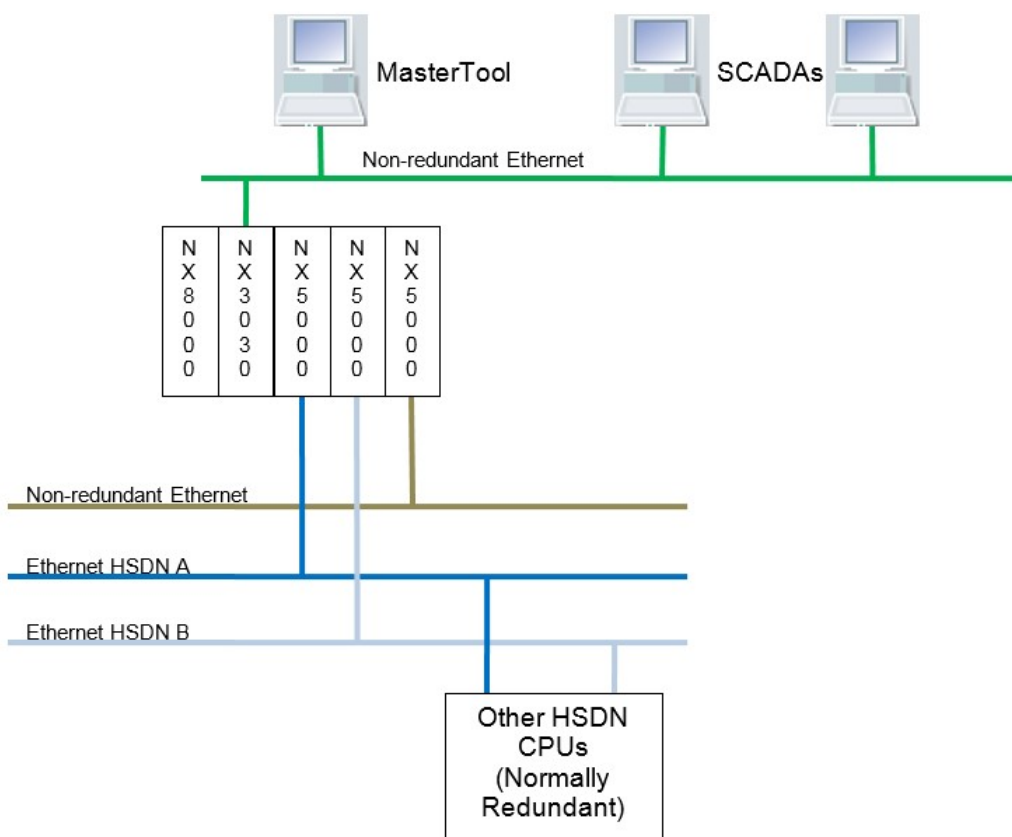


Рисунок 53: Простые и резервные сети Ethernet с использованием NX5000

5. КОНФИГУРАЦИЯ

Два первых модуля NX5000 из стойки объединяются в резервную пару NIC Teaming, соединенную в двух разных коммутаторах (Ethernet HSDN A и Ethernet HSDN B). В определенный момент эти два коммутатора должны быть соединены между собой, чтобы обеспечить связь между двумя портами NIC Teaming и повысить доступность (от двойных отказов).

Подобные архитектуры Ethernet обеспечивают отличную доступность к отказам портов, кабелей и коммутаторов Ethernet.

Набор из двух портов Ethernet, образующих пару NIC Teaming, представляет уникальный IP-адрес, подключенный к этой паре портов. Таким образом, клиенту, например, SCADA или MasterTool, подключенному к серверу на ПЛК, не нужно заботиться о смене IP-адреса в случае отказа одного из портов NIC Teaming.

Диагностика указывает на возможные сбои, которые могут возникнуть в любом из портов пары NIC Teaming.

Для того чтобы объединить два модуля NX5000 в резервную пару, необходимо, чтобы эти два модуля занимали соседние позиции в стойке объединительной панели и был установлен флажок Redundancy of Communication у модуля слева, как показано на рисунке ниже.

При этом редактирование параметров модуля справа блокируется. Параметры, редактируемые в модуле, вставленном слева, становятся общими для двух модулей.

С другой стороны, снятие флажка Redundancy of Communication в модуле слева приводит к разделению модулей, которые возвращаются к работе как отдельные модули без резервирования NIC Teaming.

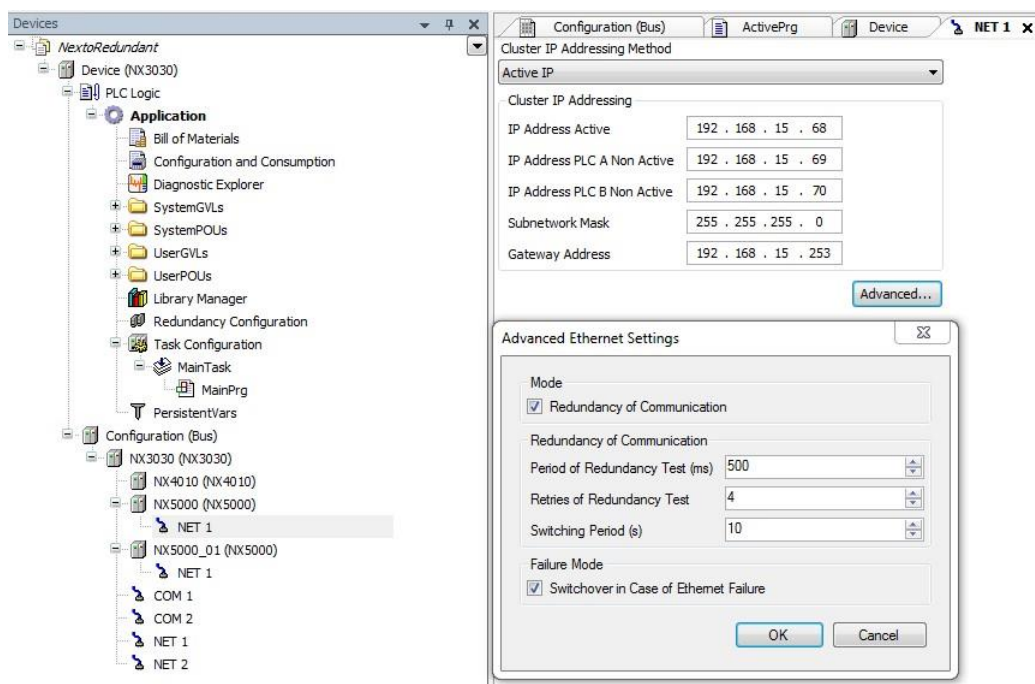


Рисунок 54: Параметр резервирования NX5000

5.5. Конфигурация протоколов

Независимо от протоколов, используемых в каждом конкретном приложении, для процессоров серии Nexto установлены некоторые максимальные ограничения для каждой модели процессора. В основном существует два различных типа протоколов обмена данными: символьное и прямое отображение. Максимальное количество отображений, а также максимальное количество протоколов (экземпляров) определено в следующей таблице:

	NX3030
Отображенные точки	20480
Отображения	5120

5. КОНФИГУРАЦИЯ

Запросы	512
NETs - клиентские или серверные экземпляры (на NET / всего)	4 / 16
COM (n) - ведущие или ведомые экземпляры	1
Центры управления	3

Таблица 65: Лимиты протоколов на процессор

Примечания:

Mapped Points: Обозначает максимальное количество точек отображения, поддерживаемое процессором. Каждое отображение поддерживает одну или несколько точек отображения, в зависимости от размера данных при использовании переменных типа ARRAY.

Отображения: Отображение" - это связь между внутренней переменной приложения и объектом прикладного протокола. Данное поле сообщает максимальное количество отображений, поддерживаемых процессором. Оно соответствует сумме всех отображений, сделанных в экземплярах коммуникационных протоколов и соответствующих им устройств.

Запросы: Сумма запросов протоколов связи, объявленных на устройствах, не может превышать максимальное количество запросов, поддерживаемое процессором.

NETs - Clients or Servers Instances: Это поле определяет максимальное количество экземпляров протоколов на один интерфейс Ethernet, а также общее максимальное количество, распределенное по всем интерфейсам Ethernet системы.

COM (n) - Master или Slave Instances: В силу своих особенностей каждый последовательный интерфейс поддерживает только один экземпляр протокола связи. Примеры экземпляров, совместимых с последовательными интерфейсами: MODBUS RTU Master и MODBUS RTU Slave.

Центры управления: "Центр управления" - это все клиентские устройства, подключенные к центральному процессору по протоколу IEC 60870-5-104. Данное поле информирует о максимальном количестве клиентских устройств типа "Центр управления", поддерживаемых процессором. Соответствует сумме всех клиентских устройств протокола связи IEC 60870-5-104 Server (не включает ведущие или клиенты протоколов MODBUS RTU Slave, MODBUS Server и DNP3 Server).

Ограничения протокола MODBUS для прямого представления и символьного отображения для центральных процессоров можно увидеть в табл. 66 и 67 соответственно.

Ограничения	MODBUS RTU Master	MODBUS RTU Slave	MODBUS Ethernet Client	MODBUS Ethernet Server
Сопоставления для каждого экземпляра	128	32	128	32
Устройства на один экземпляр	64	1 ⁽¹⁾	64	64 ⁽²⁾
Сопоставления на одно устройство	32	32	32	32
Одновременные запросы на экземпляр	-	-	128	64
Одновременные запросы на устройство	-	-	8	64

Таблица 66: Ограничения протокола MODBUS для прямого представления

Примечания:

Устройства на один экземпляр:

- Ведущие или клиентские протоколы: количество ведомых или серверных устройств, поддерживаемых каждым экземпляром ведущего или ведомого протокола.
- Ведомый протокол MODBUS RTU: сообщенное ограничение (1) относится к последовательным интерфейсам, которые не позволяют ведомому устройству устанавливать связь через один и тот же последовательный интерфейс одновременно с более чем одним ведущим устройством. Нет

5. КОНФИГУРАЦИЯ

необходимости или возможности объявлять или конфигурировать ведущее устройство в экземпляре ведомого протокола MODBUS RTU. Ведущее устройство будет иметь доступ ко всем сопоставлениям, сделанным непосредственно на экземпляре ведомого протокола MODBUS RTU.

- Серверный протокол MODBUS RTU: информированное ограничение (2) относится к интерфейсам Ethernet, которое ограничивает количество соединений, которые могут быть установлены с другими устройствами через один интерфейс Ethernet. Нет необходимости и возможности объявлять или конфигурировать клиентские устройства в экземпляре протокола MODBUS Server. Все клиентские устройства будут иметь доступ ко всем отображениям, сделанным непосредственно в экземпляре протокола MODBUS Server.

Сопоставления на одно устройство: Максимальное количество отображений на устройство, несмотря на то, что оно указано выше, также ограничено максимальным количеством отображений протокола. Также следует учитывать максимальное количество отображений на процессор, как показано в табл. 65.

Simultaneous Requests per Instance: Количество запросов, которые могут быть одновременно переданы каждым экземпляром протокола Client или получены каждым экземпляром протокола Server. Экземпляры протокола MODBUS RTU, Master или Slave, не поддерживают одновременные запросы.

Одновременные запросы на устройство: Количество запросов, которые могут одновременно передаваться каждому устройству MODBUS Server или одновременно приниматься каждым клиентским устройством MODBUS. Устройства MODBUS RTU, Master или Slave, не поддерживают одновременные запросы.

Ограничения	MODBUS RTU Master	MODBUS RTU Slave	MODBUS Ethernet Client	MODBUS Ethernet Server
Устройства на один экземпляр	64	1 ⁽¹⁾	64	64 ⁽²⁾
Запросы на одно устройство	32	-	32	-
Одновременные запросы на один экземпляр	-	-	128	64
Одновременные запросы на одно устройство	-	-	8	64

Таблица 67: Ограничения протокола MODBUS для символьных отображений

Примечания:

Устройства на один экземпляр:

- Протокол Master или Client: Количество ведомых или серверных устройств, поддерживаемых каждым экземпляром протокола Master или Client.
- Протокол MODBUS RTU Slave: информированное ограничение (1) относится к последовательным интерфейсам, которые не позволяют ведомому устройству устанавливать связь через один и тот же последовательный интерфейс одновременно с более чем одним ведущим устройством. Нет необходимости или возможности объявлять или конфигурировать ведущее устройство в экземпляре ведомого протокола MODBUS RTU. Ведущее устройство будет иметь доступ ко всем сопоставлениям, сделанным непосредственно на экземпляре ведомого протокола MODBUS RTU.
- Серверный протокол MODBUS RTU: информированное ограничение (2) относится к интерфейсам Ethernet, которые ограничивают количество соединений, которые могут быть установлены с другими устройствами через один интерфейс Ethernet. Нет необходимости и возможности объявлять или конфигурировать клиентские устройства в экземпляре протокола MODBUS Server. Все клиентские устройства будут иметь доступ ко всем связям, созданным непосредственно в экземпляре протокола MODBUS Server.

Запросы по устройствам: Количество запросов, таких как чтение или запись регистров удержания, которые могут быть сконфигурированы для каждого из устройств (ведомых или серверов) из экземпляров протоколов Master или Client. Этот параметр не относится к экземплярам протоколов Slave или Server.

Simultaneous Requests per Instance: Количество запросов, которые могут быть одновременно переданы каждым экземпляром клиентского протокола или получены каждым экземпляром серверного протокола. Экземпляры протокола MODBUS RTU, Master или Slave, не поддерживают одновременные запросы.

5. КОНФИГУРАЦИЯ

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

ВНИМАНИЕ

Одновременные запросы к переменной, связанной с точками связи, поддерживающими режим работы SBO (Выбор перед эксплуатацией), даже поступающие от разных устройств, не поддерживаются. После начала выбора/операции точки определенным устройством, которая должна быть завершена до того, как эта точка станет доступной для командования другим устройством.

Ограничения протокола IEC 60870-5-104 Server можно посмотреть в таблице ниже.

Ограничения	IEC 60870-5-104 Сервер
Устройства на один экземпляр	3
Одновременные запросы на один экземпляр	3
Одновременные запросы на одно устройство	1

Таблица 68: Протокол IEC 60870-5-104 Пределы сервера

Примечания:

Устройства на экземпляр: Количество клиентских устройств типа АСУ ТП, поддерживаемых для каждого экземпляра протокола IEC 60870-5-104 Server. Сообщаемый предел может быть меньше из-за ограничений на суммарную производительность процессора (см. табл. 65).

Одновременные запросы на экземпляр: Количество запросов, которые могут быть получены одновременно каждым экземпляром протокола Server.

Одновременные запросы на устройство: Количество запросов, которые могут быть получены одновременно каждым клиентским устройством IEC 60870-5-104.

5.5.1. Поведение протокола x Состояние ЦПУ

В таблице ниже подробно показано поведение каждого конфигурируемого протокола в процессорах серии Nexto в каждом состоянии работы.

Протокол	Тип	Рабочее состояние ЦПУ					
		ОСТАНОВКА			РАБОТА		
		После загрузки, до начала работы приложения	После перехода приложения в состояние ОСТАНОВКА (ПАУЗА)	После исключения	Не резервируемый или активный	Резервирование в режиме ожидания	После точки останова в MainPrg
MODBUS Symbol	Slave/Server	✓	✓	✓	✓	✓	✓
	Master/Client	✗	✗	✗	✓	✓	✓
MODBUS	Slave/Server	✗	✗	✗	✓	✓	✗
	Master/Client	✗	✗	✗	✓	✓	✓
Протокол	Тип	Рабочее состояние ЦПУ					

5. КОНФИГУРАЦИЯ

		ОСТАНОВКА			РАБОТА		
		После загрузки, до начала работы приложения	После перехода приложения в состояние ОСТАНОВКА (ПАУЗА)	После исключения	Не резервируемый или активный	Резервирование в режиме ожидания	После точки останова в MainPrg
SOE (DNP3)	Розетка	✓	✓	✓	✓	✗	✓
IEC 60870-5-104	Сервер	✗	✗	✗	✓	✗	✓
EtherCAT	Мастер	✗	✗	✗	✓	неприменимо	✗
OPC DA	Сервер	✓	✓	✓	✓	✗	✓
OPC UA	Сервер	✓	✓	✓	✓	✓	✓
SNTP	Клиент	✓	✓	✓	✓	✓	✓
HTTP	Сервер	✓	✓	✓	✓	✓	✓
SNMP	Агент	✓	✓	✓	✓	✓	✓
EtherNet/IP	Сканер	✓	✓	✗	✓	неприменимо	✗
	Адаптер	✗	✓	✗	✓	неприменимо	✗

Таблица 69: Поведение протокола x Состояние процессора

Примечания:

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

Знак ✗: Протокол отключен.

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

MODBUS Symbol Slave/Server: Для поддержания связи по протоколу, когда CPU не находится в режиме RUN или после точки останова, необходимо отметить опцию "Сохранять связь при остановке CPU".

5.5.2. Двойные точки

Представление двойных цифровых точек на входе и выходе осуществляется через специальный тип данных DBP (определен в библиотеке LibDataTypes). Этот тип представляет собой структуру из двух элементов типа BOOL, называемых ВЫКЛ и ВКЛ (эквивалентно TRIP и CLOSE соответственно).

В Nexto переменные этого типа не могут быть связаны с модулями дискретных входов и выходов, поэтому необходимо отображение в одну цифровую точку, тип BOOL, и обработка приложением для преобразования в двойные точки.

Более подробную информацию об отображении двойных точек в цифровых модулях ввода и вывода можно найти в разделе [сервера IEC 60870-5-104](#).

5.5.3. Очередь событий процессора

В процессоре имеется очередь событий типа FIFO (первым вошел, первым вышел), которая используется для временного хранения событий, связанных с точками связи, пока они не перейдут в свою конечную судьбу..

Все события точек связи, генерируемые в процессоре, направляются и сохраняются в очереди процессора. Эта очередь обладает следующими свойствами:

5. КОНФИГУРАЦИЯ

- Размер: 1000 событий
- Ретентивность: не является ретентивной
- Политика переполнения: сохранять самые новые

ВНИМАНИЕ

В ПЛК Nexto очередь событий хранится в неретенсивной области памяти (volatile). Таким образом, события, находящиеся в очереди процессора и еще не переданные в центр управления, будут потеряны при выключении процессора.

Очередь событий процессора является избыточной, то есть она синхронизируется каждый цикл между обоими процессорами, когда используется избыточность процессора. Более подробную информацию можно найти в разделе об избыточности процессора.

Ввод и вывод событий из этой очереди осуществляется в соответствии с концепцией производитель/потребитель. Производителями являются те элементы системы, которые способны генерировать события, добавляя их в очередь процессора, а потребителями - те элементы системы, которые получают и используют эти события, забирая их из очереди процессора. На рисунке ниже показана эта работа, в том числе на примере некоторых потребителей и производителей событий.

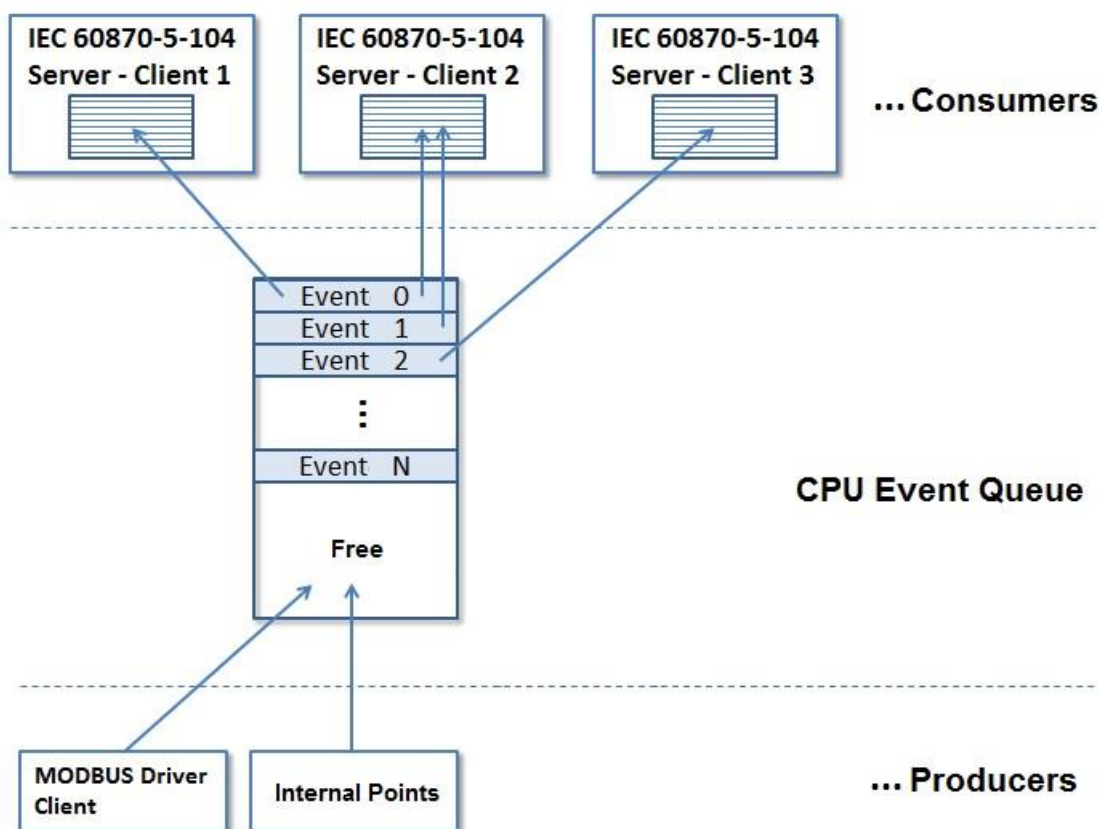


Рисунок 55: Очередь событий процессора

5.5.3.1. Пользователи

В качестве потребителей обычно выступают коммуникационные драйверы, осуществляющие связь со SCADA или HMI. После сохранения в очереди процессора потребители получают события, связанные с точками связи, отображенными в его конфигурации. Эти события затем сохраняются в собственной очереди событий потребителя, размер и работа которой описаны в разделе, посвященном драйверам связи..

5. КОНФИГУРАЦИЯ

5.5.3.2. Принципы функционирования очередей

После сохранения в очереди процессора каждое событие передается потребителю, в базе данных которого есть эта точка связи.

На приведенном рисунке событие 0 относится к точке связи, сопоставленной с двумя центрами управления МЭК 60870-5-104 (клиент 1 и 2). При этом событие 1 относится к точке связи, сопоставленной только с одним центром управления МЭК 60870-5-104 (Клиент 2). Событие 2 по времени относится к точке связи с другим центром управления МЭК 60870-5-104 (Клиент 3).

События остаются в очереди ЦПУ до тех пор, пока все его потребители не подтвердят его получение. Критерии, используемые для подтверждения приема, индивидуальны для каждого потребителя. В случае сервера МЭК 60870-5-104 подтверждение происходит при передаче события клиенту МЭК 60870-5-104.

В случае с Nexto Series для наблюдения за занятостью очереди событий процессора не предусмотрена диагностика, нет даже информации о переполнении очереди. Однако у потребителей есть группа диагностики, относящаяся к его очереди событий. Более подробную информацию можно найти в разделе, посвященном взаимодействию с конкретным драйвером.

5.5.3.2.1. Знак переполнения

Признак переполнения очереди событий потребителей возникает в двух ситуациях:

- Если в очереди событий потребителя закончилось место для хранения новых событий
- Если процессор прервал генерацию событий (поскольку за один цикл выполнения произошло больше событий, чем общий размер очереди событий)

5.5.3.3. Производители

В качестве производителей обычно выступают коммуникационные драйверы или внутренние элементы ПЛК, способные генерировать события. На предыдущем рисунке показаны некоторые примеры.

- Внутренние точки: Это внутренний элемент микропрограммы ПЛК, который каждый цикл выполнения (MainTask) обнаруживает события в тех точках связи, которые не имеют определенного происхождения, и затем помещает эти события в очередь процессора. Максимальное количество событий, которое может быть обнаружено в каждом цикле MainTask, равно размеру очереди событий процессора. Если количество генерируемых событий больше этого значения, то за один цикл превышение будет потеряно.
- Драйвер MODBUS (клиент/сервер/ведущий/ведомый): Изменение значений переменных, вызванное чтением/записью MODBUS, фиксируется в каждом цикле MainTask, после чего события помещаются в очередь процессора. В случае Client/Master генерируются также события качества при нарушении связи с ведомым устройством.

5.5.4. Перехват команд, поступающих из центра управления

В ПЛК Nexto имеется функциональный блок, позволяющий обрабатывать пользовательской логикой команды выбора и работы с точками выхода, поступающие от драйверов серверов (IEC 60870-5-104 Server). Этот ресурс позволяет реализовать блокировку, а также обрабатывать полученные командные данные в логике пользователя или перенаправлять команды на различные IED.

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

Параметр	Тип	Описание
bExec	BOOL	Если TRUE, то выполняется перехват команды
bDone	BOOL	Указывает на то, что выходные данные команды уже обработаны, освобождая функциональный блок для приема другой команды
dwVariableAddr	DWORD	Переменный адрес, отображаемый в драйвере сервера, на который будет поступать команда клиента

eCommandResult	ENUM	<p>Входное действие, определяемое пользователем из следующего списка:</p> <p>SUCCESS(0) NOT_SUPPORTED(1) BLOCKED_BY_SWITCHING_HIERARCHY(2) SELECT_FAILED(3) INVALID_POSITION(4) POSITION_REACHED(5) PARAMETER_CHANGE_IN_EXECUTION(6) STEP_LIMIT(7) BLOCKED_BY_MODE(8) BLOCKED_BY_PROCESS(9) BLOCKED_BY_INTERLOCKING(10) BLOCKED_BY_SYNCHROCHECK(11) COMMAND_ALREADY_IN_EXECUTION(12) BLOCKED_BY_HEALTH (13) ONE_OF_N_CONTROL(14) ABORTION_BY_CANCEL(15) TIME_LIMIT_OVER(16) ABORTION_BY_TRIP(17) OBJECT_NOT_SELECTED(18) OBJECT_ALREADY_SELECTED(19) NO_ACCESS_AUTHORITY(20) ENDED_WITH_OVERSHOOT(21) ABORTION_DUE_TO_DEVIATION(22) ABORTION_BY_COMMUNICATION_LOSS(23) BLOCKED_BY_COMAND(24) NONE(25) INCONSISTENT_PARAMETERS(26) LOCKED_BY_OTHER_CLIENT(27) HARDWARE_ERROR(28) UNKNOWN(29)</p>
dwTimeout	DWORD	Тайм-аут [мс] на обработку пользовательской логикой

Таблица 70: Входные параметры функционального блока "Приемник команд"

Примечания:

bExес: bExес: Если FALSE, то команда просто перестает перехватываться для пользовательского приложения, но продолжает нормально обрабатываться сервером.

bDоne: После перехвата команды пользователь будет отвечать за ее обработку. По окончании обработки этот вход должен быть включен, чтобы можно было получить новую команду. Если этот вход не включен, то блок будет ждать время, определенное в dwTimeout, после чего станет способен перехватывать новые команды.

eCommandResult: Результат обработки перехваченной пользователем команды. Результат, возвращаемый клиенту, пославшему команду, который должен быть отнесен вместе с входом bDоne, преобразуется в формат протокола, от которого была получена команда. В Nexto Series поддерживается только перехват команд, поступающих по протоколу IEC 608705-104. При перехвате протокола любой возврат, отличный от SUCCESS, приводит к отрицательному подтверждению (Acknowledge).

ВНИМАНИЕ

Не рекомендуется одновременный перехват команд к одной и той же переменной двумя и более функциональными блоками CommandReceiver. Только один из функциональных блоков корректно перехватит команду, при этом он может подвергнуться нежелательной интерференции со стороны других функциональных блоков при обращении к одной и той же переменной.

Параметр	Тип	Описание
bCommandAvailable	BOOL	Указывает на то, что команда была перехвачена и данные доступны для обработки
sCommand	STRUCT	В этой структуре хранятся данные о принятой команде, которые состоят из следующих полей: eCommand sSelectParameters sOperateParameters Описание каждого поля приведено в данном разделе.
eStatus	ENUM (TYPE_RESULT)	Действие вне функции из полученного результата, в соответствии со списком: OK_SUCCESS(0) ERROR_FAILED(1)

Таблица 71: Выходные параметры функционального блока CommandReceiver

Примечания:

eStatus: Возврат процесса регистрации перехвата команды точки связи. При успешной регистрации перехвата возвращается OK_SUCCESS, в противном случае - ERROR_FAILED. В случае сбоя регистра перехватчика команды на определенную точку не перехватываются данным функциональным блоком. TYPE_RESULT определен в библиотеке LibDataTypes.

Поддерживаемые команды описаны в таблице ниже:

Параметр	Тип	Описание
eCommand	ENUM	NO_COMMAND(0) ВЫБОР(1) ВЫПОЛНИТЬ(2)

Таблица 72: Поддерживаемые команды функционального блока CommandReceiver

Параметры, на основе которых строятся структуры sSelectParameters, sOperateParameters и sCancelParameters, описаны в следующей таблице:

Параметр	Тип	Описание
sSelectConfig	STRUCT	Конфигурация команды выбора приема. Параметры этой структуры описаны в таблице 74
sValue	STRUCT	Received value in a select, когда принимается команда выбора со значением. Параметры данной структуры описаны в таблице 77

Таблица 73: Параметры sSelectParameters

Параметр	Тип	Описание
----------	-----	----------

5. КОНФИГУРАЦИЯ

bSelectWithValue	BOOL	При значении true указывает на прием команды выбора со значением.
------------------	------	---

Таблица 74: Параметры sSelectConfig

Параметр	Тип	Описание
sOperateConfig	STRUCT	Конфигурация команды выбора приема. Параметры этой структуры описаны в таблице 76
sValue	STRUCT	Поле относимого значения принятой команды операции. Параметры данной структуры описаны в таблице 77

Таблица 75: Параметры sOperateParameters

Параметр	Тип	Описание
bDirectOperate	BOOL	Если true, то это означает, что была получена команда операции без выбора.
bNoAcknowledgement	BOOL	Когда true указывает, что была получена команда, не требующая подтверждения приема.
bTimedOperate	BOOL	Когда true указывает на то, что была получена команда операции, активируемая временем.
liOperateTime	LINT	Программирование момента, в который должна быть выполнена команда. Это поле действительно только в том случае, если bTimedOperate имеет значение true.
bTest	BOOL	Если true, то это означает, что полученная команда была отправлена только для проверки, поэтому выполнять ее не следует.

Таблица 76: Параметры sOperateConfig

Параметр	Тип	Описание
eParamType	ENUM	Информирует о типе полученной команды: NO_COMMAND(0) SINGLE_POINT_COMMAND(1) DOUBLE_POINT_COMMAND(2) INTEGER_STATUS_COMMAND(3) ENUMERATED_STATUS_COMMAND(4) ANALOGUE_VALUE_COMMAND(5)
sSinglePoint	STRUCT	При получении команды в функции типа полученной команды, определяемой eParamType, заполняется соответствующая структура данных. Параметры этой структуры описаны в таблицах 78 - 82
sDoublePoint	STRUCT	
sIntegerStatus	STRUCT	
sEnumeratedStatus	STRUCT	
sAnalogueValue	STRUCT	

Таблица 77: Параметры sValue

Параметр	Тип	Описание
----------	-----	----------

5. КОНФИГУРАЦИЯ

bValue	BOOL	Значение работы точки.
sPulseConfig	STRUCT	В этой структуре хранятся параметры конфигурации импульсных команд. Параметры этой структуры описаны в табл. 83.

Таблица 78: Параметры sSinglePoint

Параметр	Тип	Описание
bValue	BOOL	Значение работы точки.
sPulseConfig	STRUCT	В этой структуре хранятся параметры конфигурации импульсных команд. Параметры этой структуры описаны в табл. 83.

Таблица 79: Параметры sDoublePoint

Параметр	Тип	Описание
diValue	DINT	Значение работы точки.

Таблица 80: Параметры sIntegerStatus

Параметр	Тип	Описание
dwValue	DWORD	Значение работы точки.

Таблица 81: Параметры sEnumeratedStatus

Параметр	Тип	Описание
eType	ENUM	Информирует о типе данных полученного аналогового значения. INTEGER (0) FLOAT (1)
diValue	DINT	Значение точечной операции, целочисленный формат.
fValue	REAL	Значение точечной операции, формат float.

Таблица 82: Параметры sAnalyseValue

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

Таблица 83: Параметры sPulseConfig

Для перехвата команд в конкретной точке сначала необходимо загрузить в параметр dwVariableAddr адрес переменной, соответствующей точке, в которой необходимо перехватить команду, а затем выполнить импульс в

5. КОНФИГУРАЦИЯ

параметре bExec. После перехвата команды функциональный блок сообщает о том, что команда была перехвачена, через параметр bCommandAvailable. Затем информация о перехваченной команде заполняет выходные параметры sCommand и eStatus в соответствии с типом полученной команды. Эта операция зависит только от типа полученной команды, тип данных переменной, в которую перехватывается команда, не имеет значения. Перехват завершается, после чего функциональный блок может быть освобожден для перехвата новой команды, если параметр bDone равен true. При этом результат обработки команды должен быть указан в eCommandResult.

Пример применения перехватчика команд на языке ST, который пересылает команду, полученную сервером IEC 60870-5-104, в двойную точку цифрового выхода серии Nexto, приведен в разделе [Двойные точки цифрового выхода](#).

5.5.5. MODBUS RTU Master

Этот протокол доступен для процессоров серии Nexto в их последовательных каналах. При выборе этой опции в MasterTool IEC XE процессор становится ведущим устройством связи MODBUS, что позволяет ему в режиме исполнения (Run Mode) обращаться к другим устройствам с тем же протоколом.

Для этого протокола существует два режима конфигурирования. Один из них использует прямое представление (%Q), при котором переменные определяются по их адресу. В другом, называемом 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 MODBUS RTU Master в последовательный канал COM 1 или COM 2 (или оба, если речь идет о двух сетях связи). Для выполнения этой процедуры см. раздел "[Вставка экземпляра протокола](#)".
- Настройте последовательный интерфейс, выбрав скорость передачи, поведение сигналов RTS/CTS, четность, канал Стоп-биты и другие конфигурации двойным щелчком мыши на последовательном канале COM 1 или COM 2. См. раздел [Конфигурация последовательных интерфейсов](#).

5.5.5.1. Конфигурирование протокола ведущего устройства MODBUS с помощью символьного отображения

- Для настройки этого протокола с использованием символьного отображения необходимо выполнить следующие действия:
- Настройте общие параметры протокола MODBUS Master, такие как: время задержки передачи и минимальный интерфрейм, как показано на рис. 56.
- Добавление и настройка устройств на закладке General Parameters, определение адреса ведомого устройства, тайм-аута связи и количества повторных попыток связи, как показано на рис. 57.
- Добавление и конфигурирование отображений MODBUS на вкладке Mappings (рис. 58), указав имя переменной, тип данных и начальный адрес данных, размер и диапазон данных заполняются автоматически.
- Добавление и настройка запросов MODBUS, как показано на рис. 59, с указанием функции, времени сканирования запроса, начального адреса (чтение/запись), размера данных (чтение/запись) и генерации диагностических переменных, а также отключение запроса с помощью кнопок в нижней части окна.

5.5.5.1.1. Общие параметры протокола MODBUS Master - Конфигурация символьного отображения

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

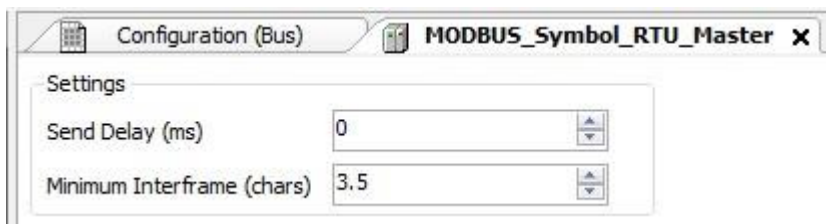


Рисунок 56: Экран конфигурации ведущего устройства MODBUS RTU

Конфигурация	Описание	По умолчанию	Опции
Задержка передачи (мс)	Задержка передачи ответа.	0	0 до 65535
Минимальный интерфрейм (символов)	Минимальное время молчания между различными кадрами.	3.5	3.5 до 100.0

Таблица 84: 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 ответа ведомого и молчанием физической линии (задержка ведомого на установку RTS в ноль и перевод RS-485 в высокоимпедансное состояние). Для решения этой проблемы ведущий может выждать определенное в этом поле время перед отправкой нового запроса. В противном случае первые байты, переданные ведущим, могут быть потеряны.

Минимальный интерфрейм: Стандарт MODBUS определяет это время как 3,5 символа, однако этот параметр является настраиваемым для того, чтобы обслуживать устройства, которые не следуют стандарту.

Диагностика и команды протокола MODBUS, сконфигурированные либо путем символьного отображения, либо путем прямого представления, хранятся в переменных T_DIAG_MODBUS_RTU_MASTER_1. Для прямого отображения они также находятся в 4 байтах и 8 словах, которые описаны в таблице ниже (где "n" - сконфигурированное значение в поле %Q Начальный адрес области диагностики):

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_MASTER_1.*	Размер	Описание
Биты диагностики:			
%Q(n).0	tDiag. bRunning	БИТ	Ведущее устройство работает.
%Q(n).1	tDiag. bNotRunning	БИТ	Мастер не работает (см. бит: bInterruptedByCommand).
%Q(n).2	tDiag. bInterruptedByCommand	БИТ	Бит bNotRunning был включен, так как работа мастера была прервана пользователем с помощью командных битов.
%Q(n).3	tDiag. bConfigFailure	БИТ	Прерванная диагностика.

5. КОНФИГУРАЦИЯ

%QX(n).4	tDiag. bRXFailure	БИТ	Прерванная диагностика.
%QX(n).5	tDiag. bTXFailure	БИТ	Прерванная диагностика.
%QX(n).6	tDiag. bModuleFailure	БИТ	Указывает на наличие сбоя в модуле или на отсутствие модуля.
%QX(n).7	tDiag. bDiag_7_reserved	БИТ	Зарезервировано
Коды ошибок:			

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS _RTU_MASTER_1.*	Размер	Описание
%QB(n+1)	eErrorCode	SERIAL_STATUS (BYTE)	<p>0: ошибки отсутствуют.</p> <p>1: недопустимый последовательный порт.</p> <p>2: недопустимый режим работы последовательного порта</p> <p>3: недопустимая скорость передачи данных</p> <p>4: недопустимые Биты данных 5: недопустимая четность</p> <p>6: invalid Стоп-биты</p> <p>7: недопустимый параметр сигнала модема</p> <p>8: недопустимый параметр UART RX Threshold</p> <p>9: недопустимый параметр тайм-аута</p> <p>10: занят последовательный порт</p> <p>11: аппаратная ошибка UART</p> <p>12: аппаратная ошибка удаленного устройства</p> <p>20: недопустимый размер буфера передачи</p> <p>21: недопустимый метод сигнального модема</p> <p>22: тайм-аут CTS = true</p> <p>23: тайм-аут CTS = false</p> <p>24: ошибка тайм-аута передачи</p> <p>30: недопустимый размер буфера приема</p> <p>31: ошибка тайм-аута приема</p>

5. КОНФИГУРАЦИЯ

			<p>32: управление потоком сконфигурировано иначе, чем вручную</p> <p>33: недопустимое управление потоком для сконфигурированного последовательного порта</p> <p>34: прием данных в нормальном режиме запрещен</p> <p>35: прием данных в расширенном режиме запрещен</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>
Командные биты, инициализируемые автоматически:			
%QX(n+2).0	tCommand. bStop	БИТ	Остановить мастер.
%QX(n+2).1	tCommand. bRestart	БИТ	Перезапустить мастер.
%QX(n+2).2	tCommand. bResetCounter	БИТ	Перезапуск статистики (счетчиков) диагностики.
%QX(n+2).3	tCommand. bDiag_19_reserved	БИТ	Зарезервировано
%QX(n+2).4	tCommand. bDiag_20_reserved	БИТ	Зарезервировано
%QX(n+2).5	tCommand. bDiag_21_reserved	БИТ	Зарезервировано
%QX(n+2).6	tCommand. bDiag_22_reserved	БИТ	Зарезервировано
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_MASTER_1.*	Размер	Описание
%QX(n+2).7	tCommand. bDiag_23_reserved	БИТ БИТ	Зарезервировано
%QB(n+3)	byDiag_3_reserved	БАЙТ	Зарезервировано
Статистика коммуникаций:			
%QW(n+4)	tStat. wTXRequests	СЛОВО	Счетчик запросов, переданных ведущим устройством (от 0 до 65535).
%QW(n+6)	tStat. wRXNormalResponses	СЛОВО	Счетчик нормальных ответов, полученных ведущим (от 0 до 65535).

5. КОНФИГУРАЦИЯ

%QW(n+8)	tStat. wRXExceptionResponses	СЛОВО	Счетчик ответов с кодами исключений, полученных ведущим (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalResponses	СЛОВО	Счетчик неправомерных ответов, полученных ведущим - неверный синтаксис, недостаточное количество принятых байт, неверный CRC - (от 0 до 65535).
%QW(n+12)	tStat. wRXOverrunErrors	СЛОВО	Счетчик ошибок превышения при приеме - UART FIFO или линия RX - (от 0 до 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	СЛОВО	Счетчик ответов с ошибками построения, четности или сбоя во время приема (от 0 до 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	СЛОВО	Счетчик ошибок тайм-аута CTS при использовании рукопожатия RTS/CTS при передаче (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	СЛОВО	Зарезервировано

Таблица 85: Диагностика ведущего устройства MODBUS RTU

Примечание:

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

5.5.5.1.2. Конфигурация устройств - конфигурация символьного отображения

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

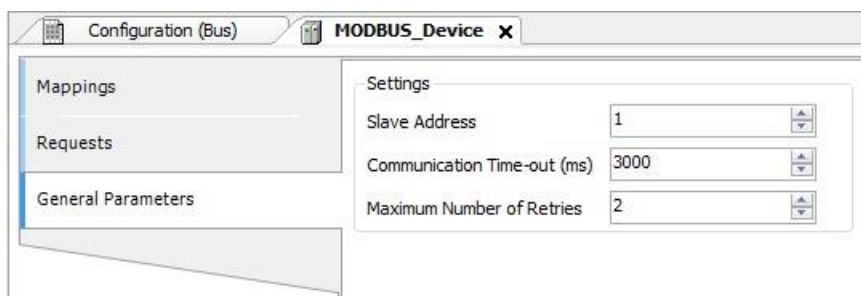


Рисунок 57: Настройки общих параметров устройства

Конфигурация	Описание	По умолчанию	Опции
Адрес ведомого устройства	адрес ведомого устройства MODBUS	1	0 до 255
Конфигурация	Описание	По умолчанию	Опции
Тайм-аут связи (мс)	Определяет тайм-аут на уровне приложения	3000	10 до 65535

5. КОНФИГУРАЦИЯ

Максимальное количество повторных попыток	Определяет количество повторных попыток перед сообщением об ошибке связи	2	0 до 9
---	--	---	--------

Таблица 86: Конфигурации устройств

Примечания:

Адрес ведомого устройства: Согласно стандарту MODBUS, допустимыми адресами ведомых устройств являются адреса от 0 до 247, при этом адреса от 248 до 255 являются зарезервированными. Когда ведущий посылает команду записи с адресом, установленным в ноль, он осуществляет широковещательный запрос в сети.

Тайм-аут связи: Тайм-аут связи - это время, в течение которого ведущий ожидает ответа от ведомого на запрос. Для ведущего устройства MODBUS RTU необходимо учитывать как минимум следующие системные переменные: время передачи кадра ведомым (в зависимости от скорости передачи), время обработки запроса ведомым и задержка отправки ответа, если она задана в ведомом устройстве. Рекомендуется, чтобы тайм-аут был равен или больше времени передачи кадра плюс задержка отправки ответа и удвоенное время обработки запроса. Более подробная информация приведена в разделе "[Производительность связи](#)".

Maximum number of retries (Максимальное количество повторов): Устанавливает количество повторных попыток перед сообщением об ошибке связи. Например, если ведомое устройство не отвечает на запрос, а ведущее устройство настроено на отправку трех повторных попыток, то при выполнении этих трех повторных попыток номер счетчика ошибок увеличивается на единицу. После увеличения ошибки связи при попытке перезапуска процесса и при повторном достижении числа повторных попыток новая ошибка инкрементирует счетчик.

5.5.5.1.3. Конфигурация отображений - Настройки символьных отображений

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

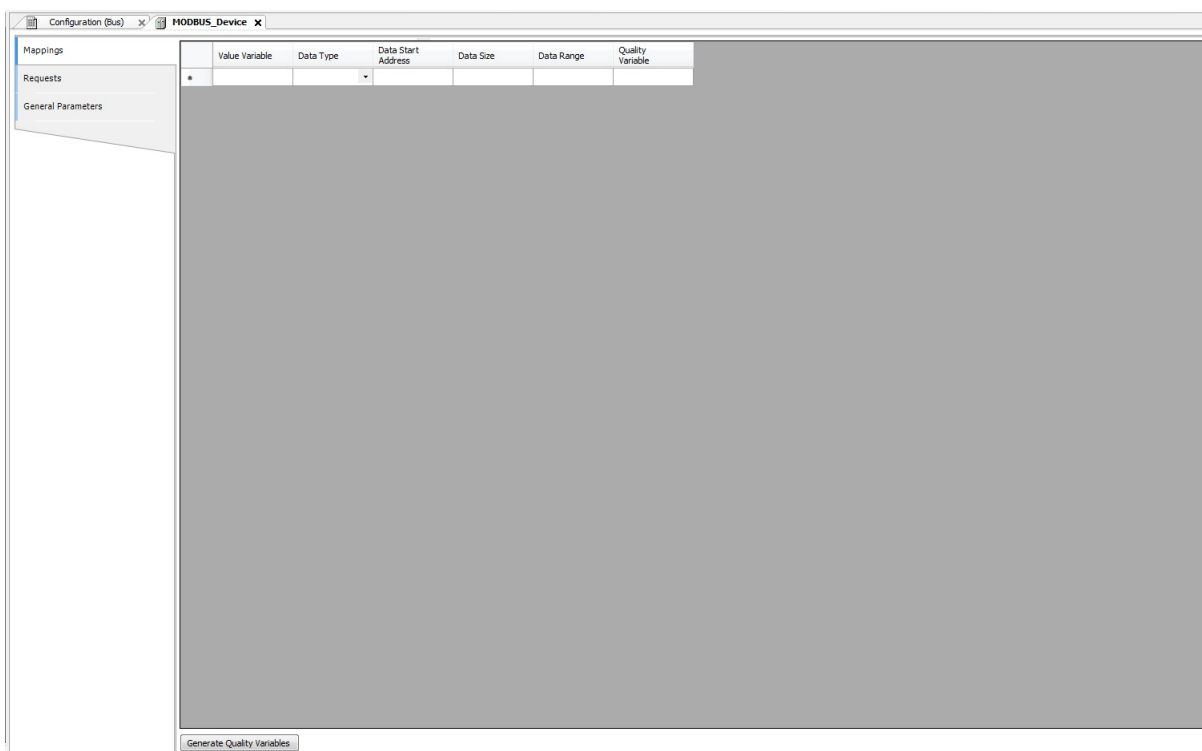


Рисунок 58: Экран сопоставления данных MODBUS

5. КОНФИГУРАЦИЯ

Конфигурация	Описание	По умолчанию	Опции
Значение Переменная	Символьное имя переменной	-	Имя переменной, объявленной в программе или GVL
Тип данных	Тип данных MODBUS	-	Катушка - запись (1 бит) Катушка - чтение (1 бит) Регистр удержания - запись (16 бит) Регистр удержания - чтение (16 бит) Регистр удержания - Маска И (16 бит) Регистр удержания - Маска Ор (16 бит) Входной регистр (16 бит) Состояние входа (1 бит)
Адрес начала данных	Начальный адрес данных MODBUS	-	1 до 65536
Размер данных	Размер данных MODBUS	-	1 до 65536
Диапазон данных	Диапазон адресов конфигурируемых данных	-	-

Таблица 87: Настройки отображения MODBUS

Примечания:

Значение переменной: это поле используется для указания символьной переменной в связи MODBUS.

Тип данных: это поле используется для указания типа данных, используемого в связи MODBUS.

Тип данных	Размер [бит]	Описание
Катушка - запись	1	Запись цифрового вывода.
Катушка - чтение	1	Чтение цифрового выхода.
Регистр удержания - запись	16	Запись аналогового выхода.
Регистр удержания - чтение	16	Чтение аналогового выхода.
Регистр удержания - Маска и	16	Аналоговый выход, который может быть считан или записан по маске.
Регистр удержания - маска или	16	Аналоговый выход, который может быть считан или записан по маске ИЛИ.
Входной регистр	16	Аналоговый вход, который может быть только считан.
Состояние входа	1	Цифровой вход, который может быть только считан.

Таблица 88: Типы данных, поддерживаемые в MODBUS

5. КОНФИГУРАЦИЯ

Начальный адрес данных: Начальный адрес данных в отображении MODBUS.

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

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

5.5.5.1.4. Конфигурация запросов - Настройки символьного отображения

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

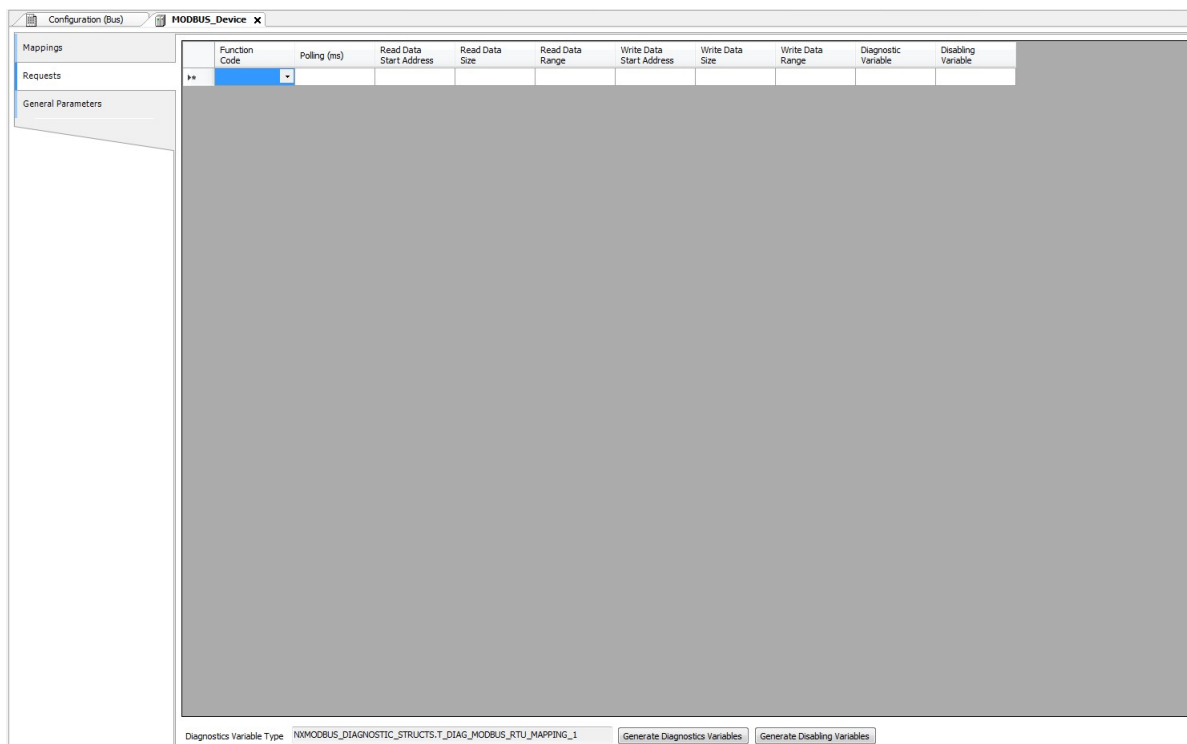


Рисунок 59: Экран запросов данных Ведущий MODBUS

Конфигурация	Описание	По умолчанию	Опции
Код функции	Тип функции MODBUS	-	01 - Считывание катушек 02 - Чтение состояния входов 03 - Чтение регистров удержания 04 - Чтение входных регистров 05 - Запись одной катушки 06 - Запись одного регистра 15 - Запись нескольких катушек

5. КОНФИГУРАЦИЯ

			16 - Запись нескольких регистров 22 - Маска записи регистра 23 - чтение/запись нескольких регистров
Опрос (мс)	Период связи (мс)	100	0 до 3600000
Адрес начала считывания данных	Начальный адрес данных чтения MODBUS	-	1 до 65536
Размер считываемых данных	Размер данных для чтения MODBUS	-	Зависит от используемой функции
Диапазон считываемых данных	Диапазон адресов данных чтения MODBUS	-	0 до 2147483646
Запись данных Начальный адрес	Начальный адрес данных записи MODBUS	-	1 до 65536
Размер записываемых данных	Размер данных записи MODBUS	-	Зависит от используемой функции
Диапазон записываемых данных	Диапазон адресов данных записи MODBUS	-	0 to 2147483647
Конфигурация	Описание	По умолчанию	Опции
Диагностическая переменная	Имя диагностической переменной	-	Имя переменной, объявленной в программе или GVL
Отключающая переменная	Переменная, используемая для отключения связи с MODBUS	-	Поле для символьной переменной, используемой для индивидуального отключения сконфигурированных запросов MODBUS. Эта переменная должна иметь тип BOOL. Переменная может быть простым элементом или элементом массива, а также находиться в структурах.

Таблица 89: Конфигурация связей MODBUS

Примечания:

Установка: количество заводских установок по умолчанию и значения для столбца Опции могут отличаться в зависимости от типа данных и функции MODBUS (FC).

Код функции: Доступны следующие функции MODBUS (FC):

Код		Описание
DEC	HEX	
1	0x01	Считывание катушек (FC 01)

5. КОНФИГУРАЦИЯ

2	0x02	Чтение состояния входов (FC 02)
3	0x03	Чтение регистров удержания (FC 03)
4	0x04	Чтение входных регистров (FC 04)
5	0x05	Запись одной катушки (FC 05)
6	0x06	Запись одного регистра удержания (FC 06)
15	0x0F	Запись нескольких катушек (FC 15)
16	0x10	Запись нескольких регистров удержания (FC 16)
22	0x16	Маска записи регистра удержания (FC 22)
23	0x17	Чтение/запись нескольких регистров удержания (FC 23)

Таблица 90: Функции MODBUS, поддерживаемые процессорами Nexto

Запрос: этот параметр указывает, как часто должна выполняться связь, установленная для данного запроса. По окончании обмена данными будет ожидаться время, равное значению, заданному в поле polling, после чего будет выполнен новый обмен данными.

Начальный адрес чтения данных: поле для указания начального адреса чтения данных MODBUS.

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

- Считывание катушек (FC 01): 2000
- Чтение состояния входа (FC 02): 2000
- Чтение регистров удержания (FC 03): 125
- Чтение входных регистров (FC 04): 125
- Чтение/запись множественных регистров (FC 23): 121

Диапазон данных чтения: в этом поле отображается диапазон данных чтения MODBUS, настроенный для каждого запроса. Начальный адрес вместе с размером считываемых данных определяют диапазон считываемых данных для каждого запроса.

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

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

- Запись одиночной катушки (FC 05): 1
- Запись одного регистра (FC 06): 1
- Запись нескольких катушек (FC 15): 1968
- Запись нескольких регистров (FC 16): 123
- Регистр записи маски (FC 22): 1
- Чтение/запись нескольких регистров (FC 23): 121

Диапазон данных записи: в этом поле отображается диапазон данных записи MODBUS, настроенный для каждого запроса. Начальный адрес вместе с размером считываемых данных определяют диапазон данных записи для каждого запроса.

Diagnostic Variable (Диагностическая переменная): Диагностика запросов MODBUS, сконфигурированная по символному отображению или по прямому представлению, хранится в переменных типа T_DIAG_MODBUS_RTU_MAPPING_1 для устройств Master и T_DIAG_MODBUS_ETH_CLIENT_1 для устройств Client, а отображение по прямому представлению - в 4-байтовых и 2-словных, которые описаны в таблице 91 (n - значение, заданное в поле %Q Начальный адрес области диагностики).

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS	Размер	Описание
----------------------------------	--	--------	----------

5. КОНФИГУРАЦИЯ

_RTU_MAPPING_1.*			
Биты состояния связи:			
%QX(n).0	byStatus. bCommIdle	БИТ	Коммуникация простаивает (ожидает выполнения).
%QX(n).1	byStatus. bCommExecuting	БИТ	Активная связь.
%QX(n).2	byStatus. bCommPostponed	БИТ	Отложенное сообщение, поскольку было 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 тайм-аута. Биты bCommIdle и bCommExecuting ложны, если бит bCommPostponed истинен.
%QX(n).3	byStatus. bCommВыключено	БИТ	Связь отключена. В этом состоянии бит bCommIdle перезапускается.
%QX(n).4	byStatus. bCommOk	БИТ	Связь, прерванная ранее, была проведена успешно.
%QX(n).5	byStatus. bCommError	БИТ	В ранее завершенной связи произошла ошибка. Проверьте код ошибки.
%QX(n).6	byStatus. bCommAborted	БИТ	Не используется в MODBUS RTU Master.
%QX(n).7	byStatus. bDiag_7_reserved	БИТ	Зарезервировано
Последний код ошибки (включается при bCommError = true):			
%QB(n+1)	eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Информирует о возможной причине последней ошибки в отображении MODBUS. Подробнее см. табл. 114 для получения более подробной информации.
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_ВЫКЛЮЧЕНО (129)* IGNORE_FRAME (255)*
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_MAPPING_1.*	Размер	Описание
Статистика коммуникаций:			

5. КОНФИГУРАЦИЯ

%QB(n+3)	byDiag_3_reserved	БАЙТ	Reserved.
%QW(n+4)	wCommCounter	СЛОВО	Счетчик завершенных коммуникаций (с ошибками или без). По изменению этого счетчика пользователь может проверить, когда закончилась связь. При достижении значения 65535 счетчик возвращается к нулю.
%QW(n+6)	wCommErrorCounter	СЛОВО	Счетчик завершенных коммуникаций (с ошибками). При достижении значения 65535 счетчик возвращается к нулю.

Таблица 91: Диагностика связей MODBUS

Примечания:

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

Переменная отключения: переменная булевского 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.

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

Код	Перечислимые	Описание
1	ERR_EXCEPTION	Ответ находится в коде исключения (см. eLastExceptionCode = Код исключения).
2	ERR_CRC	Ответ с недопустимым CRC.
3	ERR_ADDRESS	Адрес MODBUS не найден. Адрес, по которому был получен ответ 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 отличается от ожидаемого.
7	ERR_NOT_ECHO	Ответ не является эхом запроса (FC 05 и 06).
8	ERR_REFERENCE_NUMBER	Неверный номер ссылки (FC 15 и 16).
9	ERR_INVALID_FRAME_SIZE	Ответ короче, чем ожидалось.
20	ERR_CONNECTION	Ошибка при установлении соединения.
21	ERR_SEND	Ошибка на этапе передачи.
22	ERR_RECEIVE	Ошибка на этапе приема.
40	ERR_CONNECTION_TIMEOUT	Тайм-аут на уровне приложения во время соединения.
41	ERR_SEND_TIMEOUT	Тайм-аут на уровне приложения во время передачи.
42	ERR_RECEIVE_TIMEOUT	Тайм-аут на уровне приложения при ожидании ответа.

5. КОНФИГУРАЦИЯ

43	ERR_CTS_OFF_TIMEOUT	Тайм-аут при ожидании CTS = false при передаче.
44	ERR_CTS_ON_TIMEOUT	Тайм-аут при ожидании CTS = true при передаче.
128	NO_ERROR	Ошибок с момента запуска нет.

Table 92: MODBUS Relations Error Codes

ВНИМАНИЕ

В отличие от других прикладных задач, при достижении метки разрядки в MainTask задача экземпляра Master MODBUS RTU и любая другая задача MODBUS прекращает работу в тот момент, когда она пытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока основная задача не выполняется.

5.5.5.2. Конфигурация протокола ведущего устройства MODBUS для прямого представления (%Q)

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

- Настройка общих параметров протокола MODBUS, таких как: время обмена данными и переменные прямого представления (%Q) для получения диагностики.
- Добавление и конфигурирование устройств путем задания адреса, переменных прямого представления (%Q) для отключения связей, тайм-аутов связи и т.д.
- Добавление и конфигурирование связей MODBUS с указанием типа данных и функции MODBUS, тайм-аутов, переменных прямого представления (%Q) для получения диагностики связи и других для приема/записи данных, количества передаваемых данных и опроса связи.

Описание каждой конфигурации приведено ниже в этом разделе.

5.5.5.2.1. Общие параметры протокола MODBUS Master - настройка через прямое представление (%Q)

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

5. КОНФИГУРАЦИЯ

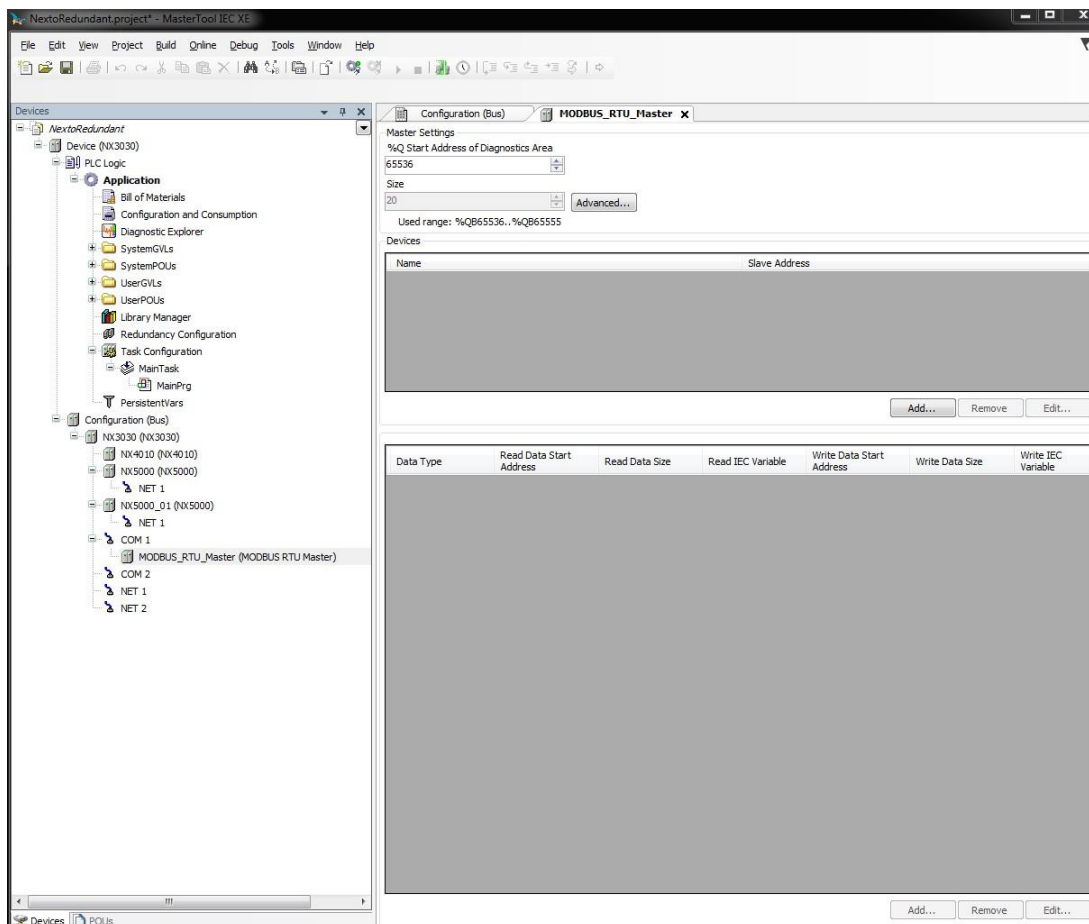


Рисунок 60: Экран настройки мастера MODBUS RTU

Переменные прямого представления (%Q) для протокольной диагностики:

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

Таблица 93: Конфигурация ведущего устройства MODBUS RTU

Примечания:

Начальный адрес области диагностики в %Q: данное поле ограничено размером переменных выходов (%Q) адресуемой памяти каждого процессора, который можно найти в разделе [Память](#).

Значение по умолчанию: заводское значение по умолчанию для поля %Q Начальный адрес области диагностики не может быть установлено, так как 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 табл. [85](#).

Время обмена данными протокола MODBUS Master, определяемое по кнопке Advanced... на экране конфигурации, делится на задержку отправки и минимальный интерфрейм, подробнее описано в разделе [Общие параметры протокола MODBUS Master - Конфигурация символьного отображения](#).

5. КОНФИГУРАЦИЯ

5.5.5.2.2. Конфигурация устройств - Конфигурация для прямого представления (%Q)

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

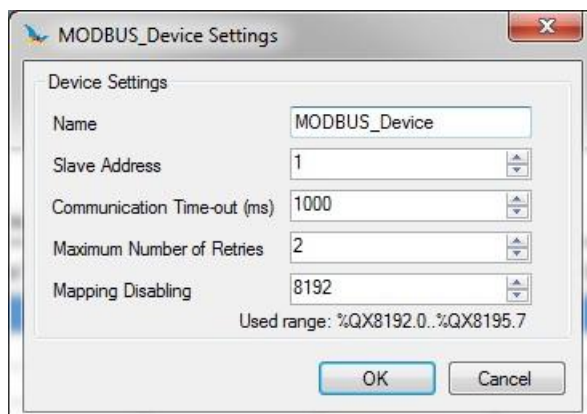


Рисунок 61: Конфигурация устройства

Конфигурация	Описание	Значение по умолчанию	Опции
Имя	Имя экземпляра	MODBUS_Device	Идентификатор, в соответствии с IEC 61131-3
Адрес ведомого устройства	Адрес ведомого устройства MODBUS	1	0 to 255
Тайм-аут связи (мс)	Устанавливает тайм-аут прикладного уровня	1000	10 to 65535
Максимальное количество повторных попыток	Устанавливает количество повторных попыток перед сообщением об ошибке связи	2	0 до 9
Конфигурация	Описание	Значение по умолчанию	Опции
Отключение отображения	Начальный адрес, используемый для отключения связи MODBUS	-	0 до 2147483644

Таблица 94: Конфигурация устройства - MODBUS MasterNotes:

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

Отключение отображения: состоит из 32 битов, используется для отключения по отдельности 32 связей MODBUS, сконфигурированных в пространстве Отображения устройств. Отношение отключено, если бит, соответствующий отношению, равен 1, в противном случае отображение включено. Данное поле ограничено размером адресуемой памяти переменных выходов (%Q) каждого процессора, который можно найти в разделе [Память](#).

Значение по умолчанию: заводское значение по умолчанию для поля Mapping Disabling не может быть установлено, так как 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.

Подробнее о параметрах Адрес ведомого устройства, тайм-аут связи и максимальное количество повторных попыток см. примечания в разделе [Конфигурация устройств - Конфигурация символьного отображения](#).

5. КОНФИГУРАЦИЯ

5.5.5.2.3. Конфигурация отображений - Конфигурация для прямого представления (%Q)

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

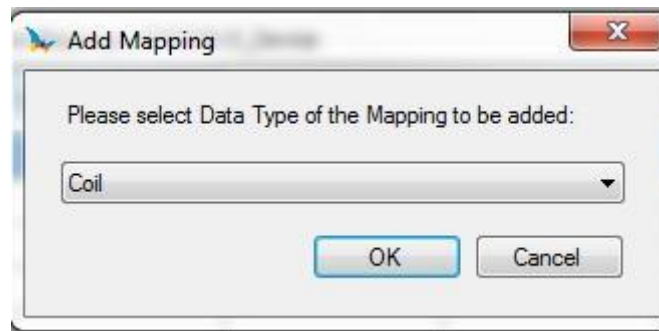


Рисунок 62: Тип данных MODBUS

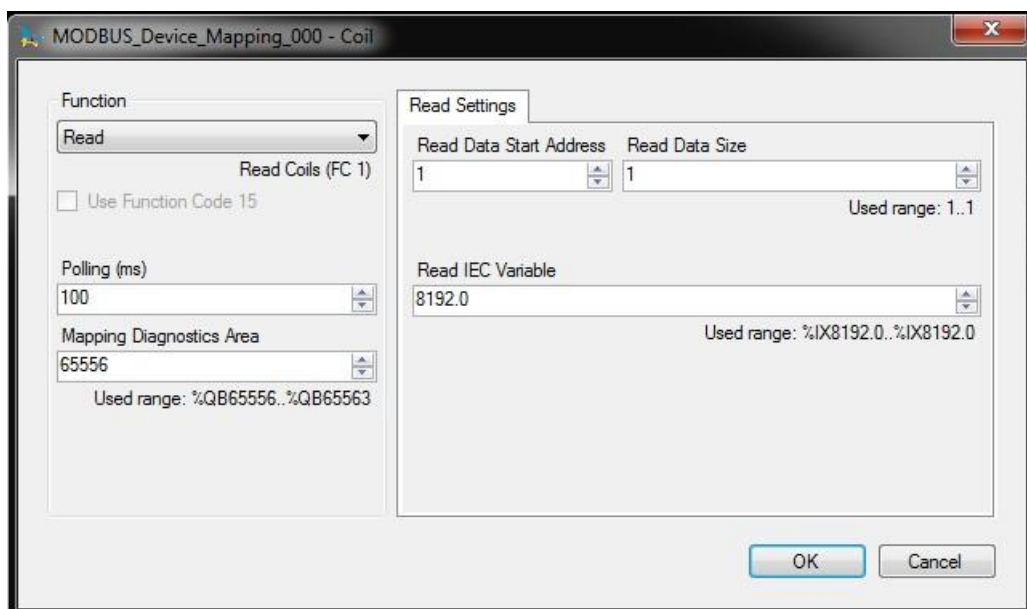


Рисунок 63: Функция MODBUS

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

Конфигурация	Описание	Значение по умолчанию	Опции
Функция	Тип функции MODBUS	Чтение	Чтение Запись Чтение/запись Маска Запись
Запрос (мс)	Период связи (мс)	100	0 до 3600000
Область диагностики	Начальный адрес диагностики связи MODBUS (%Q)	-	0 до 2147483640

5. КОНФИГУРАЦИЯ

Адрес начала чтения данных	Начальный адрес считывания данных MODBUS	1	1 до 65536
Размер считываемых данных	Количество считываемых данных MODBUS	-	Зависит от используемой функции
Чтение переменной IEC	Начальный адрес считываемых переменных (%I)	-	0 до 2147483646
Запись данных Начальный адрес	Начальный адрес записи данных MODBUS	1	1 до 65536
Запись данных Размер данных	Количество данных записи MODBUS	-	Зависит от используемой функции
Запись IEC-переменной	Начальный адрес переменных для записи (%Q)	-	0 до 2147483647
Маска Запись переменных IEC	Начальный адрес переменных для маски записи (%Q)	-	0 до 2147483644

Таблица 95: Сопоставление устройств

Примечания:

Функция: доступные типы данных подробно описаны в табл. 114, а функции MODBUS (FC) - в табл. 112.

Опрос: этот параметр указывает, как часто должна выполняться связь, установленная для данного отношения. По окончании связи будет ожидаться время, равное настроенному опросу, после чего будет выполнена новая связь как можно быстрее.

Область диагностики: это поле ограничено размером адресуемой памяти выходных переменных (%Q) процессора, который можно найти в разделе [Память](#). Конфигурируемая диагностика связей MODBUS описана в табл. 91.

Размер данных чтения/записи: подробное описание размера данных, поддерживаемых каждой функцией, приведено в примечаниях к разделу [Конфигурация запросов - Настройки символического отображения](#).

ВНИМАНИЕ

При обращении к памяти коммуникационных данных между устройствами с различным порядком следования байтов (прямой и обратный порядок) может произойти инверсия данных при чтении/записи. В этом случае пользователь должен скорректировать данные в приложении.

Чтение переменной IEC: если тип данных MODBUS - Coil или Input Status (бит), то начальный адрес переменных чтения IEC будет иметь, например, формат %IX10.1. Если же тип данных MODBUS - Holding Register или Input Register (16 бит), то начальный адрес переменных чтения IEC будет иметь формат %IW. Это поле ограничено размером адресуемой памяти входных переменных (%I) на CPU, который можно найти в разделе [Память](#).

Запись переменной IEC: если тип данных MODBUS - Coil, то начальный адрес переменных записи IEC будет иметь, например, формат %QX10.1. Если же тип данных MODBUS - Holding Register (16 бит), то начальный адрес переменных записи IEC будет иметь формат %QW. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) на ЦПУ, который можно найти в разделе [Память](#).

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

5. КОНФИГУРАЦИЯ

Значение по умолчанию: заводское значение по умолчанию для полей "Область диагностики отображения", "Чтение переменной МЭК", "Запись переменной МЭК" и "Маска записи переменных МЭК" 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 Read/Write Data Size нельзя установить заводское значение, так как они будут меняться в зависимости от выбранного типа данных MODBUS.

ВНИМАНИЕ

В отличие от других задач приложения, при достижении метки при отладке MainTask задача экземпляра MODBUS RTU Master или любая другая задача MODBUS прекращает выполнение 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 выполняется.

5.5.6. MODBUS RTU Slave

Этот протокол доступен для серии Nexto на ее последовательных каналах. При выборе этой опции в MasterTool IEC XE процессор становится ведомым устройством связи MODBUS, что позволяет подключаться к ведущим устройствам MODBUS RTU. Данный профиль доступен только в режиме исполнения (Run Mode).

Существует два способа конфигурирования этого протокола. Первый - через прямое представление (%Q), при котором переменные определяются своим адресом. Второй - через символическое отображение, при котором 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. Остальные шаги конфигурирования описаны ниже для каждого режима:

- Добавьте экземпляр протокола MODBUS RTU Slave в последовательный канал COM 1 или COM 2 (или оба, если речь идет о двух сетях связи). Для выполнения этой процедуры см. раздел "Вставка экземпляра протокола".
- Настроить последовательный интерфейс, выбрав скорость передачи данных, поведение сигналов RTS/CTS, четность, канал Стоп-биты и др. См. раздел Конфигурация последовательных интерфейсов.

5.5.6.1. Конфигурирование протокола MODBUS Slave с помощью символьного отображения

- Для настройки этого протокола с использованием символьного отображения необходимо выполнить следующие действия:
- Настроить общие параметры протокола MODBUS, такие как: адрес ведомого устройства и время обмена данными (доступно по кнопке "Дополнительные конфигурации ведомого устройства").
- Добавить и настроить связи MODBUS, указав имя переменной, тип данных MODBUS и начальный адрес данных.
- Автоматически будут заполнены размер и диапазон данных в соответствии с объявленным типом переменной.

5.5.6.1.1. Общие параметры протокола MODBUS Slave - Конфигурирование через символическое отображение

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

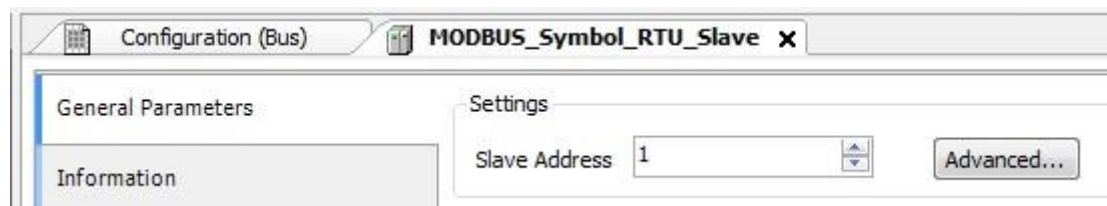


Рисунок 64: Экран конфигурации ведомого устройства MODBUS RTU

Конфигурация	Описание	Значение по умолчанию	Опции
Адрес ведомого устройства	MODBUS slave address	1	1 до 255

Таблица 96: Конфигурации ведомого устройства

Время обмена данными по ведомому протоколу MODBUS, указанное в кнопке Advanced... на экране конфигурации, делится на: *Task Cycle*, *Send Delay* и *Minimum Interframe*, как показано на рисунке ниже и в таблице ниже.

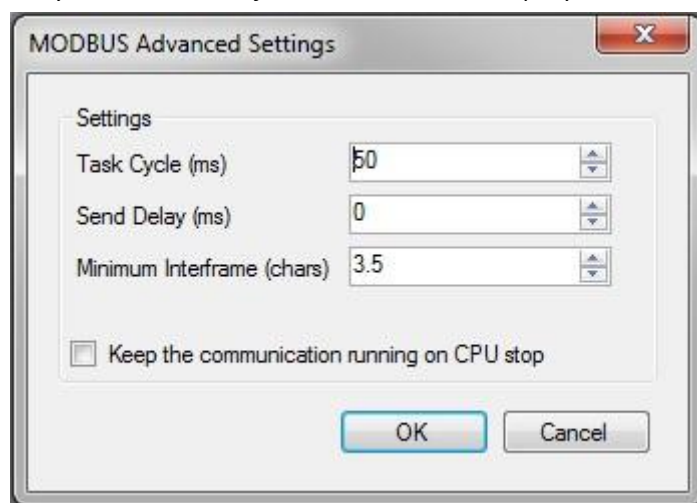


Рисунок 65: Дополнительные настройки Modbus Slave

Конфигурация	Описание	Значение по умолчанию	Возможности
Task Cycle (мс)	Время выполнения экземпляра в цикле, без учета собственного времени выполнения	50	20 до 100
Send Delay (мс)	Задержка ответа на передачу	0	0 до 65535
Конфигурация	Описание	Значение по умолчанию	Возможности
Минимальный интерфрейм (символов)	Минимальное время молчания между различными кадрами	3.5	3.5 до 100.0
Поддерживать связь при остановке процессора	Разрешить ведомому устройству MODBUS Symbol работать, когда процессор находится в состоянии ОСТАНОВ или после точки останова	Непроверенно	Проверено или не проверено

Таблица 97: Расширенные конфигурации Modbus Slave

5. КОНФИГУРАЦИЯ

Примечания:

Task Cycle: пользователь должен быть осторожен при изменении этого параметра, поскольку он непосредственно влияет на время ответа, объем данных для сканирования и, главным образом, на баланс ресурсов процессора между коммуникациями и другими задачами.

Send Delay: в некоторых моментах ответ на протокол MODBUS может вызывать проблемы, как, например, в случае интерфейса RS-485 или других полудуплексных интерфейсов. Иногда возникает задержка между временем ответа ведомого и молчанием физической линии (задержка ведомого на установку RTS в ноль и перевод RS-485 в высокоимпедансное состояние). Чтобы решить эту проблему, ведущий может выждать определенное в этом поле время перед отправкой нового запроса. В противном случае первые байты, переданные ведущим, могут быть потеряны.

Минимальный интерфрейм: стандарт MODBUS определяет это время как 3,5 символа, однако этот параметр может быть сконфигурирован для того, чтобы обслуживать устройства, которые не следуют стандарту.

Диагностика и команды протокола MODBUS Slave, сконфигурированные либо путем символического отображения, либо путем прямого представления, хранятся в переменных T_DIAG_MODBUS_RTU_SLAVE_1. При прямом отображении они также находятся в 4 байтах и 8 словах, которые описаны в таблице ниже (где "n" - сконфигурированное значение в %Q Стартовый адрес области диагностики):

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

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_SLAVE_1.*	Размер	Описание
----------------------------------	--	--------	----------

5. КОНФИГУРАЦИЯ

%QB(n+1)	eErrorCode	SERIAL_STATUS (BYTE)	<p>0: ошибка отсутствует. 1: недопустимый последовательный порт. 2: недопустимый режим работы последовательного порта 3: недопустимая скорость передачи данных 4: недопустимые Биты данных 5: недопустимая четность 6: недопустимые Стоп-биты 7: недопустимый параметр сигнала модема 8: недопустимый параметр UART RX Threshold 9: недопустимый параметр тайм-аута 10: последовательный порт занят 11: аппаратная ошибка UART 12: аппаратная ошибка удаленного устройства 20: недопустимый размер буфера передачи 21: недопустимый способ передачи сигнала модема 22: тайм-аут CTS = true 23: тайм-аут CTS = false 24: ошибка тайм-аута передачи 30: недопустимый размер буфера приема 31: ошибка тайм-аута приема 32: управление потоком сконфигурировано не так, как указано в руководстве 33: недопустимое управление потоком для сконфигурированного последовательного порта 34: прием данных в нормальном режиме запрещен 35: прием данных в расширенном режиме запрещен 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>
Командные биты, инициализируемые автоматически:			
%QX(n+2).0	tCommand. bStop	БИТ	Остановить ведомое устройство.

5. КОНФИГУРАЦИЯ

%QX(n+2).1	tCommand. bRestart	БИТ	Перезапустить ведомое устройство.
%QX(n+2).2	tCommand. bResetCounter	БИТ	Перезапуск статистики диагностики (счетчиков).
%QX(n+2).3	tCommand. bDiag_19_reserved	БИТ	Зарезервировано.
%QX(n+2).4	tCommand. bDiag_20_reserved	БИТ	Зарезервировано.
%QX(n+2).5	tCommand. bDiag_21_reserved	БИТ	Зарезервировано.
%QX(n+2).6	tCommand. bDiag_22_reserved	БИТ	Зарезервировано.
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_SLAVE_1.*	Размер	Описание
%QX(n+2).7	tCommand. bDiag_23_reserved	БИТ	Зарезервировано.
%QB(n+3)	byDiag_3_reserved	БАЙТ	Зарезервировано.
Статистика коммуникаций:			
%QW(n+4)	tStat. wRXRequests	СЛОВО	Счетчик нормальных запросов, полученных ведомым устройством и получивших нормальный ответ. В случае широковещательной команды этот счетчик инкрементируется, но не передается (от 0 до 65535).
%QW(n+6)	tStat. wTXExceptionResponses	СЛОВО	Счетчик обычных запросов, полученных ведомым устройством и ответивших на них кодом исключения. В случае широковещательной команды этот счетчик увеличивается, но не передается (от 0 до 65535). Коды исключений: 1: код функции (FC) разрешен, но не поддерживается. 2: отношение не встречается в данных MODBUS. 3: недопустимое значение для данного поля. 128: ведущий/клиент не имеет права на запись или чтение. 129: связь MODBUS отключена.
%QW(n+8)	tStat. wRXFrames	СЛОВО	Счетчик кадров, принятых ведомым устройством. Считается, что кадр - это то, что обрабатывается, и за ним следует минимальное межкадровое молчание, другими словами,

5. КОНФИГУРАЦИЯ

			вычисляется также нелегальное сообщение (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalRequests	СЛОВО	Счетчик нелегальных запросов. Это кадры, которые начинаются с адреса 0 (широковещание) или с адреса ведомого устройства MODBUS, но не являются легальными запросами - неверный синтаксис, меньшие кадры, неверный CRC - (от 0 до 65535).
%QW(n+12)	tStat. wRXOverrunErrors	СЛОВО	Счетчик кадров с ошибками обгона при приеме - UART FIFO или линия RX - (от 0 до 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	СЛОВО	Счетчик кадров с ошибками построения, четности или сбоя при приеме - (от 0 до 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	СЛОВО	Счетчик ошибок тайм-аута CTS, использующего рукопожатие RTS/CTS, во время передачи (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	СЛОВО	Зарезервировано.

Таблица 98: Диагностика ведомого устройства MODBUS RTU

Примечание:

Счетчики: все счетчики диагностики MODBUS RTU Slave возвращаются к нулю при превышении предельного значения 65535.

5.5.6.1.2. Конфигурация отношений - настройка символьного отображения

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

Mappings

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

Рисунок 66: Экран сопоставления данных MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Значение Переменная	Символьное имя переменной	-	Имя переменной, объявленной в программе или GVL
Тип данных	Тип данных MODBUS	-	Катушка Состояние входа Регистр удержания Входной регистр

5. КОНФИГУРАЦИЯ

Адрес начала данных	Начальный адрес данных MODBUS	-	1 до 65536
Абсолютный адрес начала данных	Абсолютный начальный адрес данных MODBUS в соответствии с их типом	-	-
Размер данных	Размер данных MODBUS	-	1 до 65536
Диапазон данных	Конфигурируемый диапазон адресов данных	-	-

Таблица 99: Конфигурации сопоставления MODBUS

Примечания:

Переменная значения: это поле используется для указания символьной переменной в связи MODBUS.

Тип данных: это поле используется для указания типа данных, используемого в связи MODBUS.

Тип данных	Размер [бит]	Описание
Катушка	1	Цифровой выход, который может быть прочитан или записан.
Состояние входа	1	Цифровой вход (только чтение).
Регистр удержания	16	Аналоговый выход, который может быть считан или записан.
Входной регистр	16	Аналоговый вход (только чтение).

Таблица 100: Типы данных MODBUS, поддерживаемые процессорами Nexto

Начальный адрес данных: начальный адрес данных отношения MODBUS.

Размер данных: значение Размер данных задает максимальный объем данных, к которому может обращаться отношение MODBUS с начального адреса. Таким образом, для чтения непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном отношении. Это поле изменяется в зависимости от настроенного типа данных MODBUS.

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

ВНИМАНИЕ

В отличие от других прикладных задач, при достижении метки разрядности в MainTask задача экземпляра MODBUS RTU Slave и любая другая задача MODBUS прекращает работу в тот момент, когда пытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока основная задача не выполняется.

5.5.6.2. Конфигурирование протокола MODBUS Slave через прямое представление (%Q)

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

- Настройка общих параметров ведомого протокола MODBUS, таких как: время обмена данными, адрес и переменные прямого представления (%Q) для получения диагностических и управляющих связей.
- Добавление и настройка отношений MODBUS с указанием типа данных MODBUS, переменных прямого представления (%Q) для приема/записи данных и количества данных для передачи.

5. КОНФИГУРАЦИЯ

Описание каждой настройки приведено ниже, в этом разделе.

5.5.6.2.1. Общие параметры протокола MODBUS Slave - конфигурирование через прямое представление (%Q)

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

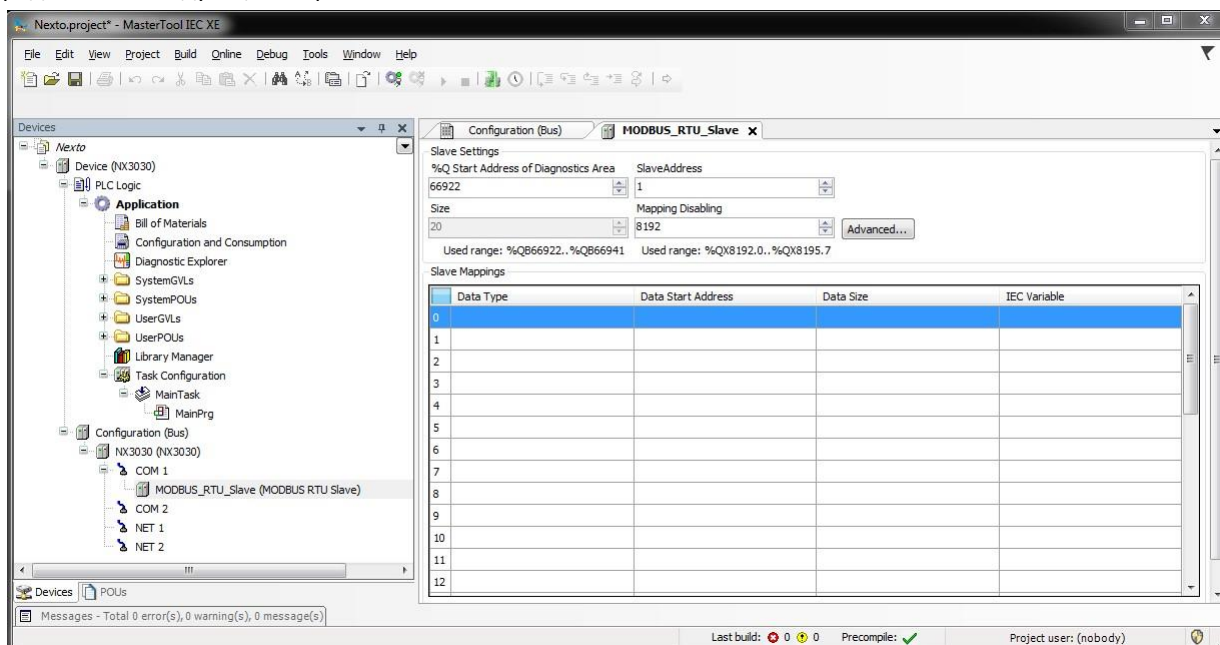


Рисунок 67: Экран конфигурации ведомого устройства MODBUS RTU

Обращение и направление переменных представления (%Q) к управляющим связям и диагностике:

Конфигурация	Описание	Значение по умолчанию	Опции
%Q Начальный адрес области диагностики	Начальный адрес диагностических переменных	-	0 до 2147483628
Размер	Размер области диагностики	-	Отключено для редактирования
Адрес ведомого устройства	Адрес ведомого устройства MODBUS	1	1 до 255
Отключение отображения	Начальный адрес, используемый для отключения связей MODBUS	-	0 до 2147483644

Таблица 101: Настройки переменных адреса и прямого представления

Примечания:

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

Адрес ведомого: важно отметить, что ведомый принимает запросы широковещательно, когда ведущий посылает команду с адресом, установленным в ноль. Кроме того, в соответствии со стандартом MODBUS, диапазон допустимых адресов ведомых устройств составляет от 1 до 247. Адреса с 248 по 255 являются зарезервированными.

5. КОНФИГУРАЦИЯ

Mapping Disabling: состоит из 32 битов и используется для индивидуального отключения 32 отношений MODBUS, сконфигурированных в пространстве Slave Mappings. Отношение отключено, если соответствующий бит равен 1, в противном случае отображение включено. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе [Память](#).

Значение по умолчанию: заводское значение по умолчанию для полей %Q Стартовый адрес области диагностики и отключение отображения не может быть установлено, так как создание связи может быть выполнено в любой момент при разработке приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), пока неиспользуемое.

Диагностика и команды протокола MODBUS описаны в табл. 98.

Время обмена данными по протоколу MODBUS Slave, найденное по кнопке Advanced... экрана конфигурации, описано в табл. 97.

5.5.6.2.2. Конфигурация отображений - Конфигурация через прямое представление (%Q)

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

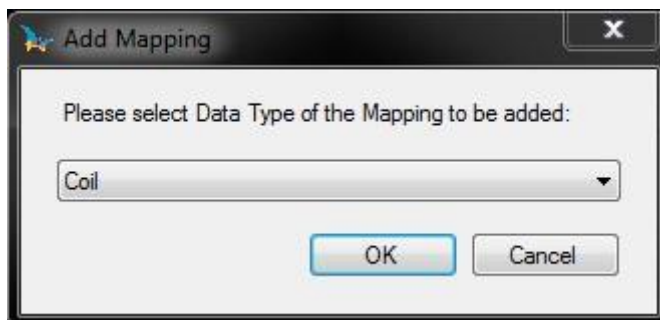


Рисунок 68: Добавление связей MODBUS

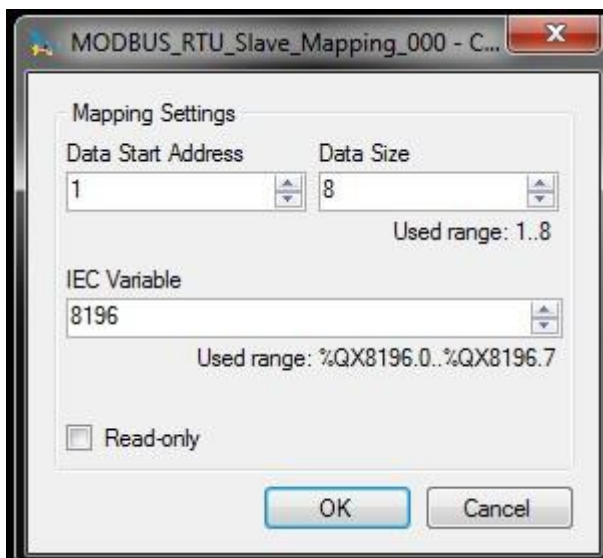


Рисунок 69: Настройка отношения MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Тип данных	Тип данных MODBUS	Катушка	Катушка (1 бит) Регистр удержания (16 бит) Входной регистр (16 бит)

5. КОНФИГУРАЦИЯ

			Состояние входа (1 бит)
Адрес начала данных	Начальный адрес данных MODBUS	1	1 до 65536
Размер данных	Количество данных MODBUS	-	1 до 65536
Переменная IEC	Начальный адрес переменных (%Q)	-	0 до 2147483647
Только для чтения	Разрешает только чтение	Отключено	Включено или отключено

Таблица 102: Сопоставления ведомых устройств

Примечания:

Опции: значения, записанные в столбце Опции, могут меняться в зависимости от конфигурации данных MODBUS.

Размер данных: значение параметра Размер данных определяет максимальный объем данных, к которому может получить доступ отношение MODBUS, начиная с начального адреса. Таким образом, для чтения непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном интерфейсе. Это поле зависит от настроенного типа данных MODBUS, т.е. при выборе Coil или Input Status поле Data Size должно быть кратно восьми. Кроме того, максимальный размер не должен превышать размер выходной адресуемой памяти и не должен присваивать те же значения, которые используются в приложении.

ВНИМАНИЕ

При обращении к памяти коммуникационных данных между устройствами с различным порядком (прямой и обратный порядок) может произойти инверсия данных при чтении/записи. В этом случае пользователь должен скорректировать данные в приложении.

Переменная IEC: если тип данных MODBUS - Coil или Input Status (бит), то начальный адрес переменных IEC будет в формате %QX10.1. Если же тип данных MODBUS - Holding Register или Input Register (16 бит), то начальный адрес переменных IEC будет в формате %QW. Это поле ограничено объемом памяти адресуемых выходных переменных (%Q) от каждого процессора, который можно посмотреть в разделе [Память](#).

Только для чтения: если эта опция включена, то она позволяет ведущему устройству связи только читать данные переменной. Запись не разрешается. Эта опция действительна только для функций записи.

Значение по умолчанию: значение по умолчанию для поля IEC Variable не может быть определено, так как создание связи может быть выполнено в любой момент при разработке приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), пока неиспользуемое. Для поля Data Size нельзя определить значение по умолчанию, так как оно будет зависеть от выбранного типа данных MODBUS.

В ранее определенных отношениях максимальный размер данных MODBUS может составлять 65535 (максимальное значение, настроенное в поле Data Size). Однако запрос, поступающий на ведомое устройство MODBUS RTU, должен обращаться к подгруппе этого отображения, и эта группа должна иметь, максимум, размер данных, зависящий от кода функции, который определен ниже:

- Считывание катушек (FC 1): 2000
- Чтение состояния входа (FC 2): 2000
- Чтение регистров удержания (FC 3): 125
- Чтение входных регистров (FC 4): 125
- Запись одиночной катушки (FC 5): 1

5. КОНФИГУРАЦИЯ

- Запись одиночного регистра удержания (FC 6): 1
- Принудительная запись нескольких катушек (FC 15): 1968
- Запись регистров удержания (FC 16): 123
- Регистр записи маски (FC 22): 1
- Чтение/запись Регистры удержания (FC 23):
 - Чтение: 121
 - Запись: 121

ВНИМАНИЕ

В отличие от других прикладных задач, при достижении метки разрядности в MainTask, задача Slave экземпляра MODBUS RTU и любая другая задача MODBUS прекращает работу в тот момент, когда пытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока основная задача не выполняется.

5.5.7. MODBUS Ethernet

Мультимастерная связь позволяет центральным процессорам Nexto считывать или записывать переменные MODBUS в других контроллерах или HMI, совместимых с протоколом MODBUS TCP или MODBUS RTU через TCP. Процессор Nexto CPU может быть одновременно и клиентом, и сервером в одной коммуникационной сети, или даже иметь несколько экземпляров, связанных с интерфейсом Ethernet. При этом не имеет значения, являются ли они MODBUS TCP или MODBUS RTU через TCP, как описано в табл. 65.

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

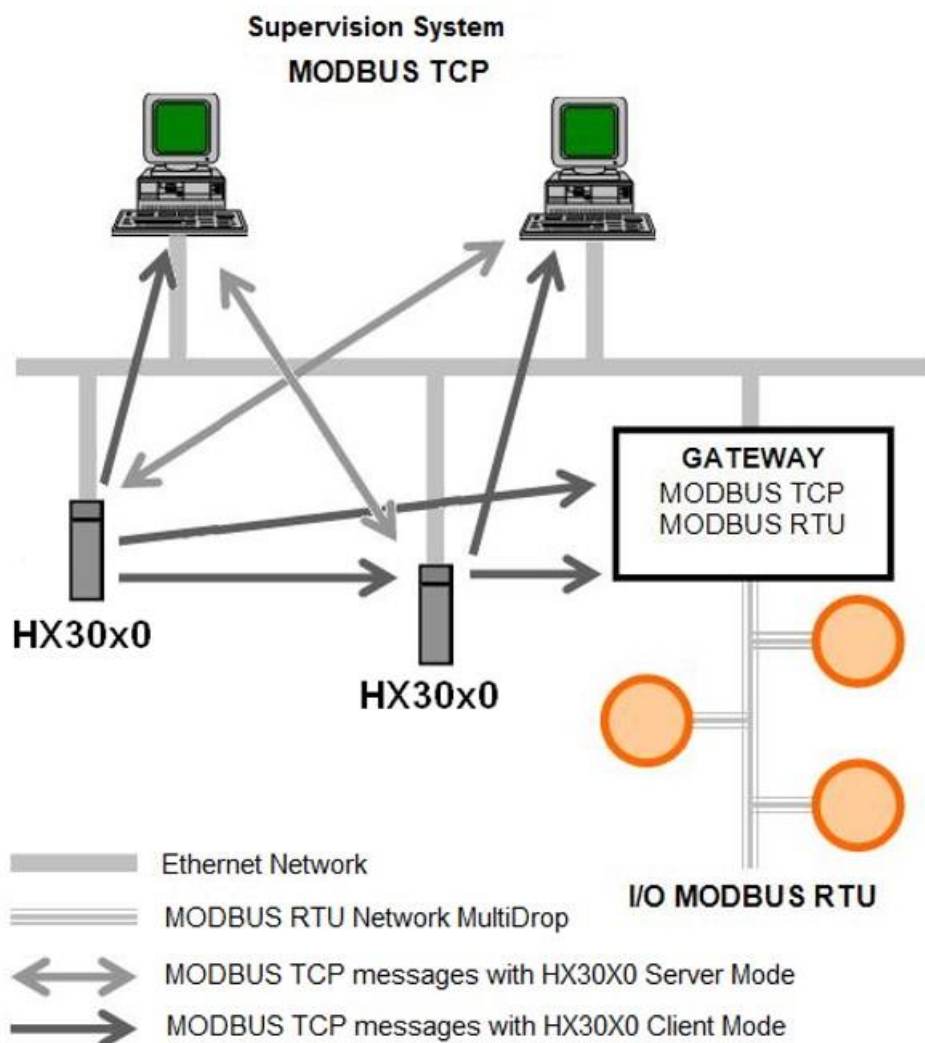


Рисунок 70: Сеть передачи данных MODBUS TCP

Связь переменных MODBUS с символьными переменными процессора осуществляется пользователем путем определения отношений с помощью конфигуратора MasterTool IEC XE. В серверном режиме можно сконфигурировать до 32 связей, а в клиентском - до 128. При этом отношения в клиентском режиме должны соответствовать максимальному размеру данных функции MODBUS: 125 регистров (входные регистры или регистры удержания) или 2000 бит (катушки или состояние входа). Эта информация подробно изложена в описании каждого протокола.

Все связи, как в клиентском, так и в серверном режиме, могут быть отключены через переменные прямого представления (%Q), которые в MasterTool IEC XE обозначены как 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 записи и чтения, называемыми фильтрами. Это делается через определение сетевого IP-адреса и маски подсети, в результате чего образуется группа клиентских IP-адресов, которые могут читать и записывать в переменные отношения. Функции чтения/записи фильтруются, другими словами, они не могут быть запрошены ни одним клиентом, независимо от IP-адреса. Эта информация подробно описана в протоколе MODBUS Ethernet Server.

5. КОНФИГУРАЦИЯ

При использовании протокола MODBUS TCP в режиме клиента для ускорения взаимодействия с серверами можно использовать функцию множественных запросов, используя одно и то же TCP-соединение. Если эта функция не нужна или 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 процессорами Nexto.

Тип данных	Размер [бит]	Описание
Катушка	1	Цифровой выход, который может быть прочитан или записан.
Состояние входа	1	Цифровой вход (только чтение).
Регистр удержания	16	Аналоговый выход, который может быть считан или записан.
Входной регистр	16	Аналоговый вход (только чтение).

Таблица 103: Типы данных MODBUS, поддерживаемые процессорами Nexto

Код		Описание
DEC	HEX	
1	0x01	Считывание катушек (FC 01)
2	0x02	Чтение состояния входов (FC 02)
3	0x03	Чтение регистров удержания (FC 03)
4	0x04	Чтение входных регистров (FC 04)
5	0x05	Запись одной катушки (FC 05)
6	0x06	Запись одного регистра удержания (FC 06)
15	0x0F	Запись нескольких катушек (FC 15)
16	0x10	Запись нескольких регистров удержания (FC 16)
22	0x16	Маска записи регистра удержания (FC 22)
23	0x17	Чтение/запись нескольких регистров удержания (FC 23)

Таблица 104: Функции MODBUS, поддерживаемые процессорами Nexto

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

- Добавьте в канал Ethernet один или несколько экземпляров клиентского или серверного протокола MODBUS Ethernet. Для выполнения этой процедуры обратитесь к разделу "Вставка экземпляра протокола".
- Сконфигурировать интерфейс Ethernet. Для выполнения этой процедуры обратитесь к разделу Конфигурация интерфейсов Ethernet.

5.5.8. MODBUS Ethernet Client

5. КОНФИГУРАЦИЯ

Этот протокол доступен для всех процессоров серии Nexto на их каналах Ethernet. При выборе этой опции в MasterTool IEC XE процессор становится клиентом связи MODBUS, что позволяет ему в режиме исполнения (Run Mode) получать доступ к другим устройствам с тем же протоколом.

Существует два способа конфигурирования этого протокола. Первый - через прямое представление (%Q), при котором переменные определяются своим адресом. Второй - через символическое отображение, 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](#)".

5.5.8.1. Конфигурирование клиента MODBUS Ethernet с помощью символического отображения

Для конфигурирования этого протокола с использованием символического отображения необходимо выполнить следующие действия:

- настройка общих параметров клиента протокола MODBUS, с протоколом управления передачей (TCP) или RTU через TCP.
- Добавить и сконфигурировать устройства, задав IP-адрес, порт, адрес ведомого устройства и тайм-аут связи (доступно по кнопке Advanced Settings устройства).
- Добавьте и настройте сопоставления MODBUS, указав имя переменной, тип данных, начальный адрес данных, размер данных и переменную, которая будет получать данные о качестве.
- Добавить и настроить запрос MODBUS, указав желаемую функцию, время сканирования запроса, начальный адрес (чтение/запись), размер данных (чтение/запись), переменную, которая будет получать данные о качестве, и переменную, отвечающую за отключение запроса.

5.5.8.1.1. Общие параметры клиентского протокола MODBUS - конфигурирование через символическое отображение

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

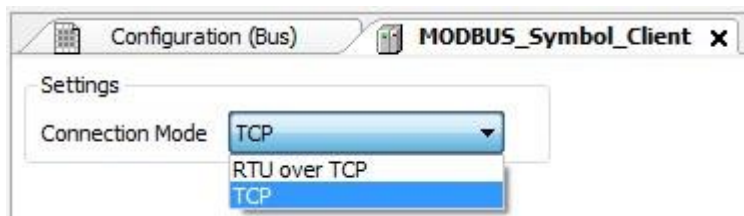


Рисунок 71: Экран конфигурации общих параметров клиента MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Режим подключения	Выбор протокола	TCP	RTU через TCP TCP

Table 105: MODBUS Client General Configurations

Диагностика и команды протокола MODBUS Client, сконфигурированные либо по символическому отображению, либо по прямому представлению, хранятся в переменных T_DIAG_MODBUS_ETH_CLIENT_1. При прямом отображении они также находятся в 4 байтах и 8 словах, которые описаны в таблице ниже (где "n" - сконфигурированное значение в %Q Начальный адрес области диагностики):

5. КОНФИГУРАЦИЯ

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_CLIENT_1.*	Размер	Описание
Диагностические биты:			
%QX(n).0	tDiag. bRunning	БИТ	Клиент находится в режиме выполнения.
%QX(n).1	tDiag. bNotRunning	БИТ	Клиент не находится в режиме выполнения (см. бит bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	БИТ	Бит bNotRunning был включен, так как клиент был прерван пользователем через командные биты.
%QX(n).3	tDiag. bConfigFailure	БИТ	Прерванная диагностика.
%QX(n).4	tDiag. bRXFailure	БИТ	Прекращенная диагностика.
%QX(n).5	tDiag. bTXFailure	БИТ	Прекращенная диагностика.
%QX(n).6	tDiag. bModuleFailure	БИТ	Указывает на наличие сбоя в модуле или на отсутствие модуля.
%QX(n).7	tDiag. bAllDevicesCommFailure	БИТ	Указывает на то, что все устройства, сконфигурированные в клиенте, находятся в состоянии отказа.
%QB(n+1)	byDiag_1_reserved	БАЙТ	Зарезервировано.
Командные биты, инициализируемые автоматически:			
%QX(n+2).0	tCommand. bStop	БИТ	Остановка пользователя.
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_CLIENT_1.*	Размер	Описание
%QX(n+2).1	tCommand. bRestart	БИТ	Перезапустить клиента.
%QX(n+2).2	tCommand. bResetCounter	БИТ	Перезапустить диагностическую статистику (счетчики).
%QX(n+2).3	tCommand. bDiag_19_reserved	БИТ	Зарезервировано.
%QX(n+2).4	tCommand. bDiag_20_reserved	БИТ	Зарезервировано.

5. КОНФИГУРАЦИЯ

%QX(n+2).5	tCommand. bDiag_21_reserved	БИТ	Зарезервировано.
%QX(n+2).6	tCommand. bDiag_22_reserved	БИТ	Зарезервировано.
%QX(n+2).7	tCommand. bDiag_23_reserved	БИТ	Зарезервировано.
%QB(n+3)	byDiag_3_reserved	БАЙТ	Зарезервировано.
Статистика коммуникаций:			
%QW(n+4)	tStat. wTXRequests	СЛОВО	Счетчик количества запросов, переданных клиентом (от 0 до 65535).
%QW(n+6)	tStat. wRXNormalResponses	СЛОВО	Счетчик нормальных ответов, полученных клиентом (от 0 до 65535).
%QW(n+8)	tStat. wRXExceptionResponses	СЛОВО	Счетчик ответов с кодом исключения (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalResponses	СЛОВО	Счетчик неправомерных ответов, полученных клиентом, - неверный синтаксис, неверный CRC или недостаточно полученных байт (от 0 до 65535).
%QW(n+12)	tStat. wDiag_12_reserved	СЛОВО	Зарезервировано.
%QW(n+14)	tStat. wDiag_14_reserved	СЛОВО	Зарезервировано.
%QW(n+16)	tStat. wDiag_16_reserved	СЛОВО	Зарезервировано.
%QW(n+18)	tStat. wDiag_18_Reserved	СЛОВО	Зарезервировано.

Таблица 106: Диагностика протокола клиента MODBUS

Примечание:

Счетчики: все счетчики диагностики MODBUS TCP Client возвращаются к нулю при превышении предельного значения 65535.

5.5.8.1.2. Конфигурация устройства - Конфигурация через символьное отображение

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

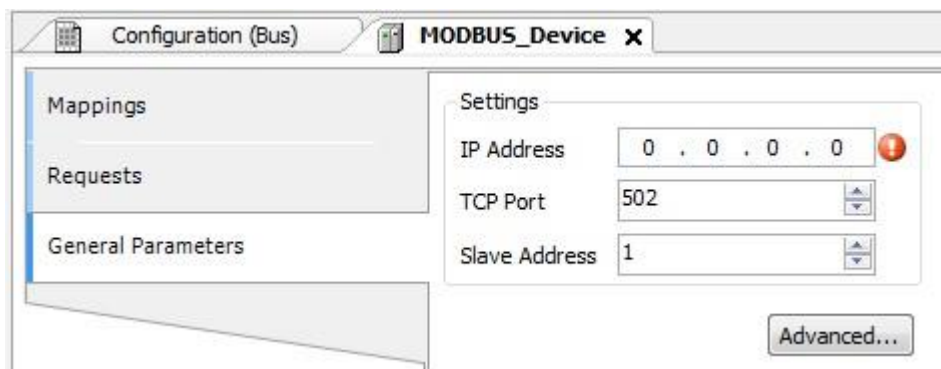


Рисунок 72: Настройки общих параметров устройства

Конфигурация	Описание	Значение по умолчанию	Опции
IP-адрес	IP-адрес сервера	0.0.0.0	1.0.0.1 до 223.255.255.255
TCP-порт	TCP-порт	502	2 до 65534
Адрес ведомого	Адрес ведомого устройства MODBUS	1	0 до 255

Таблица 107: Общие конфигурации клиента MODBUS

Примечания:

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

TCP-порт: если в один Ethernet-интерфейс добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные TCP-порты. Некоторые TCP-порты, среди перечисленных выше, зарезервированы и поэтому не могут быть использованы. См. таблицу [Зарезервированные TCP/UDP-порты](#).

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 ведущий посылает команду записи с адресом, установленным в ноль, он выполняет широковещательные запросы по сети.

Параметры в расширенных настройках устройства MODBUS Client, находящиеся по кнопке Advanced... на вкладке General Parameters, делятся на: Максимальное количество одновременных запросов, Тайм-аут связи, Тайм-аут режима связи и Время бездействия.

Конфигурация	Описание	Значение по умолчанию	Опции
Максимальный одновременный запрос	Количество одновременных запросов, которые клиент может запросить у сервера	1	1 до 8
Тайм-аут связи (мс)	Тайм-аут на уровне приложения в мс	3000	10 до 65535
Режим	Определяет момент завершения соединения клиента с сервером	Соединение закрывается по истечении времени бездействия (с): 10 - 3600.	Соединение закрывается по истечении тайм-аута. Соединение закрывается по окончании каждого обмена данными. Соединение закрывается по истечении времени

5. КОНФИГУРАЦИЯ

			бездействия (с): от 10 до 3600.
Время бездействия (с)	Время бездействия	10	3600

Таблица 108: Дополнительные настройки клиента MODBUS

Примечания:

Максимальное количество одновременных запросов: используется при высоком цикле сканирования. Этот параметр фиксирован в 1 (не редактируется), если конфигурируемый протокол - MODBUS RTU over TCP.

Тайм-аут связи: тайм-аут связи - это время, в течение которого клиент будет ожидать ответа сервера на запрос. Для устройства MODBUS Client необходимо учитывать две переменные системы: время обработки запроса сервером и задержку отправки ответа, если она задана на сервере. Рекомендуется, чтобы тайм-аут был равен или превышал удвоенную сумму этих параметров. Более подробную информацию можно найти в разделе [Производительность связи](#).

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

- Соединение закрывается по истечении тайм-аута или соединение никогда не закрывается в обычных ситуациях: Эти варианты представляют собой аналогичное поведение клиента, закрывающего соединение из-за отсутствия ответа на запрос со стороны сервера до достижения тайм-аута связи.
- Соединение закрывается по окончании каждой связи: 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.
- Время бездействия: время бездействия соединения.

5.5.8.1.3. Конфигурация отображений - Конфигурация через символическое отображение

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



Рисунок 73: Тип данных MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Значение переменной	Символьное имя переменной	-	Имя переменной, объявленной в программе или GVL
Конфигурация	Описание	Значение по умолчанию	Опции
Тип данных	Тип данных MODBUS	-	Катушка - запись (1 бит) Катушка - чтение (1 бит) Регистр удержания - запись (16 бит) Регистр удержания - чтение (16 бит) Регистр удержания - Маска И (16 бит) Регистр удержания - Маска или (16 бит) Входной регистр (16 бит) Состояние входа (1 бит)
Адрес начала данных	Начальный адрес данных MODBUS	-	1 до 65536
Размер данных	Размер данных MODBUS	-	1 до 65536
Диапазон данных	Диапазон адресов конфигурируемых данных	-	-

Таблица 109: Настройки отображения MODBUS

Примечания:

Значение переменной: это поле используется для указания символьной переменной в связи MODBUS.

Тип данных: это поле используется для указания типа данных, используемого в связи MODBUS.

Тип данных	Размер [бит]	Описание
Катушка - запись	1	Запись цифрового вывода.
Катушка - чтение	1	Чтение цифрового выхода.
Регистр удержания - запись	16	Запись аналогового выхода.
Регистр удержания - чтение	16	Чтение аналогового выхода.
Регистр удержания - Маска и	16	Аналоговый выход, который может быть считан или записан по маске.
Регистр удержания - маска или	16	Аналоговый выход, который может быть считан или записан по маске.
Входной регистр	16	Аналоговый вход, который может быть только считан.
Состояние входа	1	Цифровой вход, который может быть только считан.

5. КОНФИГУРАЦИЯ

Таблица 110: Типы данных, поддерживаемые в MODBUS

Начальный адрес данных: Начальный адрес данных в отображении MODBUS.

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

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

5.5.8.1.4. Конфигурация запросов - Конфигурация через символическое отображение

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

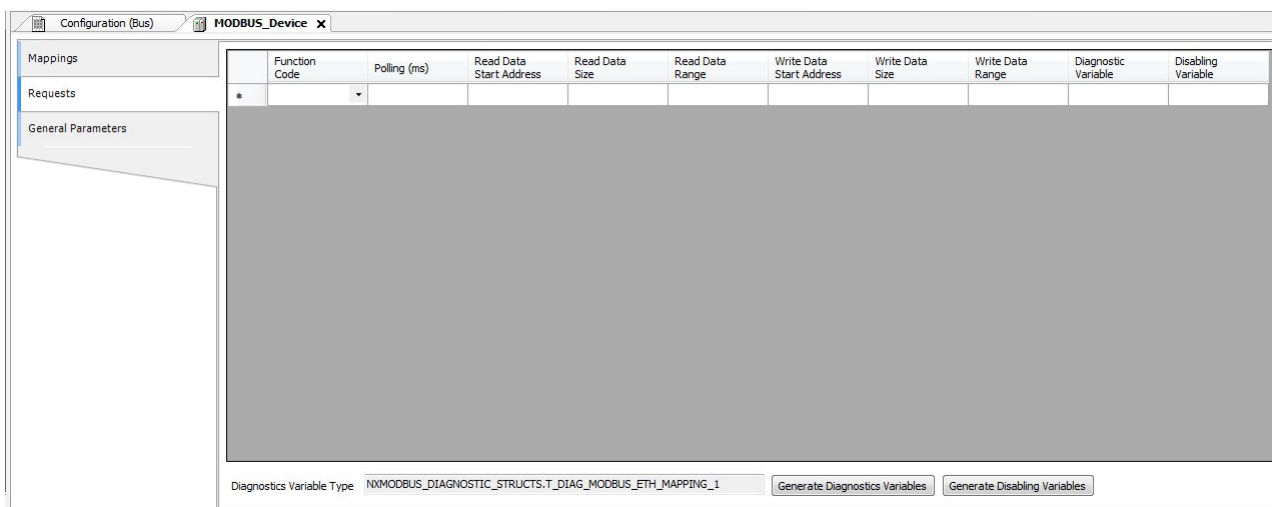


Рисунок 74: Экран запроса данных MODBUS

Конфигурация	Описание	Значение по умолчанию	Опции
Код функции	MODBUS тип функции	-	01 - Считывание катушек 02 - Чтение состояния входов 03 - Чтение регистров удержания 04 - Чтение входных регистров 05 - Запись одной катушки 06 - Запись одного регистра 15 - Запись нескольких катушек 16 - Запись нескольких регистров 22 - Маска записи регистра 23 - Чтение/запись нескольких

5. КОНФИГУРАЦИЯ

			Регистры
Опрос (мс)	Период обмена данными (мс)	100	0 до 3600000
Адрес начала чтения данных	Начальный адрес данных чтения MODBUS	-	1 до 65536
Размер считываемых данных	Размер данных для чтения MODBUS	-	Зависит от используемой функции
Диапазон считываемых данных	Диапазон адресов данных считывания MODBUS	-	0 до 2147483646
Запись данных Начальный адрес	Начальный адрес данных записи MODBUS	-	1 до 65536
Размер записываемых данных	Размер данных записи MODBUS	-	Зависит от используемой функции
Диапазон записываемых данных	Диапазон адресов данных записи MODBUS	-	0 до 2147483647
Диагностическая переменная	Имя диагностической переменной	-	Имя переменной, объявленной в программе или GVL
Конфигурация	Описание	Значение по умолчанию	Опции
Отключение переменной	Переменная, используемая для отключения связи с MODBUS	-	Поле для символьной переменной, используемой для индивидуального отключения сконфигурированных запросов MODBUS. Эта переменная должна иметь тип BOOL. Переменная может быть простым элементом или элементом массива и может находиться в структурах.

Таблица 111: Конфигурация отношений MODBUS

Примечания:

Установка: количество заводских установок по умолчанию и значения для столбца Опции могут отличаться в зависимости от типа данных и функции MODBUS (FC).

Код функции: Доступны следующие функции MODBUS (FC):

Код		Описание
DEC	HEX	
1	0x01	Считывание катушек (FC 01)
2	0x02	Чтение состояния входов (FC 02)
3	0x03	Чтение регистров удержания (FC 03)

5. КОНФИГУРАЦИЯ

4	0x04	Чтение входных регистров (FC 04)
5	0x05	Запись одной катушки (FC 05)
6	0x06	Запись одного регистра удержания (FC 06)
15	0x0F	Запись нескольких катушек (FC 15)
16	0x10	Запись нескольких регистров удержания (FC 16)
22	0x16	Маска записи регистра удержания (FC 22)
23	0x17	Чтение/запись нескольких регистров удержания (FC 23)

Таблица 112: Функции MODBUS, поддерживаемые процессорами Nexto

Опрос: этот параметр указывает, как часто должна выполняться связь, установленная для данного запроса. По окончании обмена данными будет ожидаться время, равное значению, заданному в поле polling, после чего будет выполнен новый обмен данными.

Начальный адрес чтения данных: поле для указания начального адреса чтения данных MODBUS.

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

- Считывание катушек (FC 01): 2000
- Чтение состояния входа (FC 02): 2000
- Чтение регистров удержания (FC 03): 125
- Чтение входных регистров (FC 04): 125
- Чтение/запись множественных регистров (FC 23): 121

Диапазон данных чтения: в этом поле отображается диапазон данных чтения MODBUS, настроенный для каждого запроса. Начальный адрес вместе с размером считываемых данных определяют диапазон считываемых данных для каждого запроса.

Адрес начала записи данных: поле для начального адреса данных записи MODBUS.

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

- Запись одиночной катушки (FC 05): 1
- Запись одного регистра (FC 06): 1
- Запись нескольких катушек (FC 15): 1968
- Запись нескольких регистров (FC 16): 123
- Регистр записи маски (FC 22): 1
- Чтение/запись нескольких регистров (FC 23): 121

Диапазон данных записи: в этом поле отображается диапазон данных записи MODBUS, настроенный для каждого запроса. Начальный адрес вместе с размером считываемых данных определяют диапазон данных записи для каждого запроса.

Diagnostic Variable (Диагностическая переменная): Диагностика запросов MODBUS, сконфигурированная по символному отображению или по прямому представлению, хранится в переменных типа T_DIAG_MODBUS_RTU_MAPPING_1 для устройств Master и T_DIAG_MODBUS_ETH_CLIENT_1 для устройств Client, а отображение по прямому представлению - в 4-байтовых и 2-словных, которые описаны в таблице 91 (n - значение, заданное в поле %Q Начальный адрес области диагностики).

Переменная прямого представления	Диагностическая переменная	Размер	Описание
	T_DIAG_MODBUS_ETH_MAPPING_1.*		

5. КОНФИГУРАЦИЯ

Биты состояния связи:			
%QX(n).0	byStatus. bCommIdle	БИТ	Коммуникация простаивает (ожидание выполнения).
%QX(n).1	byStatus. bCommExecuting	БИТ	Активная связь.
%QX(n).2	byStatus. bCommPostponed	БИТ	Отложенное сообщение, поскольку было 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 при определении таймаута. Биты bCommIdle и bCommExecuting ложны, если бит bCommPostponed истинен.
%QX(n).3	byStatus. bCommВыключено	БИТ	Связь отключена. В этом состоянии бит bCommIdle перезапускается.
%QX(n).4	byStatus. bCommOk	БИТ	Связь, прерванная ранее, была проведена успешно.
%QX(n).5	byStatus. bCommError	БИТ	В ранее завершенной связи произошла ошибка. Проверьте код ошибки.
%QX(n).6	byStatus. bCommAborted	БИТ	Ранее прерванная связь была прервана из-за разрыва соединения.
%QX(n).7	byStatus. bDiag_7_reserved	БИТ	Зарезервировано.
Последний код ошибки (включается, если bCommError = true):			
%QB(n+1)	eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Информирует о возможной причине последней ошибки в отображении MODBUS. Более подробная информация приведена в таблице 114.
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_ВЫКЛЮЧЕНО (129)* IGNORE_FRAME (255)*
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_MAPPING_1.*	Размер	Описание
Статистика коммуникаций:			

5. КОНФИГУРАЦИЯ

%QB(n+3)	byDiag_3_reserved	БАЙТ	Зарезервировано.
%QW(n+4)	wCommCounter	СЛОВО	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	СЛОВО	Счетчик связи завершен с ошибками. При достижении значения 65535 счетчик возвращается к нулю.

Таблица 113: Диагностика клиентских отношений MODBUS

Примечания:

Коды исключений: коды исключений, показанные в этом файле, представляют собой возвращаемые сервером значения. Определения кодов исключений 128, 129 и 255 действительны только для ведомых устройств Altus. Для ведомых других производителей эти коды исключений могут иметь другое значение.

Disabling Variable: 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.

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

Код	Перечислимые	Описание
1	ERR_EXCEPTION	Ответ находится в коде исключения (см. eLastExceptionCode = Exception Code).
2	ERR_CRC	Ответ с недопустимым CRC.
3	ERR_ADDRESS	Адрес MODBUS не найден. Адрес, по которому был получен ответ 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 отличается от ожидаемого.
7	ERR_NOT_ECHO	Ответ не является эхом запроса (FC 05 и 06).
8	ERR_REFERENCE_NUMBER	Неверный номер ссылки (FC 15 и 16).
9	ERR_INVALID_FRAME_SIZE	Ответ короче, чем ожидалось.
20	ERR_CONNECTION	Ошибка при установлении соединения.
21	ERR_SEND	Ошибка на этапе передачи.
22	ERR_RECEIVE	Ошибка на этапе приема.
40	ERR_CONNECTION_TIMEOUT	Тайм-аут на уровне приложения во время соединения.
41	ERR_SEND_TIMEOUT	Тайм-аут на уровне приложения во время передачи.
42	ERR_RECEIVE_TIMEOUT	Тайм-аут на уровне приложения при ожидании ответа.
43	ERR_CTS_OFF_TIMEOUT	Тайм-аут при ожидании CTS = false при передаче.

5. КОНФИГУРАЦИЯ

44	ERR_CTS_ON_TIMEOUT	Тайм-аут при ожидании CTS = true при передаче.
128	NO_ERROR	Ошибок с момента запуска нет.

Таблица 114: Коды ошибок связи MODBUS

ВНИМАНИЕ

В отличие от других задач приложения, при достижении метки при отладке MainTask задача экземпляра MODBUS Ethernet Client или любая другая задача MODBUS прекращает выполнение 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.

5.5.8.2. Конфигурация клиента MODBUS Ethernet через прямое представление (%Q)

- Для конфигурирования данного протокола с использованием прямого представления (%Q) необходимо выполнить следующие действия:
- Настройка общих параметров протокола MODBUS, таких как: время обмена данными и переменные прямого представления (%Q) для получения диагностики.
- Добавление и конфигурирование устройств путем задания адреса, переменных прямого представления (%Q) для отключения связей, тайм-аутов связи и т.д.
- Добавление и конфигурирование связей MODBUS с указанием типа данных и функции MODBUS, тайм-аутов, переменных прямого представления (%Q) для получения диагностики связи и других для приема/записи данных, количества передаваемых данных и опроса связи.

Описание каждой конфигурации приведено ниже в этом разделе.

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

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

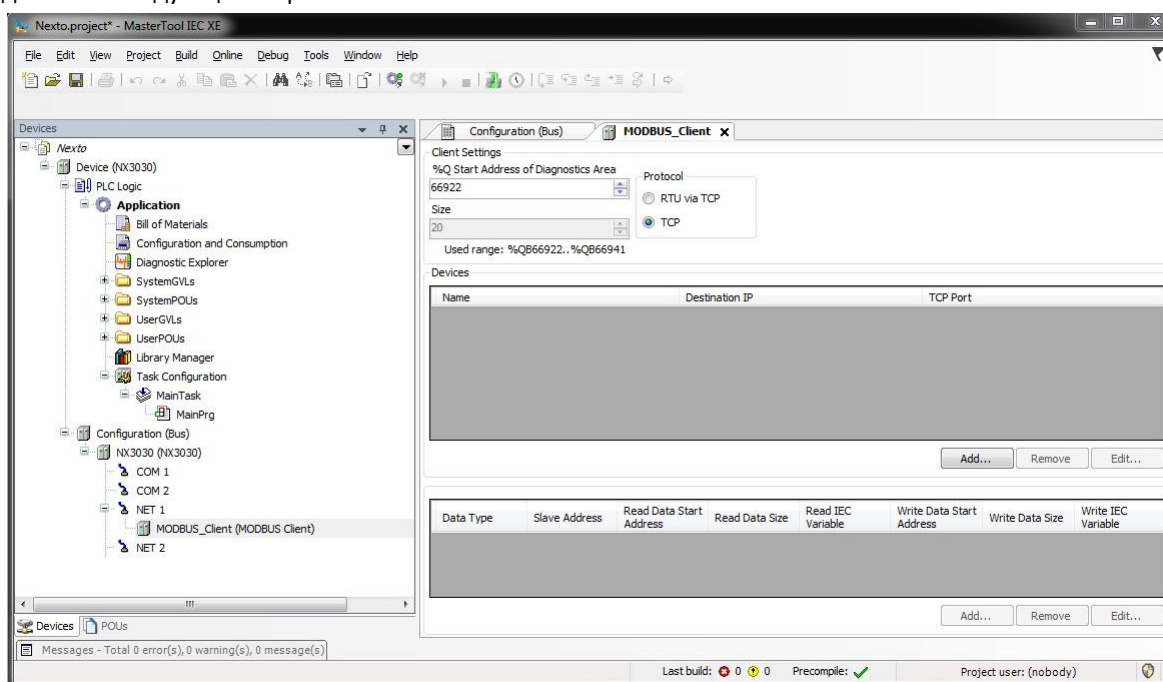


Рисунок 75: Экран настройки клиента MODBUS

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

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

Table 115: MODBUS Client настройки

Примечания:

%Q Начальный адрес области диагностики: данное поле ограничено размером адресуемой памяти выходных переменных (%Q) на процессоре, который можно найти в разделе [Память](#).

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

Описание диагностики и команд MODBUS приведено в табл. [106](#).

5.5.8.2.2. Конфигурирование устройства - Конфигурирование через прямое представление (%Q)

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

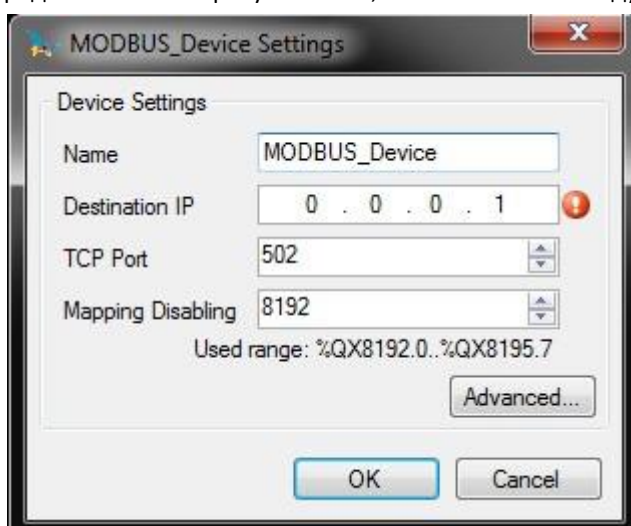


Рисунок 76: Конфигурация MODBUS Client

Конфигурация	Описание	Заводское значение	Опции
--------------	----------	--------------------	-------

5. КОНФИГУРАЦИЯ

Имя	Имя экземпляра	MODBUS_Device	Идентификатор, в соответствии с МЭК 61131-3
IP-адрес назначения	IP-адрес сервера	0. 0. 0.1	1.0.0.1 до 223.255.255.255
TCP Порт	TCP-порт	502	2 до 65534
Сопоставление Отключение	Начальный адрес, используемый для отключения связи с MODBUS	-	Любой адрес области %Q, ограниченный моделью процессора

Таблица 116: Конфигурация клиентских устройств

Примечания:

Название экземпляра: это поле является идентификатором устройства, который проверяется в соответствии с МЭК 61131-3, т.е. не допускает пробелов, специальных символов и начинающихся с цифр. Он ограничен 24 символами.

TCP-порт: если в один Ethernet-интерфейс добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные TCP-порты. Некоторые TCP-порты, среди перечисленных выше, зарезервированы и поэтому не могут быть использованы. См. таблицу Зарезервированные TCP/UDP-порты.

Mapping Disabling: состоит из 32 битов и используется для индивидуального отключения 32 связей MODBUS, сконфигурированных в пространстве Device Mappings. Отношение отключается, если соответствующий бит равен 1, в противном случае отображение включено. Данное поле ограничено размером адресуемой памяти выходных переменных (%Q) процессора, который можно найти в разделе Память.

Значение по умолчанию: заводское значение для поля Mapping Disabling не может быть установлено, так как создание экземпляра протокола может быть произведено в любой момент разработки приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), которое пока не используется.

Тайм-аут связи: настройки, присутствующие на кнопке Advanced... на TCP-соединении, описаны в примечаниях к разделу Конфигурация устройства - Конфигурация через символическое отображение.

5.5.8.2.3. Конфигурация отображения - Конфигурация через прямое представление (%Q)

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

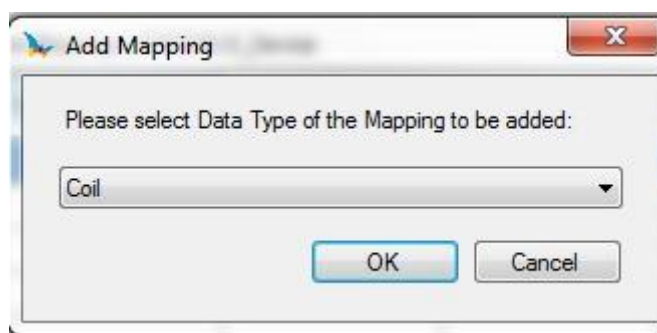


Рисунок 77: Тип данных MODBUS

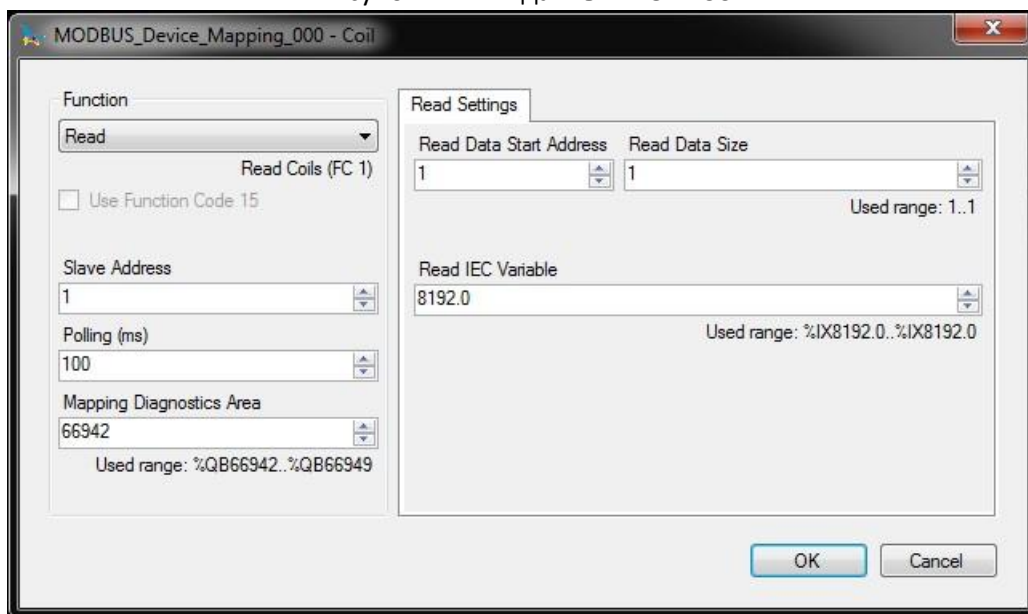


Рисунок 78: Функция MODBUS

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

Конфигурация	Описание	Значение по умолчанию	Опции
Функция	Тип функции MODBUS	Чтение	Чтение Запись Чтение/запись Маска Запись
Адрес ведомого	Адрес ведомого устройства MODBUS	1	0 до 255
Опрос (мс)	Период связи (мс)	100	0 до 3600000
Область диагностики	Начальный адрес диагностики интерфейса MODBUS	-	0 до 2147483640
Адрес начала чтения данных	Начальный адрес считывания данных MODBUS	1	1 до 65536
Размер считываемых данных	Количество прочитанных данных MODBUS	-	Зависит от используемой функции
Чтение переменной IEC	Начальный адрес считываемых переменных (%)	-	0 до 2147483647

5. КОНФИГУРАЦИЯ

Запись данных Начальный адрес	Стартовый адрес записи данных MODBUS	1	1 до 65536
Запись данных Размер данных	Количество записываемых данных MODBUS	-	Зависит от используемой функции
Запись IEC-переменной	Начальный адрес переменных записи (%Q)	-	0 до 2147483647
Маска Запись переменных IEC	Начальный адрес переменных для маски записи (%Q)	-	0 до 2147483644

Таблица 117: Сопоставление устройств

Примечания:

Таблица сопоставления устройств: количество настроек и значений, описанных в колонке Options, может меняться в зависимости от типа данных и функции MODBUS.

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

Запрос: этот параметр указывает, как часто должна выполняться связь, установленная для данного отношения. По окончании связи будет ожидаться время, равное настроенному опросу, после чего будет выполнена новая связь как можно быстрее.

Составление карты диагностической области: это поле ограничено размером адресуемой памяти выходных переменных (%Q) процессора, который можно найти в разделе [Память](#). Конфигурируемые диагностики связей MODBUS описаны в табл. 91.

Размер данных для чтения и записи: подробности о размере данных, поддерживаемых каждой функцией, описаны в примечаниях раздела [Конфигурация запросов - Настройки символьного отображения](#).

ВНИМАНИЕ

При обращении к памяти коммуникационных данных между устройствами с различным порядком (прямой и обратный порядок) может произойти инверсия данных при чтении/записи. В этом случае пользователь должен скорректировать данные в приложении.

Чтение переменной IEC: если тип данных MODBUS - Coil или Input Status (bit), то начальный адрес переменных IEC будет в формате %IX10.1. Если же тип данных MODBUS - Holding Register или Input Register (16 бит), то начальный адрес переменных IEC будет в формате %IW. Это поле ограничено объемом памяти адресуемых входных переменных (%I) от каждого процессора, который можно посмотреть в разделе [Память](#).

Запись IEC-переменной: если тип данных MODBUS - Coil (bit), то начальный адрес IEC-переменной будет в формате %QX10.1. Если же тип данных MODBUS - Holding Register (16 бит), то начальный адрес переменных МЭК будет в формате %QW. Это поле ограничено объемом памяти адресуемых выходных переменных (%Q) от каждого процессора, который можно посмотреть в разделе [Память](#).

Запись маски IEC-переменных: функция Mask Write Register (FC 22) использует логику между уже записанным значением и двумя словами, которые конфигурируются в этом поле, используя %QW(0) для маски AND и %QW(2) для маски OR; что позволяет пользователю обрабатывать слово. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе [Память](#).

Значение по умолчанию: заводское значение по умолчанию не может быть установлено для полей Mapping Diagnostics Area, Read IEC Variable, Write IEC Variable и Mask Write IEC Variables, 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 Read/Write Data Size нельзя установить заводское значение, так как они будут меняться в зависимости от выбранного типа данных MODBUS.

ВНИМАНИЕ

В отличие от других задач приложения, при достижении метки при отладке MainTask задача экземпляра MODBUS Ethernet Client или любая другая задача MODBUS прекращает выполнение 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.

5.5.8.3. Начало клиентской связи MODBUS в ациклической форме

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

- Определите максимальное время опроса для отношений;
- Держать отношение в отключенном состоянии;
- Включить отношение в тот момент, когда требуется его выполнение;
- Дождаться подтверждения окончания выполнения отношения и в этот момент снова отключить его.

5.5.9. MODBUS Ethernet сервер

Этот протокол доступен для всех ЦПУ серии Nexto на его Ethernet-каналах. При выборе этой опции в MasterTool IEC XE центральный процессор становится сервером связи MODBUS, что позволяет устанавливать связь с клиентскими устройствами MODBUS. Этот протокол доступен только в том случае, если центральный процессор находится в режиме исполнения (Run Mode).

Существует два способа конфигурирования этого протокола. Первый - через прямое представление (%Q), при котором переменные определяются своим адресом. Второй - через символическое отображение, при котором переменные определяются по имени.

Процедура вставки экземпляра протокола подробно описана в руководстве пользователя MasterTool IEC XE - MU299609.

5.5.9.1. Конфигурация протокола Ethernet сервера MODBUS для символьного отображения

Для конфигурирования этого протокола с использованием символьных отображений необходимо выполнить следующие действия:

- Настройте общие параметры протокола сервера MODBUS, такие как: TCP-порт, выбор протокола, IP-фильтры для чтения и записи (доступны по кнопке Конфигурация фильтров) и время обмена данными (доступны по кнопке Расширенные конфигурации сервера).
- Добавление и настройка сопоставлений MODBUS с указанием имени переменной, типа данных, начального адреса и размера данных. Описание каждой конфигурации приведено далее в этом разделе.

5.5.9.1.1. Общие параметры протокола сервера MODBUS - конфигурирование через символическое отображение

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

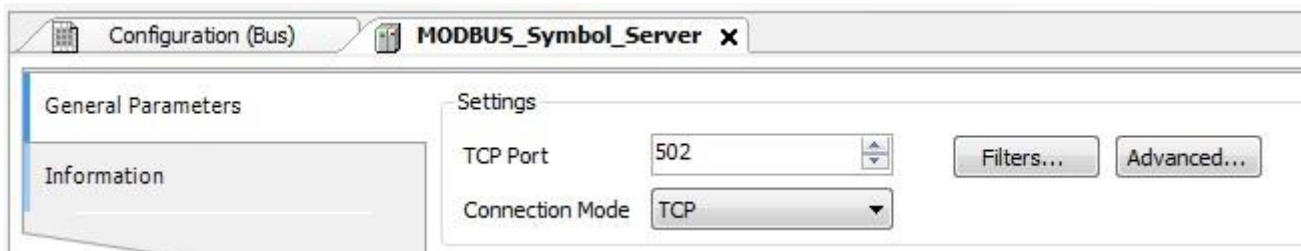


Рисунок 79: Экран конфигурации общих параметров сервера MODBUS

5. КОНФИГУРАЦИЯ

Конфигурация	Описание	По умолчанию	Опции
ТСР порт	ТСР порт	502	2 до 65534
Режим подключения	Выбор протокола	ТСР	RTU через ТСР ТСР

Таблица 118: Общие конфигурации сервера MODBUS

Примечания:

Порт ТСР: если в один интерфейс Ethernet добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные порты ТСР. Некоторые ТСР-порты, среди перечисленных выше, зарезервированы и поэтому не могут быть использованы. См. таблицу [Зарезервированные ТСР/UDP-порты](#).

Настройки, присутствующие в кнопке Фильтры..., описанные в таблице ниже, относятся к фильтрам ТСР-коммуникаций:

Конфигурация	Описание	Значения по умолчанию	Опции
IP-адрес фильтра записи	Задаёт диапазон IP-адресов с доступом на запись в переменные, объявленные в интерфейсе MODBUS.	0.0.0.0	0.0.0.0 до 255.255.255.255
Маска фильтра записи	Указание маски подсети в сочетании с параметром IP-фильтра для записи.	0.0.0.0	0.0.0.0 до 255.255.255.255
IP-адрес фильтра чтения	Указание диапазона IP-адресов с доступом на чтение в переменных, объявленных в интерфейсе MODBUS.	0.0.0.0	0.0.0.0 до 255.255.255.255
Маска фильтра чтения	Указывается маска подсети в сочетании с параметром IP-фильтра для чтения.	0.0.0.0	0.0.0.0 до 255.255.255.255

Таблица 119: IP Фильтры

Примечание:

Фильтры: фильтры используются для установления диапазона IP-адресов, которые имеют доступ на запись или чтение к MODBUS-отношениям, будучи индивидуально настроенными. Критерии разрешения выполняются с помощью операции логического И 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 Read Filter для определения прав на чтение.

Время обмена данными по серверному протоколу MODBUS, находящееся на кнопке Advanced... экрана конфигурации, делится на: Цикл задачи и Тайм-аут бездействия соединения.

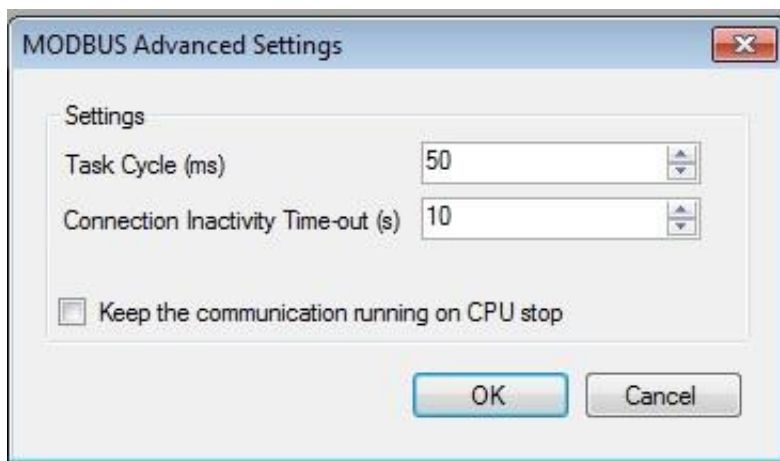


Рисунок 80: Экран конфигурации дополнительных настроек сервера MODBUS

Конфигурация	Описание	Значения по умолчанию	Опции
Цикл задачи (мс)	Время выполнения экземпляра в рамках цикла, без учета собственного времени выполнения	50	5 до 100
Тайм-аут бездействия соединения (с)	Максимальное время простоя между клиентом и сервером до закрытия соединения сервером	10	1 до 3600
Поддерживать связь при остановке процессора.	Разрешить ведомому устройству MODBUS Symbol Slave работать, пока процессор находится в состоянии STOP или после точки останова	немаркированные	Маркированные или немаркированные

Таблица 120: Расширенные конфигурации сервера MODBUS

Примечания:

Цикл задачи: пользователь должен быть осторожен при изменении этого параметра, так как он напрямую влияет на время ответа, объем данных для сканирования и, главным образом, на баланс ресурсов процессора между коммуникациями и другими задачами.

Тайм-аут неактивности соединения: этот параметр был создан для того, чтобы избежать достижения максимального количества TCP-соединений, представляя, что неактивные соединения остаются открытыми из-за 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.

5.5.9.1.2. Диагностика сервера MODBUS - Конфигурирование через символическое отображение

Диагностика и команды протокола сервера MODBUS, сконфигурированные либо по символическому отображению, либо по прямому представлению, хранятся в переменных типа T_DIAG_MODBUS_ETH_SERVER_1, а отображение по

5. КОНФИГУРАЦИЯ

прямому представлению - в 4- и 8-байтовых, которые описаны в таблице ниже (n - значение, сконфигурированное в поле %Q Начальный адрес области диагностики):

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_SERVER_1.*	Размер	Описание
Диагностические биты:			
%QX(n).0	tDiag. bRunning	БИТ	Сервер работает.
%QX(n).1	tDiag. bNotRunning	БИТ	Сервер не работает (см. бит bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	БИТ	Бит bNotRunning был включен, так как работа сервера была прервана пользователем через бит команды.
%QX(n).3	tDiag. bConfigFailure	БИТ	Прерванная диагностика.
%QX(n).4	tDiag. bRXFailure	БИТ	Прерванная диагностика.
%QX(n).5	tDiag. bTXFailure	БИТ	Прекращенная диагностика.
%QX(n).6	tDiag. bModuleFailure	БИТ	Прекращенная диагностика.
%QX(n).7	tDiag. bDiag_7_reserved	БИТ	Зарезервировано.
%QB(n+1)	byDiag_1_reserved	БАЙТ	Зарезервировано.
Биты команд, перезапускаются автоматически:			
%QX(n+2).0	tCommand. bStop	БИТ	Остановите сервер.
%QX(n+2).1	tCommand. bRestart	БИТ	Перезапустить сервер.
%QX(n+2).2	tCommand. bResetCounter	БИТ	Сбросить статистику диагностики (счетчики).
%QX(n+2).3	tCommand. bDiag_19_reserved	БИТ	Зарезервировано.
%QX(n+2).4	tCommand. bDiag_20_reserved	БИТ	Зарезервировано.
%QX(n+2).5	tCommand. bDiag_21_reserved	БИТ	Зарезервировано.
%QX(n+2).6	tCommand. bDiag_22_reserved	БИТ	Зарезервировано.

5. КОНФИГУРАЦИЯ

%QX(n+2).7	tCommand. bDiag_23_reserved	БИТ	Зарезервировано.
%QB(n+3)	byDiag_3_reserved	БАЙТ	Зарезервировано.
Статистика коммуникаций:			
%QW(n+4)	tStat. wActiveConnections	СЛОВО	Количество установленных соединений между клиентом и сервером (от 0 до 64).
%QW(n+6)	tStat. wTimeoutClosedConnections	СЛОВО	Счетчик соединений между клиентом и сервером, прерванных после периода бездействия - тайм-аут (от 0 до 65535).
%QW(n+8)	tStat. wClientClosedConnections	СЛОВО	Счетчик соединений, прерванных по запросу клиента (от 0 до 65535).
Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_SERVER_1.*	Размер	Описание
%QW(n+10)	tStat. wRXFrames	СЛОВО	Счетчик кадров Ethernet, полученных сервером. Один Ethernet-кадр может содержать более одного запроса (от 0 до 65535).
%QW(n+12)	tStat. wRXRequests	СЛОВО	Счетчик запросов, полученных сервером, и нормальных ответов на них (от 0 до 65535).
%QW(n+14)	tStat. wTXExceptionResponses	СЛОВО	Запросы, принятые сервером, учитываются и отвечают на них кодами исключений (от 0 до 65535). Коды исключений перечислены ниже: 1: код функции (FC) разрешен, но не поддерживается. 2: отношение не найдено в данных MODBUS. 3: недопустимое значение адреса. 128: ведущий/клиент не имеет права на чтение или запись. 129: отношение MODBUS отключено.
%QW(n+16)	tStat. wRXIllegalRequests	СЛОВО	Счетчик нелегальных запросов (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	СЛОВО	Зарезервировано.

Таблица 121: Диагностика сервера MODBUS

Примечание:

5. КОНФИГУРАЦИЯ

Счетчики: все счетчики MODBUS Ethernet Server Diagnostics возвращаются к нулю при превышении предельного значения 65535.

5.5.9.1.3. Конфигурация с отображением - конфигурирование с помощью символического отображения

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

Mappings

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

Рисунок 81: Экран сопоставления данных сервера MODBUS

Конфигурация	Описание	Значения по умолчанию	Опции
Значение Переменная	Символьное имя переменной	-	Имя переменной, объявленной в программе или GVL
Тип данных	Тип данных MODBUS	-	Катушка Состояние входа Регистр удержания Входной регистр
Конфигурация	Описание	Значения по умолчанию	Опции
Адрес начала данных	Начальный адрес данных MODBUS	-	1 до 65536
Абсолютный адрес начала данных	Начальный адрес абсолютных данных Modbus как их тип	-	-
Размер данных	Размер данных MODBUS	-	1 до 65536
Диапазон данных	Настроенный адрес диапазона данных	-	-

Таблица 122: Конфигурация сопоставлений MODBUS Ethernet

Примечания:

Значение переменной: это поле используется для указания символической переменной в связи MODBUS.

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

Начальный адрес данных: начальный адрес данных в отношении MODBUS.

Абсолютный начальный адрес данных: абсолютный начальный адрес данных MODBUS в соответствии с их типом. Например, регистр удержания с адресом 5 имеет абсолютный адрес 400005. Это поле доступно только для чтения и служит для помощи в конфигурации MODBUS клиента/мастера, 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 задает максимальный объем данных, к которому может обратиться MODBUS-отношение с начального адреса. Таким образом, для чтения непрерывного диапазона адресов

5. КОНФИГУРАЦИЯ

необходимо, чтобы все адреса были объявлены в одном отношении. Это поле изменяется в зависимости от конфигурируемого типа данных MODBUS.

Диапазон данных: является полем, доступным только для чтения, и сообщает о диапазоне адресов, который используется данным отображением. Он образуется суммой полей Начальный адрес данных и Размер данных. Не допускается перекрытие диапазона другими отображениями с тем же типом данных.

ВНИМАНИЕ

В отличие от других задач приложения, при достижении метки при отладке MainTask задача экземпляра MODBUS Ethernet Server или любая другая задача MODBUS прекратит выполнение 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.

5.5.9.2. Конфигурирование протокола Ethernet сервера MODBUS через прямое представление (%Q)

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

- Настройка общих параметров протокола MODBUS Server Protocol, таких как: время обмена данными, адрес и переменные прямого представления (%Q) для приема диагностических и управляющих сообщений.
- Добавление и настройка отношений MODBUS с указанием типа данных MODBUS, переменных прямого представления (%Q) для приема/записи данных и количества передаваемых данных.

Описание каждой конфигурации приведено ниже в этом разделе.

5.5.9.2.1. Общие параметры протокола MODBUS Server - конфигурирование через прямое представление (%Q)

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

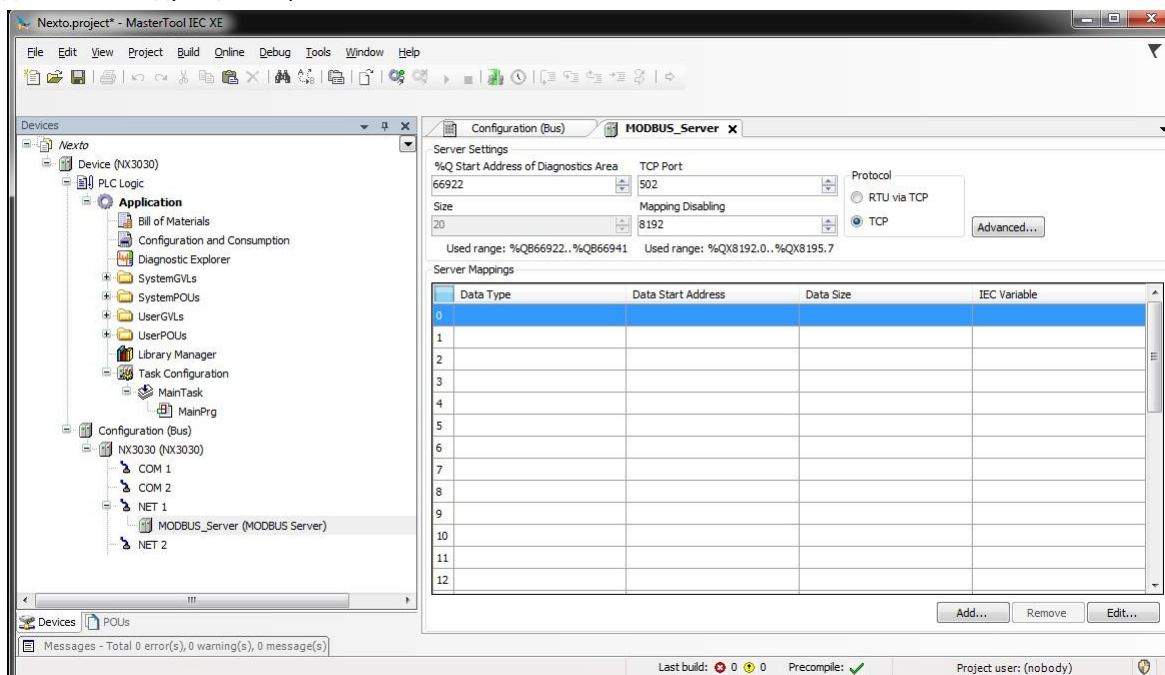


Рисунок 82: Экран настройки сервера MODBUS

5. КОНФИГУРАЦИЯ

ТСР-порт, протокол и переменные прямого представления (%Q) для управления связями и диагностики:

Конфигурация	Описание	Значения по умолчанию	Опции
%Q Начальный адрес области диагностики	Начальный адрес диагностических переменных	-	0 до 2147483628
Размер	Размер диагностических переменных	20	Отключено для редактирования
Порт ТСР	ТСР-порт	502	2 до 65534
Отображение отключения	Начальный адрес, используемый для отключения связи с MODBUS	-	0 до 2147483644
Протокол	Выбор протокола	TCP	RTU через TCP TCP

Таблица 123: Настройки для управления отношениями и диагностикой

Примечания:

%Q Начальный адрес области диагностики: это поле ограничено размером адресуемой памяти выходных переменных (%Q) на процессоре, который можно найти в разделе [Память](#).

ТСР-порт: если в один интерфейс Ethernet добавлено несколько экземпляров протокола, то для каждого экземпляра должны быть выбраны разные ТСР-порты. Некоторые ТСР-порты, среди перечисленных выше, зарезервированы и поэтому не могут быть использованы. См. таблицу [Зарезервированные ТСР/UDP-порты](#).

Отображение отключения: состоит из 32 битов и используется для отключения по отдельности 32 отношений MODBUS, сконфигурированных в пространстве сопоставления серверов. Отношение отключается, если соответствующий бит равен 1, в противном случае отображение включено. Данное поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе [Память](#).

Значение по умолчанию: заводское значение по умолчанию для полей %Q Начальный адрес области диагностики и Отключение отображения не может быть установлено, поскольку создание экземпляра протокола может быть проведено в любой момент при разработке приложения. Программа MasterTool IEC XE сама выделяет значение из диапазона выходных переменных прямого представления (%Q), которое еще не используется.

Время обмена данными по протоколу MODBUS Server, находящееся на кнопке Advanced... экрана конфигурации, делится на: Task Cycle (мс) и Connection Inactivity Time-out (с). Более подробно они описаны в разделе Общие параметры протокола MODBUS Server - Конфигурация через символическое отображение.

Диагностика и команды MODBUS описаны в табл. [121](#).

5.5.9.2.2. Конфигурация отображения - Конфигурация через прямое представление (%Q)

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

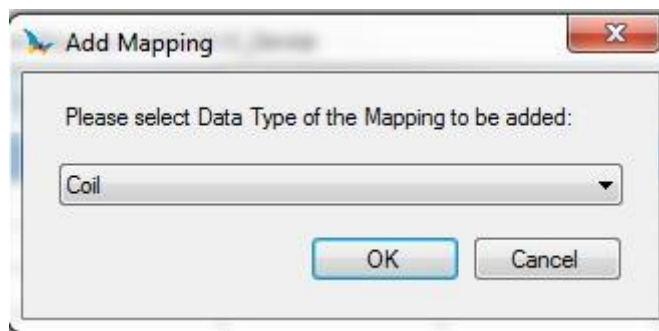


Рисунок 83: Тип данных MODBUS

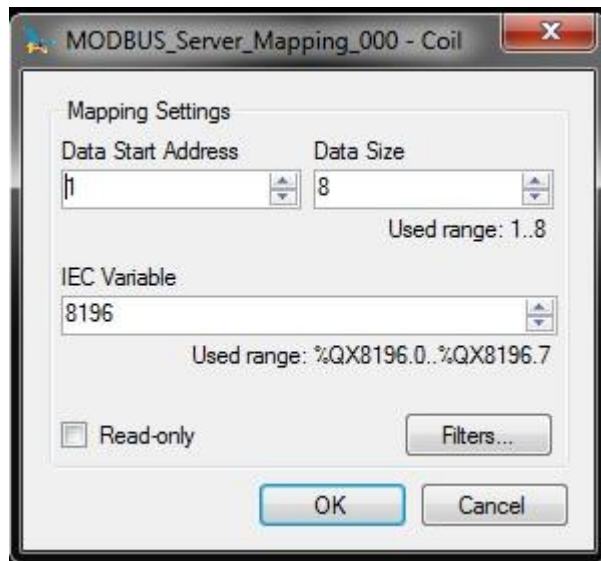


Рисунок 84: Функция сервера MODBUS

Конфигурация	Описание	По умолчанию	Опции
Тип данных	Тип данных MODBUS	Coil	Катушка (1 бит) Регистр удержания (16 бит) Состояние входа (1 бит) Входной регистр (16 бит)
Адрес начала данных	Начальный адрес данных MODBUS	1	1 до 65536
Размер данных	Количество данных MODBUS	8	От 1 до 65536 (регистр удержания и входной регистр) От 8 до 65536 (состояние катушки и входа)
Переменная IEC	Начальный адрес переменных (%Q)	-	0 до 2147483647
Только для чтения	Разрешить только чтение	отключено	Включено или отключено

Таблица 124: Сопоставление серверов

Примечания:

Опции: значения, записанные в столбце Опции, могут меняться в зависимости от настроенных данных MODBUS.

Размер данных: значение Размер данных задает максимальный объем данных, к которому может обращаться отношение MODBUS с начального адреса. Таким образом, для чтения непрерывного диапазона адресов необходимо, чтобы все адреса были объявлены в одном отношении. Это поле изменяется в зависимости от установленного типа данных MODBUS, т.е. при выборе Coil или Input Status поле Data Size должно быть числом, кратным 8. Важно также следить за тем, чтобы максимальное значение не превышало размер адресуемой выходной памяти, а присписываемые значения не совпадали с уже использованными в процессе применения.

ВНИМАНИЕ

При обращении к памяти коммуникационных данных между устройствами с различным порядком (Little-Endian и Big-Endian (прямой и обратный порядок)) может произойти инверсия данных при чтении/записи. В этом случае пользователь должен скорректировать данные в приложении.

Переменная IEC: если тип данных MODBUS - Coil или Input Status (бит), то начальный адрес переменных IEC будет иметь формат, например, %QX10.1. Если же тип данных MODBUS - регистр удержания или входной регистр (16 бит), то начальный адрес переменных МЭК будет иметь формат %QW. Это поле ограничено объемом памяти адресуемых выходных переменных (%Q) от каждого процессора, который можно посмотреть в разделе [Память](#).

Только для чтения: если эта опция включена, то она позволяет ведущему устройству связи только читать данные переменных. Запись не разрешается. Эта опция действительна только для функций записи.

По умолчанию: для поля IEC Variable нельзя задать значение по умолчанию, так как создание экземпляра протокола может быть произведено в любой момент разработки приложения, в результате чего программа MasterTool IEC XE сама выделит значение из диапазона выходных переменных прямого представления (%Q), которые еще не используются. Для поля Data Size нельзя задать значение по умолчанию, поскольку оно будет зависеть от выбранного типа данных MODBUS.

Настройки, присутствующие в кнопке Filters..., описанные в таблице ниже, относятся к фильтрам TCP-коммуникаций:

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

Таблица 125: IP Фильтры

Примечание:

Фильтры: фильтры используются для установления диапазона IP-адресов, которые имеют доступ на запись или чтение к MODBUS-отношениям, будучи индивидуально настроенными. Критерии разрешения выполняются с помощью операции логического И 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 и в параметрах фильтра чтения для определения прав на чтение.

В ранее определенных отношениях максимальный размер данных MODBUS может составлять 65536 (максимальное значение, заданное в поле "Размер данных"). Однако запрос, поступающий в MODBUS Ethernet Server,

5. КОНФИГУРАЦИЯ

должен адресоваться к подгруппе этого отображения, и эта группа должна иметь, максимум, размер данных, зависящий от кода функции, который определен ниже:

- Считывание катушек (FC 1): 2000
- Чтение состояния входа (FC 2): 2000
- Чтение регистров удержания (FC 3): 125
- Чтение входных регистров (FC 4): 125
- Запись одиночной катушки (FC 5): 1
- Запись одиночного регистра удержания (FC 6): 1
- Принудительная запись нескольких катушек (FC 15): 1968
- Запись регистров удержания (FC 16): 123
- Регистр записи маски (FC 22): 1
 - Чтение/запись Регистры удержания (FC 23):
 - Чтение: 121
 - Запись: 121

ВНИМАНИЕ

В отличие от других прикладных задач, при достижении метки разрядки в MainTask задача экземпляра Ethernet MODBUS Server и любая другая MODBUS-задача прекращает работу в тот момент, когда она пытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных в областях памяти, пока основная задача не запущена.

5.5.10. OPC DA Сервер

Связь с процессорами серии Nexto можно осуществлять с помощью технологии OPC DA ("Открытая платформа связи с доступом к данным"). Эта открытая коммуникационная платформа была разработана как стандарт в области промышленных коммуникаций. Основанная 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 принтер. При правильном подключении компьютеру требуется драйвер для взаимодействия с оборудованием. Точно так же OPC помогает обеспечить взаимодействие между системой контроля и полевыми данными на ПЛК.

При разработке проекта настроить связь и обмен информацией между системами чрезвычайно просто с помощью технологии OPC DA. При использовании других драйверов, основанных на адресах, необходимо создавать таблицы, связывающие теги из системы наблюдения с переменными из программируемого контроллера. При изменении областей данных в ходе проекта необходимо перемапировать переменные и создать новые таблицы, содержащие связи между информацией на ПЛК и системой диспетчерского контроля и сбора данных. (SCADA).

5. КОНФИГУРАЦИЯ

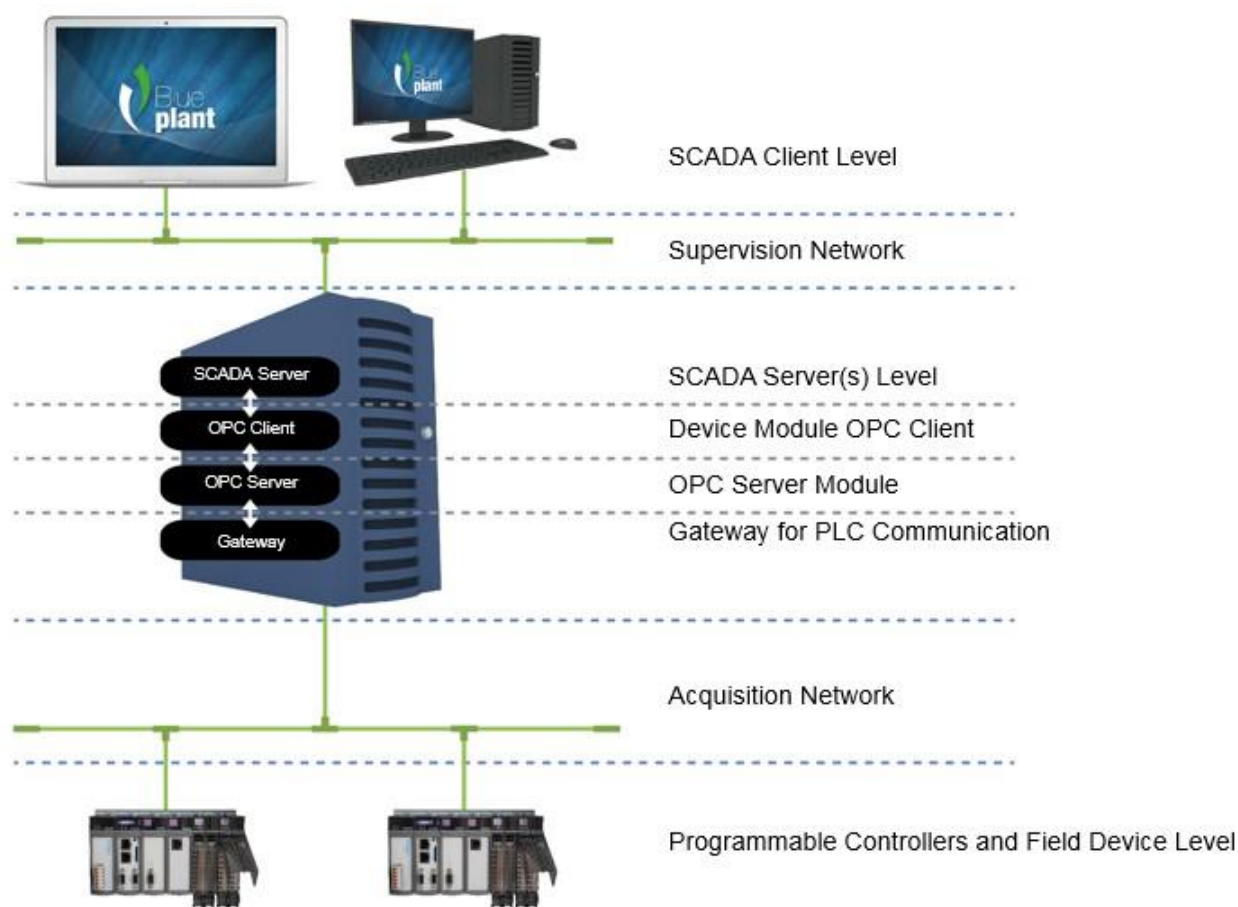


Рисунок 85: Архитектура OPC DA

На рисунке показана архитектура связи SCADA-системы и ПЛК в проектах автоматизации. Все роли, присутствующие в коммуникации, явно выражены на этом 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.

Роль	Описание
Уровень программируемых контроллеров и полевых устройств	В полевых устройствах и ПЛК хранится информация о состоянии работы и управлении установкой. Система SCADA получает доступ к информации на этих устройствах и хранит ее на сервере SCADA, чтобы клиенты SCADA могли обращаться к ней во время работы установки.
Сеть сбора данных	Сеть сбора данных - это место, через которое проходят запросы на данные, собранные полевыми устройствами, чтобы запросить данные, собранные полевыми устройствами.
Шлюз для связи с ПЛК	Шлюз обеспечивает связь между OPC DA Сервер 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.

5. КОНФИГУРАЦИЯ

Модуль OPC-сервера	OPC DA Сервер - это модуль, отвечающий за прием запросов OPC DA и их преобразование в обмен данными с полевыми устройствами.
Роль	Описание
Модуль устройства OPC-клиент	Модуль OPC Client Device отвечает за запросы к OPC DA Серверу по протоколу OPC DA. Собранные данные хранятся в базе данных SCADA Server.
Уровень сервера SCADA	SCADA-сервер отвечает за подключение к различным коммуникационным устройствам и хранение собранных ими данных в базе данных, чтобы к ним могли обращаться SCADA-клиенты.
Сеть наблюдения	Сеть наблюдения - это сеть, через которую клиенты SCADA подключаются к серверам SCADA. В топологии, где нет нескольких Клиентов или Сервер и Клиент установлены на одном и том же оборудовании, такой сети не существует.
Уровень клиента SCADA	Клиенты SCADA отвечают за запрос к серверам SCADA необходимых данных для отображения их на экране, где происходит работа установки. Через них можно выполнять считывания и записи по данным, хранящимся в базе данных SCADA-сервера.

Таблица 126: Описание ролей в архитектуре сервера OPC DA

Связь между метками в системе контроля и данными процесса в переменных контроллера абсолютно прозрачна. Это означает, что при изменении областей данных в ходе разработки проекта нет необходимости переделывать связи между информацией на ПЛК и SCADA, достаточно использовать новую переменную, предоставленную ПЛК, в системах, запрашивающих эти данные.

Использование OPC обеспечивает большую производительность и связь со SCADA-системами. Это способствует сокращению времени разработки приложений и расходов на обслуживание. Благодаря тому, что это открытый стандарт, он позволяет вводить новые данные в коммуникацию в упрощенной форме и с большей гибкостью и функциональной совместимостью между системами автоматизации.

Установка 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 Ethernet процессоров Nexto. Модули расширения Ethernet не поддерживают данную функциональность.

5.5.10.1. Создание проекта для связи с OPC DA

В отличие от связи с такими драйверами, как MODBUS и PROFIBUS DP, для настройки OPC-связи необходимо лишь правильно задать узел и указать, какие переменные будут использоваться в связи. Существует два способа указать, какие переменные проекта будут доступны в OPC DA Сервере. В обоих случаях необходимо добавить в приложение объект "Конфигурация символа", если его нет. Чтобы добавить его, щелкните правой кнопкой мыши на объекте Application и выберите опцию.

ВНИМАНИЕ

Переменные, указанные в объектах IoConfig_Globals, IoConfig_Application_Mappings и IoConfig_Global_Mappings, используются внутри системы для управления вводом/выводом и не должны использоваться пользователем.

ВНИМАНИЕ

Помимо переменных, объявленных в POU языка 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 нормой IEC 61131-1. Для каждого перехода создается переменная типа BOOL, которая определяет, является ли переход истинным или ложным. Эти переменные отображаются в объекте Конфигурация символов, к которому может быть предоставлен доступ OPC-клиенту.

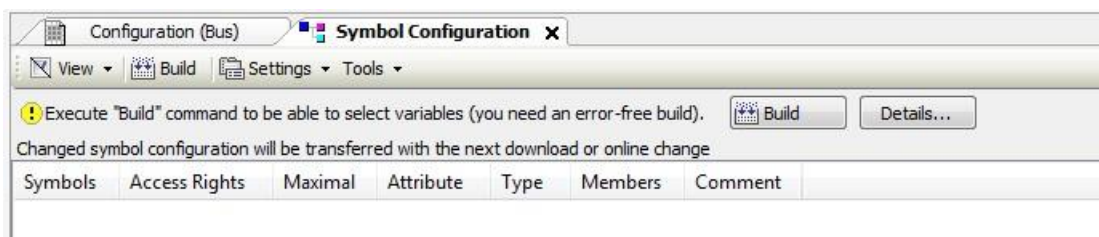





Рисунок 86: Объект конфигурации символа

В таблице ниже приведены описания полей экрана объекта "Конфигурация символа"..

Поле	Описание
Символы	Идентификатор переменной, который будет предоставлен серверу OPC DA Server.
Права доступа	Указывает возможный уровень прав доступа в объявленном символе. Если этот 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  Чтение 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 или отсутствие атрибута 'символ'




Атрибут	<p>Указывает, используется ли атрибут 'symbol' при объявлении переменной. Если атрибут не используется, то этот столбец остается пустым. Для случаев, когда атрибут используется, поведение будет следующим:</p> <p>атрибут 'символ' := 'чтение' столбец показывает </p> <p>атрибут 'символ' := 'запись' столбец показывает </p> <p>атрибут 'символ' := 'чтение-запись' столбец показывает </p>
Тип	Тип данных объявленной переменной.
Члены	Если типом данных является структура, то в этом столбце включается кнопка. Щелчок на кнопке позволяет выбрать, какие элементы этой структуры будут предоставлены серверу OPC DA Server.
Комментарий	Комментарий к переменной, вставляемый в POU или 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 колонке комментариев в табличном режиме.

Таблица 127: Описание полей экрана конфигурации объекта символа

При изменении настроек проекта, например, добавлении или удалении переменных, необходимо выполнить команду Build, чтобы обновить список переменных. Эту команду необходимо выполнять до тех пор, пока не исчезнет сообщение, показанное на рис. 86. После этого все имеющиеся в проекте переменные, независимо от того, объявлены ли они на POU, GVL или в диагностике, будут показаны здесь и могут быть выбраны. Выбранные переменные будут доступны на сервере OPC DA Server для доступа клиентов.

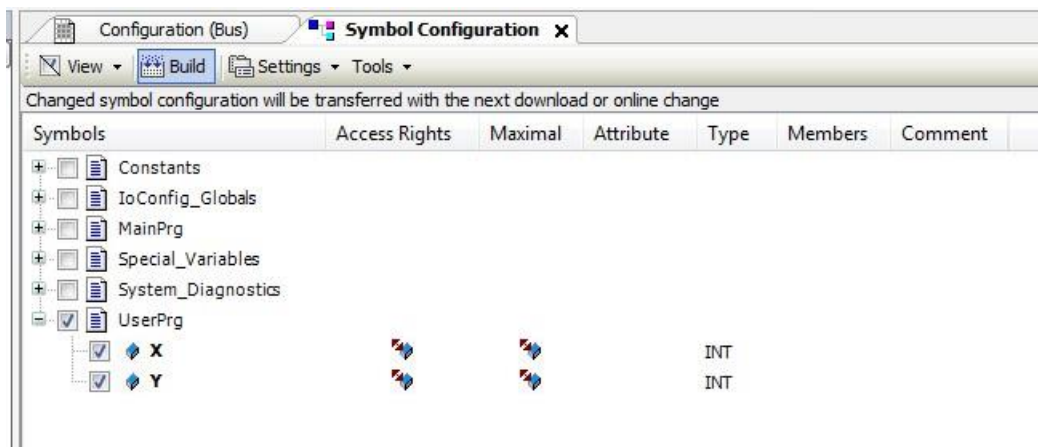


Рисунок 87: Выбор переменных в конфигурации символа

После этой процедуры проект должен быть загружен в ПЛК, чтобы переменные были доступны для связи с OPC DA Server. Если открыть экран конфигурации символов объекта и изменить любую из выбранных переменных, POU или GVL, то ее имя будет отображаться красным цветом. Это может произойти, например, при удалении переменной или изменении значения атрибута.

Также можно установить, какие переменные будут доступны на сервере OPC DA Server, с помощью атрибута, вставляемого непосредственно в POU или GVL, где объявлены переменные. При наличии атрибута "символ" на объявлении переменной, а это может быть как перед определением имени POU или GVL, так и для каждой переменной в отдельности, эти переменные отправляются непосредственно в объект "Конфигурация символов"

5. КОНФИГУРАЦИЯ

Symbol Configuration), с символом в колонке "Атрибут". В этом случае необходимо перед загрузкой проекта в процессор выполнить команду Build из объекта Symbol Configuration.

Для использования атрибута допустимы следующие синтаксисы:

- атрибут 'symbol' := 'none' - при значении атрибута 'none' переменные не будут доступны для OPC DA Server и не будут отображаться на экране конфигурации символов объекта.
- атрибут 'symbol' := 'read' - при значении атрибута 'read' переменные будут доступны OPC DA Server с правом доступа только на чтение.
- атрибут 'symbol' := 'write' - при значении атрибута 'write' переменные будут доступны OPC DA Server с правом доступа только на запись.
- атрибут 'symbol' := 'readwrite' - когда значение атрибута равно 'readwrite', переменные будут доступны OPC DA Server с правом доступа на чтение и запись.

В следующем примере объявления переменных параметры переменных А и В позволяют серверу OPC DA Server обращаться к ним с правами чтения и записи. Однако к переменной С доступ невозможен, а к переменной D можно обращаться с правами только на чтение.

```
{attribute 'symbol' := 'readwrite'} PROGRAM UserPrg
VAR
A: INT;
B: INT; {attribute 'symbol' := 'none'}
C: INT;
{attribute 'symbol' := 'read'}
D :INT;
END_VAR
```

При определении переменной с типом, отличным от базовых типов, использование атрибута должно осуществляться внутри объявления данного ИУ, а не только в контексте, в котором создается переменная. Например, в случае, когда экземпляр ИУ находится внутри ROU или GVL, имеющего атрибут, он не будет влиять на поведение элементов этого экземпляра ИУ. Необходимо будет применить тот же уровень прав доступа к объявлению ИУ.

ВНИМАНИЕ

Конфигурации символов, которые будут предоставляться OPC DA Server, хранятся внутри проекта ПЛК. При изменении этих конфигураций необходимо загрузить приложение на ПЛК, чтобы можно было получить доступ к этим переменным.

ВНИМАНИЕ

При удалении переменной из проекта и загрузке ее в ПЛК, сняв флажок с объекта "Конфигурация символов", переменная больше не может быть прочитана с помощью OPC-клиента. Если переменная снова будет добавлена в проект с тем же именем и тем же контекстом и вставлена на объект "Конфигурация символов", необходимо будет перезагрузить OPC Client для обновления адресной ссылки переменной, которая будет создана в другой области памяти ПЛК.

5.5.10.2. Конфигурирование ПЛК на сервере OPC DA

Конфигурирование ПЛК производится внутри MasterTool IEC XE с помощью опции, доступной в Online. Необходимо запустить MasterTool IEC XE от имени администратора.

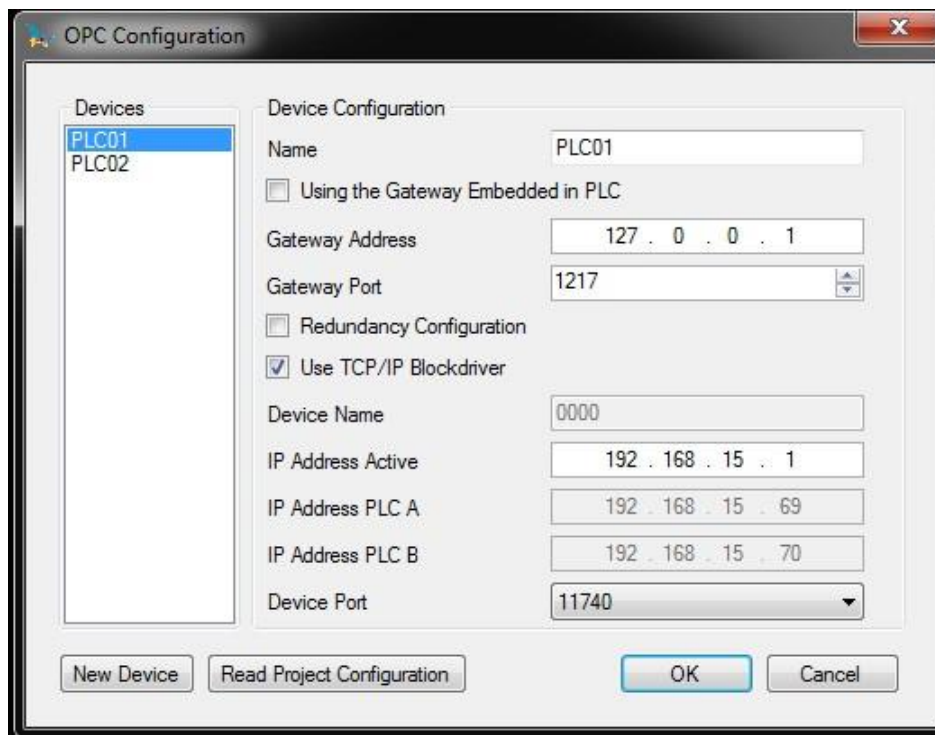


Рисунок 88: Настройки сервера OPC DA

Конфигурация шлюза задается в шлюзе, используемом для связи между MasterTool IEC XE и ПЛК и описанном в разделе Настройки связи, приведенном в руководстве пользователя MasterTool IEC XE - MU299609. Если используется конфигурация localhost, то адрес шлюза должен быть заполнен 127.0.0.1. Такая конфигурация необходима, поскольку 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, которая может быть выбрана, если необходимо использовать шлюз, находящийся в самом ПЛК. Эта опция может быть использована для оптимизации связи, так как позволяет избежать избыточного трафика через определенную станцию, когда к одному ПЛК подключено несколько станций с OPC Client.

Для конфигурирования ПЛК возможны два типа конфигурации, в зависимости от установки флажка Use TCP/IP Blockdriver. Если опция не выбрана, то поле Device Name должно быть заполнено именем ПЛК. Это имя отображается в ПЛК, выбранном в качестве активного на экране Communication Settings.

Другой вариант - использовать IP-адрес интерфейсов Ethernet. В этом поле должен быть указан тот же адрес, который был задан на экранах конфигурации. Кроме того, при использовании этого способа номер порта должен быть установлен на 11740. Подтверждение сохранит конфигурацию OPC DA Server.

Конфигурация устройства	Описание	Настройка по умолчанию	Возможности
Наименование	Описание ПЛК в конфигурационном файле OPC DA Server. Это поле может иметь любое имя, но для организационных целей рекомендуется использовать имя проекта, загруженного в ПЛК.	'PLC01'	Это поле является STRING и принимает алфавитно-цифровые (буквы и цифры) символы и символ "_". Не допускается инициировать STRING с цифрами или с "_". Допускается до 49 символов.

5. КОНФИГУРАЦИЯ

Адрес шлюза	IP-адрес компьютера, на котором установлен OPC DA Server, для случаев, когда все ПЛК находятся в одной подсети. Если какой-то ПЛК находится в другой подсети, то необходимо указать шлюз, используемый в этой подсети.	127.0.0.1	0.0.0.0 до 255.255.255.255
Порт шлюза	TCP-порт для соединения со шлюзом.	1217	2 до 65534
Наименование устройства	Это имя ПЛК, отображаемое в Настройках связи на вкладке Устройство. Имя представляет собой СТРОКУ перед шестнадцатеричным значением, находящимся между []. Включается только в том случае, если не установлен флажок Use TCP/IP Blockdriver.	'0000'	Это поле является STRING и принимает любые символы, как это сделано при настройке имени ПЛК на вкладке Device Communication Settings. Допускается до 49 символов.
IP-адрес Активный	IP-адрес ПЛК. Включается только при установленном флажке Use TCP/IP Blockdriver. Используется только в том случае, если настройка не является избыточной.	192.168.15.1	0.0.0.0 до 255.255.255.255
IP-адрес ПЛК А	IP-адрес ПЛК А. Включается только при резервировании настройки. Это основной адрес ПЛК, с которым сервер будет связываться в случае отсутствия сбоя.	192.168.15.69	0.0.0.0 до 255.255.255.255
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	0.0.0.0 до 255.255.255.255
Конфигурация устройства	Описание	Настройка по умолчанию	Возможности
Порт устройства	TCP Порт. Включается только при установленном флажке Use TCP/IP Blockdriver.	11740	11740 или 11739

Table 128: Configuration Parameter of each PLC for the OPC DA Server

5. КОНФИГУРАЦИЯ

При необходимости конфигурирования нового ПЛК на сервере OPC DA Server достаточно нажать кнопку New Device, и конфигурация будет создана. При обращении к экрану настройки отображается список всех ПЛК, уже сконфигурированных на сервере OPC DA Server. Существующие конфигурации можно редактировать, выбрав ПЛК в списке устройств и отредактировав его параметры. Параметры ПЛК, которые больше не используются, могут быть удалены. Максимальное количество ПЛК, сконфигурированных в OPC DA Server, составляет 16.

Если используемая архитектура автоматизации предусматривает, что OPC DA Server должен быть запущен на компьютере, на котором не выполняется связь с ПЛК через MasterTool IEC XE, то на этом компьютере должен быть установлен данный инструмент, позволяющий конфигурировать OPC DA Server так же, как это делается в других ситуациях.

ВНИМАНИЕ

То Для сохранения конфигурации OPC DA Server необходимо запустить MasterTool IEC XE с правами администратора на операционной системе. В зависимости от версии ОС эту привилегию необходимо сделать в момент запуска программы: щелкнуть правой кнопкой мыши на значке MasterTool IEC XE и выбрать пункт Запуск от имени администратора.

ВНИМАНИЕ

Настройки ПЛК на сервере OPC DA Server 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 OPC DA Server. При изменении настроек не требуется загружать приложение на ПЛК, но в зависимости от OPC-клиента может потребоваться повторное подключение к серверу или загрузка настроек для корректного обновления данных.

5.5.10.2.1. Импорт конфигурации проекта

С помощью кнопки Read Project Configuration, как показано на рис. 88, можно присвоить конфигурацию открытого проекта редактируемой конфигурации ПЛК. Для корректной работы этой опции должен быть открыт проект и установлен активный путь, как описано в разделе Настройки связи, 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.

При выполнении указанных условий в настройки ПЛК вводятся параметры открытого проекта. Информация об IP-адресе и порте шлюза настраивается, как описано в разделе "Настройки связи", в соответствии с активным маршрутом. Однако настройки IP-адреса считываются из настроек Ethernet-интерфейса NET 1. Порт для подключения к ПЛК в этом случае всегда назначается 11740.

5.5.10.3. Конфигурация с ПЛК на сервере OPC DA с резервированием соединений

Можно сконфигурировать сервер OPC DA Server таким образом, чтобы он работал с резервированием соединений. Таким образом, OPC DA Server будет взаимодействовать предпочтительно с одним ПЛК, но если по каким-либо причинам он не сможет установить связь с этим ПЛК, то будет обращаться ко второму, также сконфигурированному ПЛК. Такая конфигурация особенно важна для связи между SCADA-системами и ПЛК серии Nexto с полукластерным резервированием, когда один ПЛК находится в активном состоянии, выполняя процесс, а другой - в резервном состоянии, готовый взять управление процессом на себя в случае какого-либо сбоя.

Настройка проекта в этих случаях аналогична описанной в разделе [Создание проекта для связи OPC DA](#). Однако если проект создается с резервированием полукластера и связь с системой диспетчеризации будет осуществляться через OPC DA Server, то в мастере создания проекта MasterTool IEC XE необходимо выбрать опцию Configuration of OPC DA communication как включенную. При включении этой опции в проекте будет создан код, необходимый для осуществления связи с резервированием OPC-соединения.

В случае резервирования в POU объявляется переменная с именем NonSkippedPrg. Этот POU выполняется в обоих ПЛК, независимо от состояния резервирования. Внутри этого POU создается переменная типа BOOL, используемая для управления соединением с сервером OPC DA Server под названием OPCRedundancyActive. Доступ к этой

5. КОНФИГУРАЦИЯ

переменной возможен из любой точки приложения через весь контекст, т.е. Application.NonSkippedPrg.OPCRedundancyActive. Она объявляется в объекте Symbol Configuration с правом чтения только SCADA. Когда значение переменной равно TRUE, данные считываются при соединении с этим ПЛК. Таким образом, при каждом изменении состояния ПЛК состояние переменной также будет меняться, оставаясь в состоянии TRUE в том ПЛК, который находится в состоянии резервирования.

Код программы NonSkippedPrg на языке ST выглядит следующим образом:

```
PROGRAM NonSkippedPrg
VAR
  {атрибут 'symbol' := 'read'} OPCRedundancyActive : BOOL;
END_VAR

ЕСЛИ fbRedundancyManagement.m_fbDiagnosticsLocal.eRedState = REDUNDANCY_STATE. ACTIVE ТОГДА
  OPCRedundancyActive := TRUE;
ELSE
  OPCRedundancyActive := FALSE;
END_IF
```

Код программы NonSkippedPrg может быть отредактирован, если пользователь следит за тем, чтобы не изменить приведенный выше код. Этот код проверяет состояние резервирования и записывает в него переменную типа BOOL под названием OPCRedundancyActive. Если ПЛК активен, то значение переменной равно TRUE, в противном случае - FALSE. Этой переменной присваивается атрибут 'symbol' := 'read', позволяющий OPC DA Server получить доступ к содержимому и определить, откуда должна быть считана информация.

Если после создания проекта решено добавить OPC-связь, то можно сконфигурировать OPC, добавив приведенный выше код в программу NonSkippedPrg и добавив в проект объект Symbol Configuration.

Для конфигурирования резервного ПЛК на сервере OPC DA Server необходимо включить опцию Redundancy Configuration на экране конфигурации, как показано на рис. 88. При выборе этой опции всегда будет использоваться опция Use TCP/IP Blockdriver. Кроме того, будут включены поля IP Address PLC A и IP Address PLC B, как описано в табл. 128. Эти IP-адреса настраиваются на одинаковые интерфейсы Ethernet в проекте ПЛК с резервированием Half-Cluster. Для удобства конфигурирования при открытом проекте с резервированием можно использовать кнопку Read Project Configuration (Считать конфигурацию проекта).

ВНИМАНИЕ

Резервирование соединений OPC DA Server осуществляется только через один сервер. Для случаев, когда требуется повышенная доступность данных для систем наблюдения, необходимо использовать архитектуру с резервированием SCADA-сервера. В этом случае конфигурирование OPC DA Server не требуется. Какие конфигурации необходимы для работы резервированной архитектуры, см. в документации на SCADA-систему.

5.5.10.4. Переменные состояния и качества связи OPC DA

Для каждого ПЛК, созданного в OPC DA Server, формируются переменные состояния с именами _CommState и _CommStateOK. Переменная _CommState указывает на состояние связи между OPC и ПЛК. Это состояние может интерпретироваться OPC-клиентами в соответствии с приведенной ниже таблицей.

Состояние	Значение	Описание
-----------	----------	----------

5. КОНФИГУРАЦИЯ

STATE_TERMINATE	-1	Если связь между OPC DA Server и OPC Client прерывается, то будет возвращено это значение. Если одновременно подключено несколько OPC-клиентов, 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.
Состояние	Значение	Описание
STATE_PLC_NOT_CONNECTED	0	ПЛК, сконфигурированный в OPC DA Server, не подключен. Это может произойти в случае неправильной конфигурации (неверный IP-адрес ПЛК и/или шлюза) или недоступности ПЛК в данный момент.
STATE_PLC_CONNECTED	1	ПЛК, сконфигурированный в OPC DA Server, подключен. Это переходное состояние во время соединения.
STATE_NO_SYMBOLS	2	В ПЛК, сконфигурированном в OPC DA Server, нет доступных символов (переменных). Это может произойти, если нет символов или на ПЛК не загружен проект.
STATE_SYMBOLS_LOADED	3	Закончен процесс чтения символов (переменных) из ПЛК, сконфигурированного в OPC DA Server. Это переходное состояние во время соединения.
STATE_RUNNING	4	После считывания символов (переменных) OPC DA Server выполняет периодическое обновление значений доступных символов в каждом сконфигурированном ПЛК.
STATE_DISCONNECT	5	Произошло разъединение с ПЛК, сконфигурированным в 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, такое поведение не наблюдается, если MasterTool IEC XE поддерживает эту конфигурацию в действии.

Таблица 129: Описание состояний связи между OPC DA Server и ПЛК

_CommStateOK - это переменная типа Bool, которая показывает, работает ли связь между OPC DA Server и ПЛК. Если значение равно TRUE, это означает, что связь работает правильно. Если значение FALSE, то по какой-то причине связь с ПЛК невозможна.

Помимо контроля состояния связи, OPC-клиент может получить информацию о качестве связи. Биты качества образуют байт. Они разделены на три группы битов: Quality, Substatus и Limit. 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 не может быть использовано из-за проблем с соединением. Можно проконтролировать значение _CommState и диагностировать проблему.
1	01SSSSL	Неопределенно	Качество не может быть определено и может быть представлено в поле Substatus.
2	10SSSSL	Не указано	Это значение зарезервировано и не используется стандартом OPC.
3	11SSSSL	Хорошо	Качество хорошее, и считанное значение может быть использовано.

Таблица 130: Описание значения OPC Quality

В табл. 130 представлены возможные значения качества. Сервер OPC DA Server возвращает только значения Good и Bad Quality. OPC-клиент может поддерживать качество как Uncertain из-за сбоев, при которых он не может установить соединение с сервером. В случае мониторинга 8 битов качества непосредственно с сервера OPC DA Server поля Substatus и Limit должны быть нулевыми, и хорошее качество будет представлено как значение 192, а плохое качество - как значение 0.

5.5.10.5. Пределы связи с сервером OPC DA

Для связи с 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 в OPC DA Server. Таким образом, 20000 переменных - это максимальный предел для связи с одним ПЛК.

Кроме того, при конфигурировании переменных, доступных в OPC DA Server, количество объявленных переменных в каждом POU или GVL не может превышать предела в 5000. В таблице ниже представлены ограничения на конфигурацию OPC DA Server.

Максимальное количество переменных, взаимодействующих с одним ПЛК	20000
Максимальное количество переменных, объявленных в одном POU или GVL	5000
Максимальное количество ПЛК в OPC DA Server	16
Максимальное количество одновременных подключений OPC DA Server в одном ПЛК	8

Таблица 131: Ограничения связи с сервером OPC DA

ВНИМАНИЕ

Максимальное количество одновременных подключений OPC DA Server в одном ПЛК разделяется с подключениями, выполняемыми с помощью MasterTool IEC XE. Т.е. сумма подключений OPC DA Server и MasterTool IEC XE не должна превышать максимального количества, определенного в табл. 131.

Для связи между сервером OPC DA Server и ПЛК используется тот же протокол, что и для связи MasterTool IEC XE с ПЛК. Этот протокол доступен только для Ethernet-интерфейсов процессоров серии Nexto, с модулями расширения Ethernet такой обмен данными невозможен.

5. КОНФИГУРАЦИЯ

При установлении связи между OPC DA Server и ПЛК эти два элемента начинают серию транзакций, направленных на решение адресов каждой из объявленных переменных, оптимизируя связь в режиме чтения данных. Кроме того, на этом этапе для оптимизации связи определяются коммуникационные группы, используемые некоторыми клиентами. Этот начальный процесс требует определенного времени и зависит от количества отображаемых переменных.

Примерное время этого 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.

5.5.10.6. Доступ к данным через OPC DA Client

После конфигурирования OPC DA Server доступ к имеющимся данным всех ПЛК можно получить через OPC Client. В конфигурации OPC-клиента необходимо выбрать имя OPC DA Server. В данном случае это имя CoDeSys.OPC.DA. На рисунке ниже показан выбор сервера в клиентском драйвере программы BluePlant SCADA.

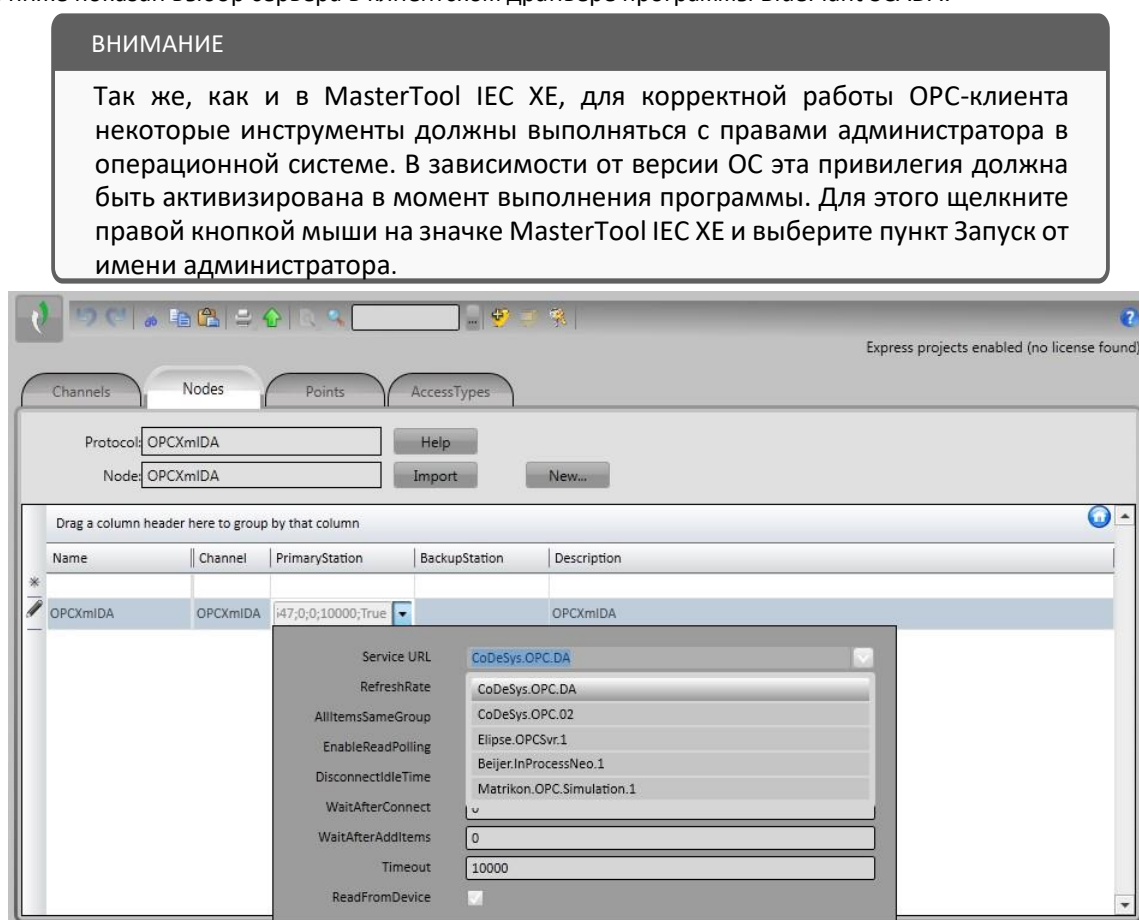


Рисунок 89: Выбор сервера OPC DA Server в конфигурации клиента

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. Однако более простым способом является использование туннельного инструмента, который абстрагирует настройки COM/DCOM и обеспечивает более безопасное взаимодействие между клиентом и сервером. Более подробная информация об этом типе инструментов приведена в статье NAP151 - Tunneller OPC.

После того как клиент соединился с сервером, можно использовать команды импорта TAGs. Эти команды обращаются к информации, объявленной в ПЛК, и возвращают список со всеми имеющимися в нем символами.

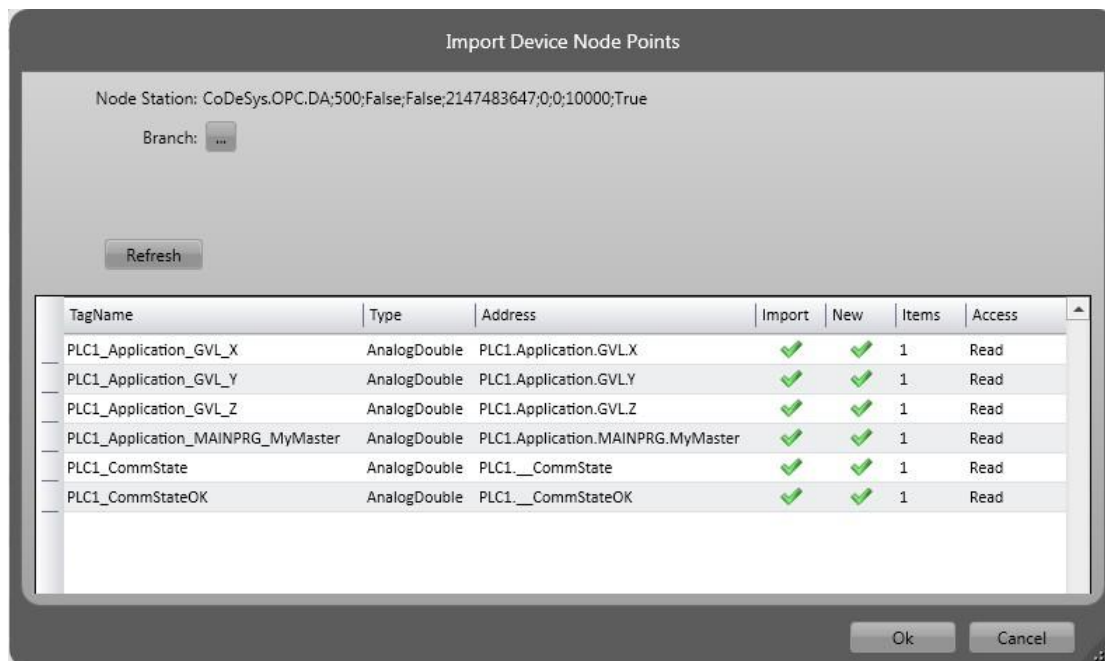


Рисунок 90: Список символов, к которому обращается OPC-клиент

Список выбранных переменных будет включен в список коммуникаций клиента и может быть использован, например, в экране SCADA-системы.

ВНИМАНИЕ

Для тестирования OPC-коммуникаций можно использовать режим симуляции программы MasterTool IEC XE. Информация по его настройке представлена в разделе "Тестирование OPC-коммуникаций с помощью симулятора" руководства пользователя MasterTool IEC XE - MU299609.

5.5.11. OPC UA Сервер

Протокол OPC UA является развитием семейства OPC. Независимо от платформы он призван стать новым стандартом, используемым в промышленных коммуникациях.

Основанный на архитектуре клиент/сервер, протокол OPC UA 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 контроллера. При изменении областей данных в процессе разработки проекта необходимо заново создавать сопоставления и новые таблицы с взаимосвязями между информацией ПЛК и SCADA-системы.

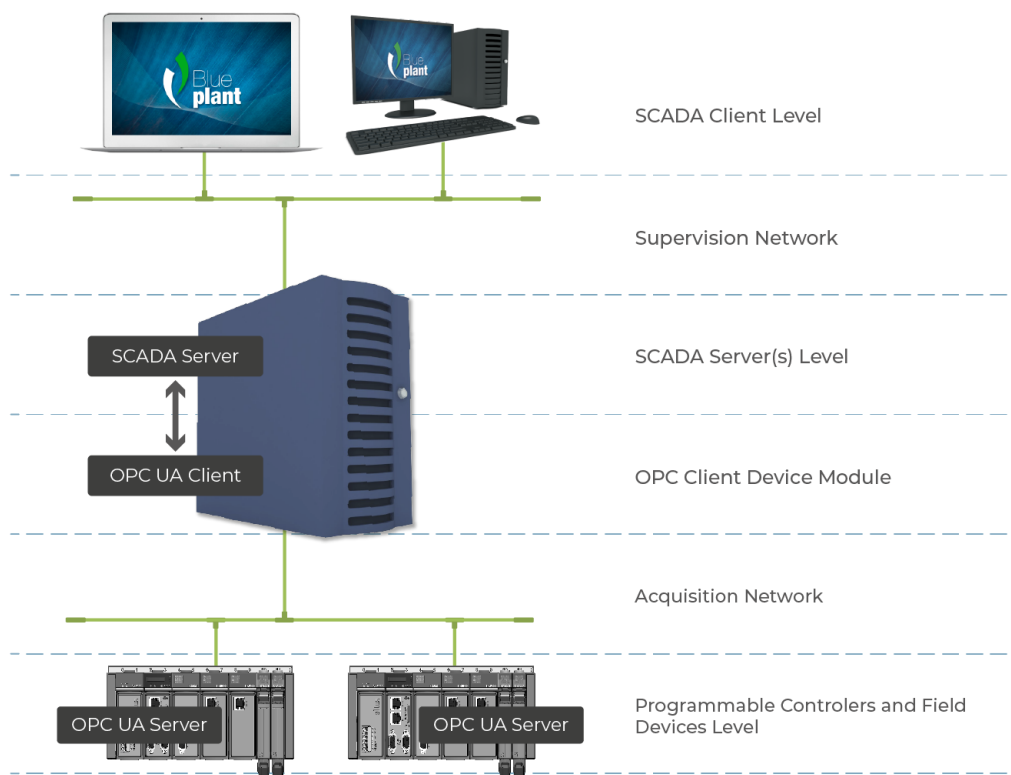


Рисунок 91: Архитектура OPC UA

На рисунке представлена типичная архитектура связи SCADA-системы и ПЛК при проектировании систем автоматизации. На этом рисунке явно представлены все роли, присутствующие в коммуникации, независимо от того, где они выполняются, - на одном и том же оборудовании или на разных. Каждая из ролей этой архитектуры описана в таблице ниже.

Роль	Описание
Уровень программируемых контроллеров и полевых устройств	В полевых устройствах и ПЛК хранится информация о состоянии работы и управлении установкой. SCADA-система получает доступ к информации на этих устройствах и хранит ее на SCADA-сервере, чтобы клиенты SCADA могли обращаться к ней во время работы установки.
Серверные модули OPC UA	Сервер OPC UA - это внутренний модуль ПЛК, отвечающий за прием запросов OPC UA и их трансляцию для связи с полевыми устройствами.
Сеть сбора данных	Сеть сбора данных - это сеть, по которой передаются сообщения OPC UA для запроса данных, собираемых с ПЛК и полевых устройств.
Модуль клиентского устройства OPC	Модуль OPC UA Client, входящий в состав SCADA-сервера, отвечает за выполнение запросов к OPC UA-серверам по протоколу OPC UA. Собранные им данные сохраняются в базе данных SCADA-сервера.

5. КОНФИГУРАЦИЯ

Уровень сервера SCADA	Сервер SCADA Server отвечает за подключение к различным коммуникационным устройствам и хранение собранных ими данных в базе данных, чтобы к ним могли обращаться клиенты SCADA.
Сеть наблюдения	Диспетчерская сеть - это сеть, по которой Клиенты SCADA подключаются к Серверам SCADA, часто с использованием проприетарного протокола системы SCADA. 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	Клиенты SCADA отвечают за запрос к серверам SCADA необходимых данных для отображения на экране, где происходит работа установки. Через них можно выполнять считывания и записи по данным, хранящимся в базе данных SCADA-сервера.

Таблица 132: Описание ролей в архитектуре сервера OPC UA

При использовании протокола OPC UA связь между метками 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. Достаточно просто использовать новую переменную, предоставленную ПЛК, в системах, запрашивающих эти данные.

Использование OPC UA обеспечивает более высокую производительность и связь с системами SCADA. Это способствует сокращению времени разработки приложений и расходов на их 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 Ethernet-интерфейсах процессоров Nexto. Модули расширения Ethernet не поддерживают эту функциональность.

5.5.11.1. 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 OPC DA](#). Как и в случае с протоколом OPC DA, конфигурация протокола OPC UA основана на конфигурации Symbol Configuration. Чтобы включить OPC UA, достаточно включить опцию Support OPC UA Features в конфигурации, как показано на рисунке ниже.

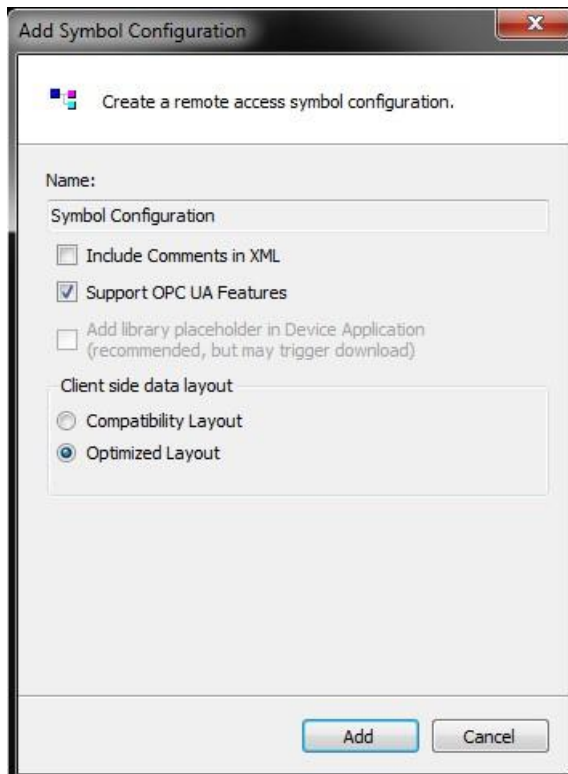


Рисунок 92: Конфигурация символа объекта

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 переменные, настроенные в объекте Symbol Configuration (Конфигурация символа) или через атрибуты.

Другой способ получить доступ к этой конфигурации, если уже создан проект с объектом Symbol Configuration, заключается в обращении к меню Settings вкладки configuration в Symbol Configuration. Просто выберите опцию Support OPC UA features, чтобы включить поддержку протокола OPC UA, как показано на рисунке ниже.

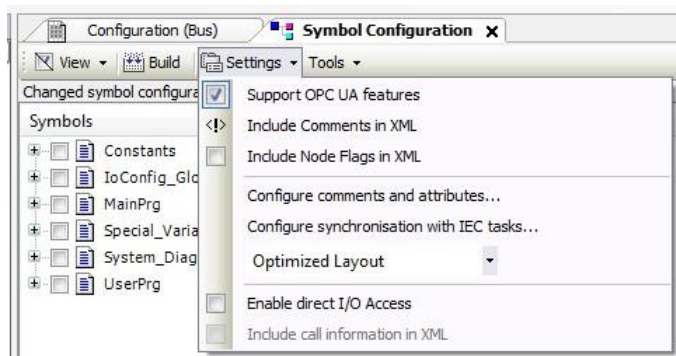


Рисунок 93: Включение OPC UA в конфигурации символа объекта

После этой процедуры проект может быть загружен в ПЛК, и выбранные переменные будут доступны для связи с OPC UA Server.

5. КОНФИГУРАЦИЯ

5.5.11.2. Типы поддерживаемых переменных

В данном разделе определены типы переменных, поддерживающих связь по протоколу OPC UA, если они объявлены в GVL или POU и выбраны в объекте 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 (STRUCTы или функциональные блоки), созданные на основе предыдущих простых типов.

Наконец, можно создавать массивы из простых или структурированных типов.

5.5.11.3. Ограничение количества подключенных клиентов на сервере OPC UA

Максимальное количество одновременно подключенных клиентов OPC UA в ПЛК составляет 8 (восемь).

5.5.11.4. Ограничение переменных связи на сервере OPC UA

Максимальное количество переменных, конфигурируемых в ПЛК для связи через OPC UA, составляет 20000. Однако при конфигурировании переменных, которые будут доступны для сервера OPC UA, количество переменных, объявленных в каждом POU или GVL, не должно превышать предела в 5000, при этом для использования 20000 переменных требуется не менее четырех POU или 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 OPC UA Server и ПЛК эти два элемента иницируют серию транзакций, направленных на решение адреса каждой объявленной переменной, оптимизируя связь в режиме чтения данных. Кроме того, на этом этапе также решаются вопросы классификации коммуникационных групп, используемых некоторыми клиентами, с целью оптимизации связи. Этот начальный процесс занимает некоторое время и зависит от количества отображаемых переменных..




5.5.11.5. Настройки шифрования

При желании пользователь может настроить шифрование для связи OPC UA с использованием профиля Basic256SHA256 для обеспечения безопасного соединения (кибербезопасности).



Чтобы настроить шифрование на сервере OPC UA, необходимо создать для него сертификат, выполнив следующие действия в программаторе MasterTool:

1. Определите активный путь для связи с контроллером (вход в систему не требуется);
2. В меню Вид выберите пункт Экран безопасности;

5. КОНФИГУРАЦИЯ

3. Перейдите на вкладку Устройства в левой части этого экрана;
4. Щелкните на значке , чтобы выполнить обновление;
5. Щелкните на значке Устройство, под которым откроется несколько сертификатов (Собственные сертификаты, Доверенные сертификаты, Недоверенные сертификаты, Сертификаты на карантине)
 6. Щелкните на  значке генерации сертификата и выберите следующие параметры
 - Длина ключа (бит): 3072
 - Срок действия (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 выполнить необходимые процедуры для подключения к серверу OPC UA и генерации сертификата с профилем Basic256Sha256 (подробности см. в руководстве по конкретному клиенту OPC UA);
10. Вернитесь в MasterTool, щелкните на значке  экрана безопасности, чтобы выполнить обновление;
11. На экране безопасности выберите папку "Карантинные сертификаты" в разделе "Устройство". В правой панели должен появиться сертификат, запрашиваемый клиентом OPC UA;
12. Перетащите этот сертификат в папку "Доверенные сертификаты";
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 выберите пункт Экран безопасности;
3. Щелкните на Устройства в левой части этого экрана;
4. Щелкните на значке , чтобы выполнить обновление;
5. Щелкните на значке Устройства, под которым откроется несколько сертификатов (Собственные сертификаты, Доверенные сертификаты, Недоверенные сертификаты, Сертификаты на карантине);
6. Щелкните по папке "Собственные сертификаты" и в правой панели выберите сертификат (OPC UA Server);
7. Щелкните на значке , чтобы удалить этот проект и сертификат драйвера;
8. Перезагрузите (выключите и включите) контроллер

5.5.11.6. Основные параметры связи, настраиваемые в клиенте OPC UA

Некоторые коммуникационные параметры OPC UA настраиваются на клиенте OPC UA и согласовываются с сервером OPC UA в момент установления соединения между ними. В следующих подразделах описаны основные коммуникационные параметры OPC UA, их значение и порядок выбора соответствующих значений.

В клиенте OPC UA можно группировать переменные сервера в различные подписки. 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 сети Ethernet. Подобные аспекты оптимизации более подробно анализируются в прикладной записке NAP165, где предлагаются некоторые правила составления подписок. В данной прикладной записке также более подробно рассматриваются некоторые понятия, связанные с протоколом OPC UA.

5. КОНФИГУРАЦИЯ

Некоторые из описанных ниже параметров связи должны быть определены для сервера в целом, другие - для каждой подписки, третьи - для каждой переменной, входящей в подписку.

5.5.11.6.1. URL конечной точки

Этот параметр задает IP-адрес и TCP-порт сервера, например:

```
opc.tcp://192.168.17.2:4840
```

В данном примере IP-адрес контроллера равен 192.168.17.2.

TCP-порт всегда должен быть равен 4840.

5.5.11.6.2. Интервал публикации (мс) и Интервал выборки (мс)

Для каждой подписки должен быть задан параметр Интервал публикации (единицы измерения: миллисекунды).

Параметр "Интервал выборки" должен быть задан для каждой переменной (единицы измерения: миллисекунды). Однако во многих клиентах OPC UA параметр "Интервал выборки" может быть задан для подписки, причем одинаково для всех переменных, объединенных в подписку.

В коммуникационном пакете "Публикация ответа" клиенту сообщаются только те переменные подписки, значения которых были изменены. Параметр "Интервал публикации" определяет минимальный интервал между последовательными пакетами Publish Response одной и той же подписки, чтобы ограничить потребление полосы обработки и Ethernet-связи.

Для того чтобы выяснить, какие переменные подписки изменились и должны быть сообщены клиенту в следующем пакете Publish Response, сервер должен выполнить сравнения, и такие сравнения (выборки) выполняются с помощью того же параметра "Интервал выборки". Рекомендуется, чтобы значение Интервала выборки колебалось между 50% и 100% от значения Интервала публикации, так как при этом возникает относительно большое потребление обработки, связанное с процессом сравнения, выполняемым в каждом Интервале выборки.

Можно сказать, что сумма между Интервалом публикации и Интервалом выборки - это максимальная задержка между изменением значения на сервере и передачей пакета Publish Response, сообщающего об этом изменении. Половина этой суммы - средняя задержка между изменением значения на сервере и передачей пакета Publish Response, сообщающего об этом изменении.

5.5.11.6.3. Счетчик срока службы Keep-Alive Count

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 коммуникационные пакеты PublishRequest клиента для этой подписки. Пакеты PublishRequest должны быть получены сервером для того, чтобы он мог транслировать пакеты Publish Response, содержащие переменные подписки, изменившие свои значения.

Если сервер не получает пакеты PublishRequest в течение времени, превышающего Lifetime Count, умноженный на Publishing Interval, то он деактивирует подписку, которая в дальнейшем должна быть вновь создана клиентом по желанию.

В ситуациях, когда переменные подписки не меняются, может долгое время не передаваться PublishResponses и, соответственно, PublishRequests, что приведет к нежелательной деактивации подписки. Чтобы этого не произошло, был создан параметр Keep-Alive Count. Если в течение времени, равного Keep-Alive Count, умноженному на Publishing Interval, не происходит изменений данных подписки, то сервер посылает небольшой пустой пакет Publish Response, указывающий на то, что переменная не изменилась. Этот пустой Publish Response разрешает клиенту немедленно отправить следующий PublishRequest.

Значение Keep-Alive Count должно быть меньше значения Lifetime Count для предотвращения нежелательной деактивации подписки. Рекомендуется, чтобы значение LifeTime Count было как минимум в 3 раза больше, чем Keep-Alive Count.

5.5.11.6.4. Размер очереди и Discard Oldest

Эти параметры должны поддерживаться со следующими фиксированными значениями, которые обычно являются значениями по умолчанию на клиентах:

- Размер очереди: 1

5. КОНФИГУРАЦИЯ

- Discard Oldest: включить

Согласно стандарту OPC UA, можно определить эти параметры для каждой переменной. Однако многие клиенты позволяют определить общие значения для всех переменных, настроенных в подписке.

Размер очереди следует оставить равным 1, поскольку в данной реализации сервера OPC UA нет поддержки событий, поэтому определять очередь нет необходимости. Увеличение значения размера очереди может привести к увеличению пропускной способности канала связи и процессорной обработки, поэтому этого следует избегать.

Discard Oldest должен поддерживаться со значением включения, чтобы пакет Publish Response всегда сообщал о последнем изменении значения, обнаруженном для каждой переменной.

5.5.11.6.5. Тип фильтра и тип мертвой зоны

Эти параметры должны поддерживаться со следующими фиксированными значениями, которые обычно являются значениями по умолчанию в клиентах:

- Тип фильтра: Фильтр изменения данных
- Тип мертвой зоны: нет

Согласно стандарту OPC UA, можно определить эти параметры для каждой переменной. Однако многие клиенты позволяют определить общие значения для всех переменных, сконфигурированных в подписке.

Параметр Filter Type должен иметь значение DataChangeFilter, указывающее на то, что изменение значения переменных должно вызывать его передачу в пакете Publish Response.

Параметр Deadband Type должен иметь значение "none", поскольку для аналоговых переменных не существует реализации deadband. Таким образом, любое изменение аналоговой переменной, каким бы минимальным оно ни было, приводит к ее передаче в пакете Publish Response.

Для снижения вычислительной мощности и пропускной способности сети Ethernet можно самостоятельно развернуть "мертвые зоны" следующим образом:

- Не включайте аналоговую переменную в подписку;
- Вместо этого включить в подписку вспомогательную переменную, связанную с аналоговой переменной;
- Копировать аналоговую переменную во вспомогательную переменную только при экстраполяции управляемой пользователем мертвой зоны.

5.5.11.6.6. Публикация включена, максимальное количество уведомлений на публикацию и приоритет

Предлагается сохранить следующие значения параметров, которые обычно используются в клиентах по умолчанию:

- *Publishing* Включено: *true*
- *MaxNotificationsPerPublish*: *0*
- *Priority*: *0*

Эти параметры должны быть настроены для каждой подписки.

PublishingEnable должен иметь значение "true", чтобы переменные подписки сообщались в случае изменения значения.

MaxNotificationsPerPublish указывает, 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 Publish Response.

Priority указывает на относительный приоритет данной подписки по сравнению с другими. Если в какой-то момент времени серверу необходимо отправить несколько пакетов 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 Publish Response будут передаваться в фиксированной последовательности.

5.5.11.7. Доступ к данным через клиент OPC UA

5. КОНФИГУРАЦИЯ

После конфигурирования сервера 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. В данном случае это адрес `opc.tcp://ip-address-of-device:4840`. На рисунке ниже показан выбор сервера в драйвере клиентского ПО SCADA BluePlant.

ВНИМАНИЕ

Как и MasterTool IEC XE, для корректной работы OPC UA Client 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 Запустить от имени администратора.

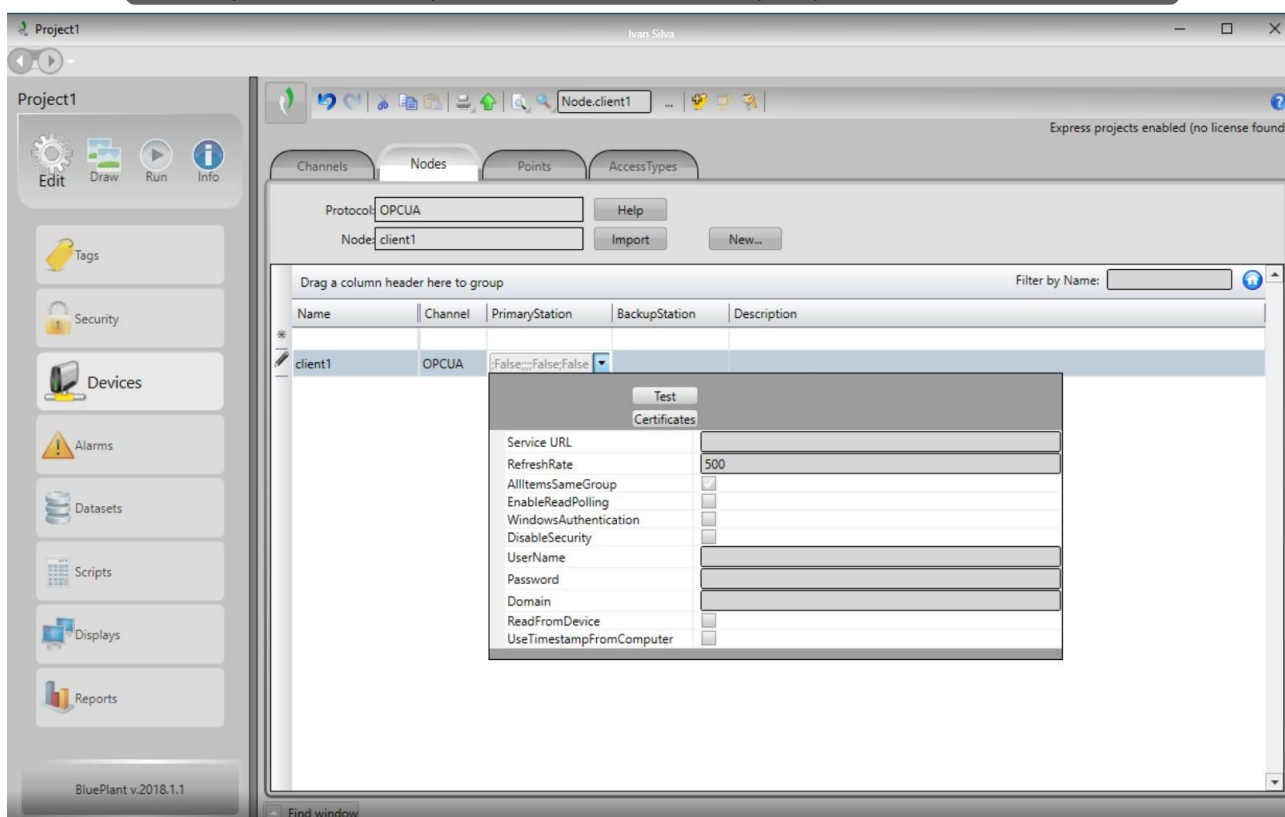


Рисунок 94: Выбор сервера OPC UA в конфигурации клиента

После того как клиент подключится к серверу, можно использовать команды импорта TAG. Эти команды запрашивают информацию, объявленную в ПЛК, и возвращают список со всеми символами, доступными в ПЛК.


5. КОНФИГУРАЦИЯ

Slave Information), поставляемые вместе с оборудованием. Описания ведомых устройств доступны в виде XML-файлов (тип файла: *.xml).

Мастер EtherCAT может быть добавлен в дерево устройств с помощью команды Add Device (Добавить устройство), через контекстное меню разъемов CPU NET.

Под ведущим устройством можно добавить одно или несколько ведомых устройств, выбрав EtherCAT-ведущее устройство и выполнив команду Add Device (контекстное меню EtherCAT-ведущего) или выполнив команду Scan For Devices.

Помимо линейной и древовидной топологий, MasterTool IEC XE также поддерживает топологию "звезда" в EtherCAT. Для конфигурирования топологии "звезда" EtherCAT необходимы специальные EtherCAT-переходы (в примере: Beckhoff EK1122). Модульная звезда EtherCAT может быть выполнена с использованием различных джойстиков. В порты джойстика могут быть подключены отдельные устройства или целая секция EtherCAT-линии. EtherCAT-переход

обозначается пиктограммой . Пример дерева устройств на рисунке ниже демонстрирует различные возможности.

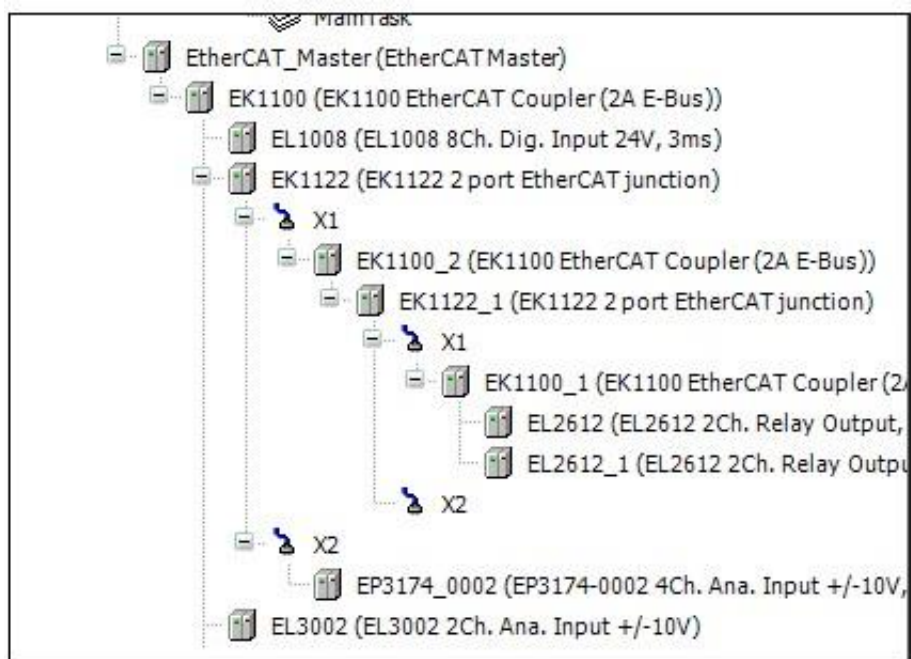


Рисунок 96: Пример конфигурации EtherCAT

ВНИМАНИЕ

- В одном проекте допускается только один экземпляр EtherCAT Master.
- Доступен только на разъемах NET ПЛК.
- Не может использоваться, если NET установлены как резервные.
- Не может использоваться, если проект имеет кластерное резервирование.
- Другие драйверы не могут быть установлены в тот же порт NET, что и EtherCAT Master.
- Поддерживается не более 128 ведомых устройств EtherCAT в одном проекте.

5.5.12.1.1. Сканирование устройств

Команда Scan For Devices, доступная в контекстном меню EtherCAT Master, выполняет поиск Slave-устройств, физически установленных в EtherCAT-сети подключенного в данный момент ПЛК. Это означает, что с помощью данной команды можно обнаружить и визуализировать аппаратные компоненты в окне, представленном на рисунке ниже, что позволяет пользователю отобразить их непосредственно в проекте Device Tree do projeto.

5. КОНФИГУРАЦИЯ

Следует отметить, что при выборе команды "Сканировать устройства" (Scan For Devices) соединение с ПЛК будет автоматически установлено перед началом поиска и разорвано по его окончании. Таким образом, для первого выполнения этой команды необходимо сконфигурировать шлюзовое соединение и загрузить в ПЛК программу с конфигурацией EtherCAT Master. В случае последующего добавления Slave-устройств для выполнения этой команды необходимо остановить EtherCAT-сеть. Для этого необходимо установить в TRUE бит bStopBus, присутствующий в переменных диагностики EtherCAT-мастера.

После выполнения команды поле Scanned Devices будет содержать список всех устройств и модулей, найденных при последнем сканировании. Чтобы добавить их в проект, достаточно щелкнуть на кнопке Copy All Devices To Project. Также можно выполнить сравнение найденных при поиске устройств с теми, что находятся в проекте, установив флажок Show differences to project.

Если добавить в проект EtherCAT Master-модуль и воспользоваться командой Scan For Devices, то появится список всех доступных EtherCAT Slave. Записи, выделенные жирным шрифтом, будут показаны, если имеется несколько устройств с одинаковым описанием. При двойном щелчке на входе откроется список, из которого можно выбрать нужное устройство.

После завершения изменений в конфигурации EtherCAT-сети необходимо выполнить новую загрузку проекта, чтобы изменения вступили в силу.

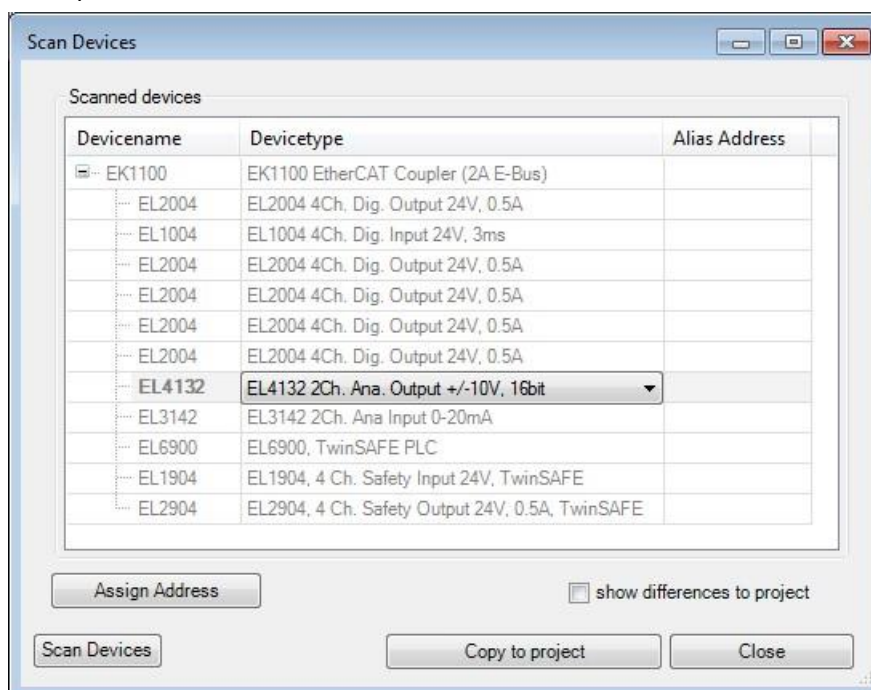


Рисунок 97: Диалог поиска EtherCAT-устройств

5.5.12.2. Диагностические переменные

При вставке ведущего и ведомого EtherCAT-устройств в GVL System_Diagnostics для них добавляется диагностическая переменная. Эта переменная предоставляет информацию о состоянии устройства. Существует два типа переменных: одна для EtherCAT Master, другая для EtherCAT Slave. Каждая переменная содержит специфическую информацию об устройстве. Описание предоставляемой диагностики и команд приведено в следующих таблицах.

Переменная DG_EtherCAT_Master.*	Тип	Возможные значения	Описание
tDiag. bRunning	BOOL	FALSE или TRUE	Если эта переменная равна TRUE, то передача всех параметров конфигурации успешно завершена, ошибок не обнаружено, и шина не была

5. КОНФИГУРАЦИЯ

			остановлена по команде пользователя. Связь запущена.
tDiag. bError	BOOL	FALSE или TRUE	Данная переменная будет иметь значение TRUE, если при запуске EtherCAT-стека или в процессе работы обнаружена ошибка, связь с ведомыми устройствами прекращается, если больше не может быть получено ни одного сообщения (например, из-за повреждения кабеля). Причина ошибки может быть выявлена с помощью списка журналов через error STRING.
tDiag. eLastErrorCode	ETC_LASTERROR	-	Информация о возможных значениях этой диагностики, а также ее описание приведены в табл. 134 .
Переменная DG_EtherCAT_Master.*	Тип	Возможные значения	Описание
tDiag. bDistributedClockInSync	BOOL	FALSE или TRUE	Если используется DC, то ПЛК будет синхронизирован с первым EtherCAT-ведомым, чья настройка DC активна. Вскоре после успешного завершения этой синхронизации данная переменная принимает значение TRUE. Этот сигнал, например, может быть использован для инициализации функциональных блоков Soft Motion в случае совместимости с устройством после того, как ПЛК перейдет в режим синхронизации, иначе могут возникнуть скачки положения. При запуске ПЛК эта переменная имеет значение FALSE и через несколько секунд изменится на TRUE. При потере синхронизации в результате какого-либо сбоя переменная изменится на FALSE.
tDiag.bReserved_00	BOOL	-	Зарезервированное пространство.
tDiag.bReserved_01	BOOL	-	Зарезервированное пространство.
byReserved_00	BYTE	-	Зарезервированное пространство.
tCommand. bRestart	BOOL	FALSE или TRUE	По нарастающему фронту мастер полностью перезагрузится. Все параметры конфигурации будут перезагружены.
tCommand. bStopBus	BOOL	FALSE или TRUE	Когда эта переменная имеет значение TRUE, обмен данными прекращается. EtherCAT-пакеты не будут отправляться. Для большинства устройств перезапуск

5. КОНФИГУРАЦИЯ

			необходим, поскольку они находятся в состоянии ошибки.
tCommand. wDCInSyncWindow	BOOL	0..65535	Временное окно для bDistributedClockSync. Джиттер должен находиться внутри этого окна, чтобы bDistributedClockSync оставался TRUE. Значение по умолчанию: 50 микросекунд.
tCommand. bySlaveUpdatedbyCycle	BOOL	0..128	Это значение определяет количество ведомых устройств, которые будут считываться каждый цикл для заполнения диагностических переменных ведомых устройств. Значение 0 означает, что диагностика ведомых обновляться не будет.
tCommand.bReserved_00	BOOL	-	Зарезервированное пространство.
tCommand.bReserved_01	BOOL	-	Зарезервированное пространство.
byReserved_01	BYTE	-	Зарезервированное пространство.

Таблица 133: Диагностика и команды ведущего устройства EtherCAT

Код	Ключевое слово	Описание
00	NO_ERROR	Ошибок нет, работает.
01	NO_COMM	Связь отсутствует. Не получено более 100 пакетов. Возможна неисправность в главном кабеле.
02	WRONG_WORKING_COUNTER	Рабочий счетчик для обработки данных неверен. Один или несколько ведомых устройств не работают или отсутствуют, и ожидаемый рабочий счетчик не найден.
03	DC_TIME_ZERO	Slave DC Time всегда равен нулю -> возможно, неправильно подключены разъемы IN и OUT и не удается считать время с эталона времени.
04	OPEN_FIRSTADAPTER_FAILED	Не удастся открыть первый сетевой адаптер.
05	OPEN_SECONDDADAPTER_FAILED	Не удастся открыть второй сетевой адаптер.
Код	Ключевое слово	Описание
06	ADAPTER_MISMATCH	Второй сетевой адаптер использует MAC-ID первого интерфейса.
07	NO_SLAVES_FOUND	Ошибка при запуске ведомых устройств: Возможно, отсутствуют ведомые устройства или нет связи.
08	VENDOR_ID_WRONG	Идентификатор поставщика не равен.
09	PRODUCT_ID_WRONG	Product ID is not equal.
10	NUMBER_DEVICE_MISMATCH	Чтение Product ID или Vendor ID не удастся, в конфигурации больше ведомых устройств, чем в реальной архитектуре.
11	SDO_WRITE_ERROR	Ошибка записи SDO во время ввода в эксплуатацию.

5. КОНФИГУРАЦИЯ

12	SDO_TIMEOUT	Тайм-аут SDO во время ввода в эксплуатацию.
13	EMERGENCY_RECEIVED	Получен аварийный сигнал от устройства.
14	IDN_WRITE_ERROR	Ошибка записи IDN во время ввода в эксплуатацию.
15	IDN_TIMEOUT	Тайм-аут IDN во время ввода в эксплуатацию.

Table 134: EtherCAT Master Error Codes

ПеременнаяDG_Slave.*	Тип	Возможные значения	Описание
tDiag. wState	ETC_SLAVE_STATE	ETC_SLAVE_BOOT=3 ETC_SLAVE_INIT=1 ETC_SLAVE_PREOPERATIONAL=2 ETC_SLAVE_SAVEOPERATIONAL=4 ETC_SLAVE_OPERATIONAL=8	Текущее рабское состояние.
tDiag. dwVendorID	DWORD	Любой DWORD	После инициализации стека EtherCAT эта переменная возвращает Vendor ID, считанный с подчиненного устройства.
tDiag. dwProductID	DWORD	Любой DWORD	После инициализации стека EtherCAT эта переменная возвращает идентификатор продукта, считанный с подчиненного устройства.
tDiag. dwRevisionID	DWORD	Любой DWORD	После инициализации стека EtherCAT эта переменная возвращает идентификатор версии, считанный с подчиненного устройства.
tDiag. tLastEmergency	ETC_CO_Emergency	-	Если сообщение получено, то эта информация сохраняется в подчиненном устройстве и может быть прочитана в приложении через эту переменную. Также добавляется сообщение журнала. Более подробная информация об этом

5. КОНФИГУРАЦИЯ

			диагнозе содержится в таблице 136.
tDiag. bReserved_01	BOOL	-	Зарезервированное место.
byReserved_00	BYTE	-	Зарезервированное место.

Табл. 135: Диагностика подчиненного устройства EtherCAT

Переменная DG_Slave.tDiag. tLastEmergency.*	Тип	Шестнадцатеричный код	Описание
wErrorCode	WORD	00XX	Сбросить ошибку или нет ошибки.
		10XX	Общая ошибка.
		20XX	Текущий.
		21XX	Ток, внутри устройства.
		22XX	Ток внутри устройства.
		23XX	Ток, вне устройства.
		30XX	Напряжение.
		31XX	Основные напряжения.
		32XX	Напряжение внутри устройства.
		33XX	Выходное напряжение.
		40XX	Температура.
		41XX	Температура окружающей среды.
		42XX	Температура устройства.
byErrorRegister	BYTE	60XX	Аппаратное обеспечение устройства. Программное обеспечение устройства.

5. КОНФИГУРАЦИЯ

		61XX	Внутреннее программное обеспечение.
		62XX	Программное обеспечение пользователя.
		63XX	Набор данных.
		70XX	Дополнительные модули.
		80XX	Мониторинг.
		81XX	Коммуникация.
		82XX	Ошибка протокола.
		8210	PDO не обрабатывается из-за ошибки длины.
		8220	Превышена длина PDO.
		90XX	Внешняя ошибка.
		A000	Неудачный переход из ПРЕДЭКСПЛУАТАЦИОННОГО режима в БЕЗОПАСНО-ЭКСПЛУАТАЦИОННЫЙ.
		A001	Неудачный переход из БЕЗОПАСНОГО-ЭКСПЛУАТАЦИОННОГО в ЭКСПЛУАТАЦИОННЫЙ.
		F0XX	Дополнительные функции.
		FFXX	Конкретное устройство.
		BYTE Range	Ошибка регистрации.
tDiag. tLastEmergency. abyErrorField	ARRAY[0..5] OF BYTE	0000-9FFF	Поле ошибки производителя.
		A000-EFFF	Диагностические данные.
		F000-FFFF	Поле ошибки производителя.

Таблица 136: Содержимое ETC_CO_Emergency

5.5.12.3. Основные настройки EtherCAT

Ниже перечислены параметры для выполнения конфигурации EtherCAT Master, например, определенные в файле описания устройства.

5.5.12.3.1. Основные параметры

Ниже приведены общие параметры начального экрана конфигурации EtherCAT Master, как показано на рисунке ниже.

5. КОНФИГУРАЦИЯ

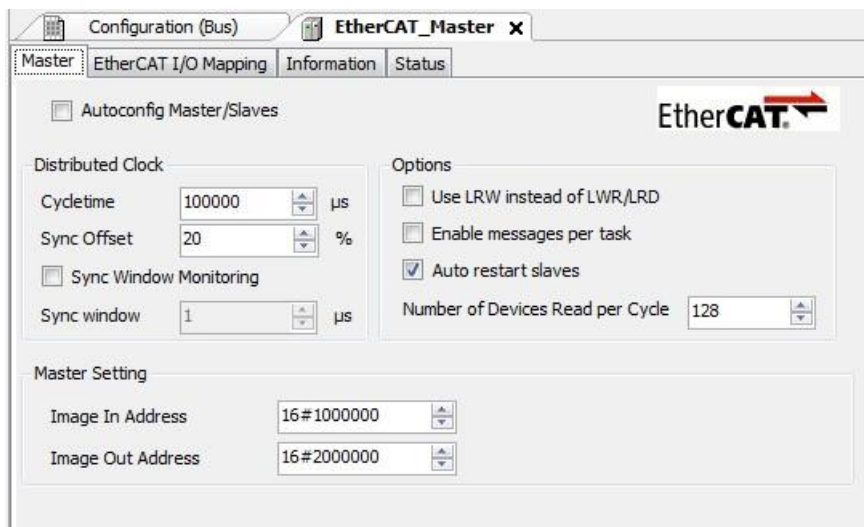


Рисунок 98: Диалоговое окно настройки главного устройства EtherCAT

Конфигурация устройства	Описание	По умолчанию	Возможные значения
Автонастройка Master/Slave	Включите автоматическую настройку Master и Slave.	Отмечено	Отмечено Без опознавательных знаков
Время цикла [мкс]	Устанавливает период времени, в течение которого новая телеграмма данных должна быть отправлена на шину.	100000	2000 до 1000000
Смещение синхронизации [%]	Отрегулируйте смещение прерывания синхронизации EtherCAT Slave относительно цикла ПЛК.	20	-50 до 50
Мониторинг окна синхронизации	Если эта опция включена, она позволяет контролировать синхронизацию Slave.	Без опознавательных знаков	Отмечено Без опознавательных знаков
Окно синхронизации [мкс]	Время Мониторинга Окна Синхронизации.	1	1 до 32768
Используйте LRW вместо LWR/LRD	Включение комбинированных команд чтения и записи.	Без опознавательных знаков	Отмечено Без опознавательных знаков
Включить сообщения для каждой задачи	Если включено, команды чтения и записи, которые имеют дело с входными и выходными сообщениями, могут выполняться в разных задачах.	Без опознавательных знаков	Отмечено Без опознавательных знаков

5. КОНФИГУРАЦИЯ

Ведомые устройства с автоматическим перезапуском	Перезапустите устройства, когда связь будет прервана.	Отмечено	Отмечено Без опознавательных знаков
Количество устройств, прочитанных за цикл	Устанавливает количество ведомых устройств, которые считываются за цикл для заполнения диагностических переменных. Значение «0» означает, что диагностика подчиненного устройства не будет обновляться.	128	0 до 128
Изображение в адресе	Первый входной адрес первого ведомого устройства.	16#1000000	16#1 до 16#1F000000
Конфигурация устройства	Описание	По умолчанию	Возможные значения
Выходной адрес изображения	Логический адрес первого выхода для первого ведомого устройства.	16#2000000	16#1 до 16#1F000000

Таблица 137: Конфигурация ведущего устройства EtherCAT

Примечания:

Autoconfig Master/Slave: если эта опция включена, большая часть конфигурации Master и Slave будет выполняться автоматически на основе файлов описания и неявных вычислений. В этом случае диалог FMMU/Sync будет недоступен. Если он не отмечен, параметры Image In Address и Image Out Address будут доступны пользователю.

ВНИМАНИЕ

Режим Autoconfig включен по умолчанию и обычно его достаточно и настоятельно рекомендуется для стандартных приложений. Если он отключен, все определения конфигурации должны быть сделаны вручную, и, следовательно, требуются некоторые специальные знания. Чтобы настроить связь между подчиненными, необходимо отключить параметр Autoconfig.

Время цикла: период времени, после которого на шину должна быть отправлена новая телеграмма данных. Если функция Distributed Clock включена, значение этого параметра будет передано на часы Slave. Таким образом достигается точная синхронизация обмена данными, что особенно важно в случаях, когда распределенный процесс требует одновременных действий. Таким образом, можно получить очень точную временную базу с дрожанием, значительно меньшим микросекунды, для всей сети.

Смещение синхронизации: это значение позволяет настроить смещение прерывания синхронизации подчиненного устройства EtherCAT относительно цикла ПЛК. Обычно цикл задач ПЛК начинается на 20 % позже, чем прерывание синхронизации ведомых устройств. Это означает, что задача ПЛК может быть задержана на 80 % времени цикла, и ни одно сообщение не будет потеряно.

Окно синхронизации: если синхронизация всех ведомых устройств находится в пределах этого временного окна, для диагностики bDistributedClockInSync главного устройства EtherCAT будет установлено значение TRUE, в противном случае — значение FALSE. При использовании распределенных часов настоятельно рекомендуется использовать выделенную задачу с высоким приоритетом в качестве задачи цикла шины главного устройства EtherCAT. Для этого необходимо использовать профили проекта, позволяющие создавать новые задачи, затем создать циклическую задачу с приоритетом 0 (задача реального времени) и связать ее с главной задачей цикла шины на вкладке EtherCAT

5. КОНФИГУРАЦИЯ

Master — I/O Mapping. мастера EtherCAT. Пользователь также может изменить значение переменной wDCInSyncWindow, настроив максимальное допустимое дрожание синхронизации между ведущим и ведомыми устройствами.

Использовать LRW вместо LWR/LRD: активация этой опции включает связь между ведомыми устройствами, поскольку вместо отдельных команд чтения (LRD) и записи (LWR) будут использоваться комбинированные команды чтения/записи (LRW).

Автоматический перезапуск ведомых устройств: при включении этой опции ведущее устройство перезапустит ведомые устройства, как только связь будет прервана, поэтому диагностика ведущего устройства bError EtherCAT в GVL System_Diagnostics не станет TRUE.

Image In Address и Image Out Address: эти определения можно редактировать только в том случае, если режим Autoconfig отключен, в противном случае это будет сделано автоматически, и этот параметр будет невидим.

Диагностическое сообщение: показывает некоторую информацию или сообщения об ошибках из стека. Сообщения также регистрируются на вкладке «Журнал ПЛК», доступ к которой осуществляется с помощью значка «Устройство» в дереве устройств. Этот параметр виден только тогда, когда EtherCAT Master находится в сети.

Нагрузка на шину: это значение показывает нагрузку на шину сетевого адаптера. Этот параметр отображается только тогда, когда мастер EtherCAT находится в сети.

5.5.12.3.2. EtherCAT Master — сопоставление входов/выходов

На этой вкладке редактора конфигурации EtherCAT Master можно назначить переменные проекта соответствующим входам и выходам EtherCAT. Таким образом, переменные EtherCAT Master могут управляться пользовательским приложением.

Кроме того, можно выбрать, какую задачу будет использовать задача цикла шины, с помощью параметров в списке выбора. Эта задача служит для выполнения операций EtherCAT Master. MainTask — параметр по умолчанию для этого поля.

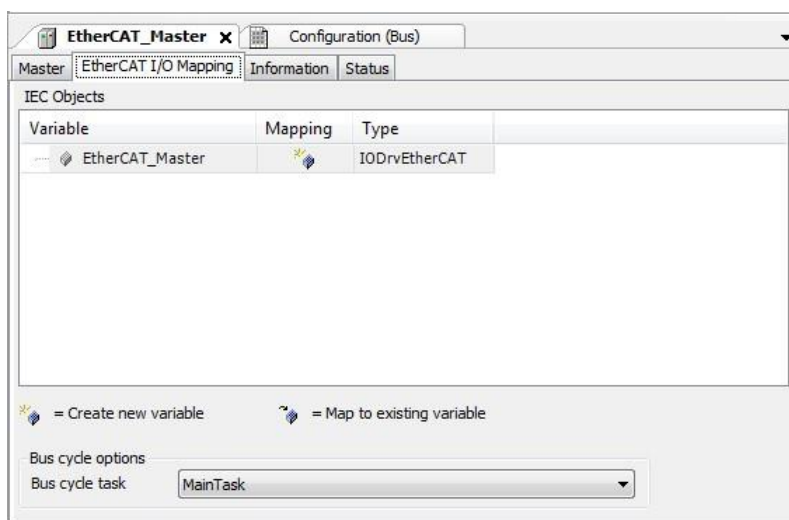


Рисунок 99: Диалоговое окно главного ввода/вывода

5.5.12.3.3. Вкладки состояния и информации

На вкладке «Состояние» редактора конфигурации EtherCAT Master отображается информация о состоянии (например, «Работает», «Остановлен») и диагностические сообщения, относящиеся к устройству и внутренней шинной системе.

Вкладка «Информация», присутствующая в редакторе конфигурации EtherCAT Master, показывает, если она доступна, следующую общую информацию о модуле: Имя, Производитель, Тип, Номер версии, Категория, Номер заказа, Описание, Изображение.

5.5.12.4. Конфигурация подчиненного устройства EtherCAT

Ниже перечислены основные параметры конфигурации ведомого устройства EtherCAT, определенные в файле описания устройства.

5. КОНФИГУРАЦИЯ

5.5.12.4.1. Ведомые параметры

Ниже представлены общие параметры начального экрана конфигурации EtherCAT Slave. Это поле доступно только в том случае, если режим автоконфигурации (главный) не включен.

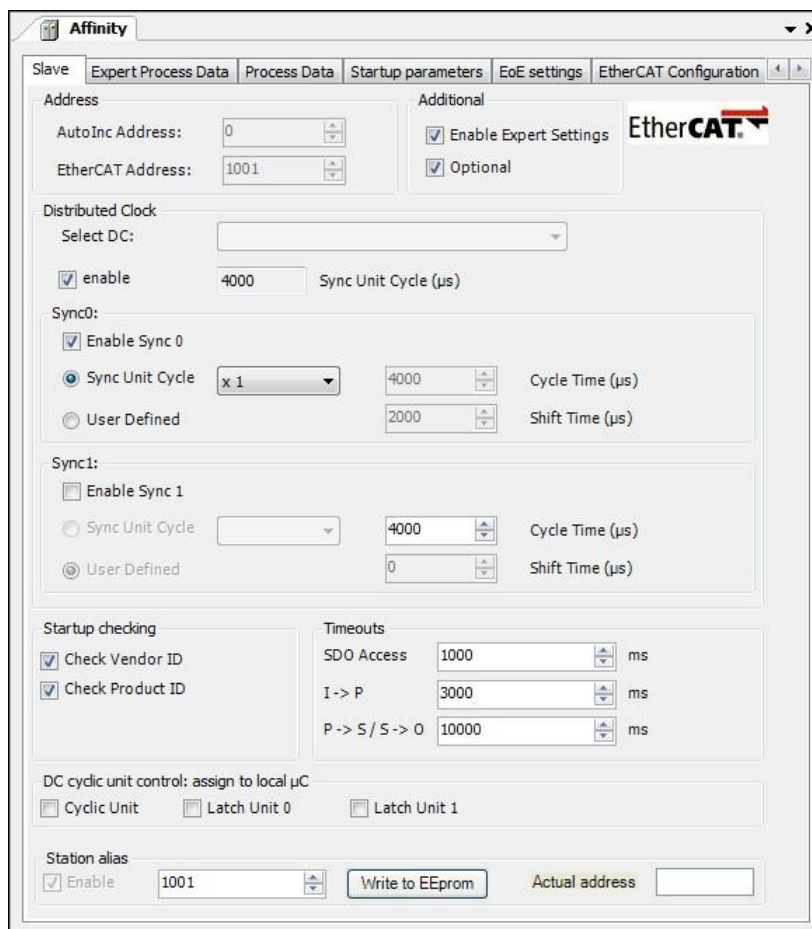


Рис. 100. Диалоговое окно конфигурации ведомого устройства EtherCAT

Конфигурация устройства	Описание	Значение по умолчанию	Варианты
Адрес AutoInc	Автоинкрементный адрес (16 бит), определяемый положением ведомого устройства в сети.	-	-65535 до 0
EtherCAT-адрес	Конечный адрес подчиненного устройства, назначаемый ведущим во время запуска. Этот адрес не зависит от положения в сети.	-	1 до 65535
Включить экспертные настройки	Включите дополнительные параметры ведомого устройства.	Без опознавательных знаков	Отмечено Без опознавательных знаков

5. КОНФИГУРАЦИЯ

Необязательный	Объявите ведомое устройство как необязательное.	Без опознавательных знаков	Отмечено Без опознавательных знаков
Выберите округ Колумбия	Показать все конфигурации распределенных часов, представленные в файле описания устройства.		
Включить распределенные часы	Включить распределенный	Без опознавательных знаков	Отмечено Без опознавательных знаков
Единичный цикл синхронизации [мкс]	Варианты конфигурации часов.	100000	2000 до 1000000

Конфигурация устройства	Описание	Значение по умолчанию	Варианты
Включить синхронизацию 0	Включите конфигурации модуля синхронизации Sync 0.	Без опознавательных знаков	Отмечено Без опознавательных знаков
Единичный цикл синхронизации (Sync 0)	При выборе этой опции время цикла будет определяться произведением коэффициента и цикла синхронизации.	Без опознавательных знаков	Отмечено Без опознавательных знаков
Определяется пользователем (синхронизация 0)	Если выбран этот параметр, желаемое время в микросекундах можно установить непосредственно в поле «Время цикла (мкс)».	Без опознавательных знаков	Отмечено Без опознавательных знаков
Время цикла [мкс] (синхронизация 0)	Показать текущее установленное время цикла.	100000	1 to 2147483647
Время сдвига [мкс] (синхронизация 0)	Время между событиями синхронизации и временем «Действителен вывод» или «Защелка ввода».	0	До -2147483648 2147483647
Включить синхронизацию 1	Включите конфигурации модуля синхронизации Sync 1.	Без опознавательных знаков	Отмечено Без опознавательных знаков
Единичный цикл синхронизации (Sync 1)	При выборе этой опции время цикла будет определяться произведением коэффициента и цикла синхронизации.	Без опознавательных знаков	Отмечено Без опознавательных знаков

5. КОНФИГУРАЦИЯ

Определяется пользователем (синхронизация 1)	Если выбран этот параметр, желаемое время в микросекундах можно установить непосредственно в поле «Время цикла (мкс)».	Без опознавательных знаков	Отмечено Без опознавательных знаков
Время цикла [мкс] (синхронизация 1)	Показать текущее установленное время цикла.	100000	1 до 2147483647
Время сдвига [мкс] (синхронизация 1)	Время между событиями синхронизации и временем «Действителен вывод» или «Защелка ввода».	0	до -2147483648 2147483647
Проверить идентификатор поставщика	Если флажок не установлен, проверка идентификатора поставщика будет отключена.	Отмечено	Отмечено Без опознавательных знаков
Проверить идентификатор продукта	Если флажок не установлен, проверка идентификатора продукта будет отключена.	Отмечено	Отмечено Без опознавательных знаков
Доступ к SDO	Установите привязку времени (в микросекундах) для проверки тайм-аута доступа к SDO.	-	0 до 100000
I -> P	Установите контрольное время (в микросекундах) для проверки тайм-аута переключения из режима инициализации в режим подготовки к работе.	-	0 до 100000
Конфигурация устройства	Описание	Значение по умолчанию	Варианты
P -> S/S -> O	Установите временную привязку (в микросекундах) для проверки тайм-аута переключения из предоперационного в безопасный режим и из безопасного режима в рабочий режим.	-	0 до 100000
Циклическая единица	Установите единичный цикл на локальный микропроцессор.	Без опознавательных знаков	Отмечено Без опознавательных знаков
Блок-защелка 0	Установите Latch Unit 0 на локальный микропроцессор.	Без опознавательных знаков	Отмечено Без опознавательных знаков

5. КОНФИГУРАЦИЯ

Блок защелки 1	Установите блок-защелку 1 на локальный микропроцессор.	Без опознавательных знаков	Отмечено Без опознавательных знаков
----------------	--	----------------------------	--

Таблица 138: Конфигурации ведомых устройств EtherCAT

Примечания:

Адрес AutoInc: этот адрес используется только во время запуска, когда мастер назначает адреса EtherCAT ведомым устройствам. Когда, например, первая телеграмма проходит через ведомые устройства, каждое проходящее ведомое устройство увеличивает свой AutoInc на 1. Ведомое устройство с адресом 0 в конечном итоге получает данные.

Необязательно: Если ведомое устройство объявлено как необязательное, сообщение об ошибке не будет создано, если устройство не существует в шинной системе. Таким образом, адрес псевдонима станции должен быть определен и записан в EEPROM. Эта опция доступна только в том случае, если опция Autoconfig Master/Slave в настройках EtherCAT Master активирована и если эта функция поддерживается EtherCAT Slave.

Включить распределенные часы: если функция распределенных часов включена, время цикла обмена данными, отображаемое в поле Цикл единицы синхронизации (мкс), будет определяться временем основного цикла. Таким образом, главные часы могут синхронизировать обмен данными внутри сети. Настройки для работы с блоком(ами) синхронизации зависят от ведомого устройства.

Включить синхронизацию 0: Если эта опция активирована, используется модуль синхронизации Sync0. Блок синхронизации описывает набор данных процесса, которыми обмениваются синхронно.

Sync Unit Cycle (Sync 0): Если эта опция активирована, время цикла главного устройства, умноженное на выбранный коэффициент, будет использоваться как время цикла синхронизации для подчиненного устройства. В поле «Время цикла» (мкс) отображается текущее установленное время цикла.

Shift Time: Shift Time описывает время между событиями синхронизации (Sync0, Sync1) и временем действия вывода или фиксации ввода. Доступное для записи значение, если ведомое устройство поддерживает сдвиг выходного значения или входной фиксации.

Enable Sync 1: Если выбран этот параметр, используется блок синхронизации Sync1. Блок синхронизации представляет собой набор данных процесса, которыми обмениваются синхронно.

Sync Unit Cycle (Sync1): Если эта опция активирована, время основного цикла, умноженное на выбранный коэффициент, будет использоваться как время цикла синхронизации для ведомого устройства. В поле «Время цикла» (мкс) отображается текущее установленное время цикла.

Текущее состояние: отображается текущее состояние. Возможные значения: Init, Pre-Operational, Safe-Operational и Operational. Если состояние Operational, конфигурация ведомого была завершена правильно.

Проверить идентификатор поставщика и идентификатор продукта: по умолчанию при запуске системы идентификатор поставщика и/или идентификатор продукта будут проверяться на соответствие текущим настроенным параметрам. При обнаружении несоответствия шина будет остановлена, и дальнейшие действия выполняться не будут. Это позволяет избежать загрузки ошибочной конфигурации. Эта опция предназначена для отключения проверки в случае необходимости.

Доступ к SDO: по умолчанию тайм-аут для отправки списка SDO при запуске системы не установлен. Однако, если необходимо проверить, превышает ли это действие определенное время, оно должно быть определено (в микросекундах) в этом поле.

I -> P: по умолчанию тайм-аут для перехода состояния из Init в Pre-Operational не установлен. Однако, если необходимо проверить, превышает ли это действие определенное время, оно должно быть определено (в микросекундах) в этом поле.

P -> S / S -> O: По умолчанию тайм-аут для перехода состояния из Pre-Operational в Safe-Operational и из Safe-Operational в Operational не установлен. Однако, если необходимо проверить, превышает ли это действие определенное время, оно должно быть определено (в микросекундах) в этом поле.

Управление блоком циклов постоянного тока: выберите желаемую(ые) опцию(и) относительно функций распределенных часов, чтобы определить, какие из них должны быть назначены локальному микропроцессору. Управление осуществляется в регистре 0x980 подчиненного устройства EtherCAT. Возможные настройки: Cyclic Unit, Latch Unit 0, Latch Unit 1.

Псевдоним станции: эти настройки видны только в том случае, если активирована опция Необязательно или если ведомое устройство поддерживает псевдонимы адресов (определенные в файле описания устройства).

Включить: Если параметр Необязательный не активирован, этот параметр можно активировать, если он явно поддерживается описанием устройства ведомого устройства. Это позволяет напрямую назначать адрес-псевдоним, чтобы получить адрес ведомого устройства независимо от его положения на шине. Если опция Необязательно активирована, этот флажок отключен.

5. КОНФИГУРАЦИЯ

Запись в EEPROM: эта команда видна только в онлайн-режиме. Это позволяет записать определенный адрес в EEPROM ведомого устройства. Если ведомое устройство не поддерживает эту команду, эта команда не будет иметь никакого эффекта, и устройство не будет работать как дополнительное ведомое устройство.

Фактический адрес: это поле, видимое только в онлайн-режиме, отображает фактический адрес ведомого устройства. Его можно использовать для проверки успешности выполнения команды записи в EEPROM.

5.5.12.4.2. FMMU/Синхронизация

Это диалоговое окно будет отображаться на вкладке редактора конфигурации ведомого устройства EtherCAT только в том случае, если режим автонастройки в ведущем устройстве отключен. Он показывает FMMU и диспетчеры синхронизации ведомого устройства, как определено в файле описания устройства. Эти настройки могут быть переработаны, например, для настройки связи между ведомыми.

ВНИМАНИЕ

Это ЭКСПЕРТНЫЕ НАСТРОЙКИ, которые обычно не нужны для стандартных приложений.

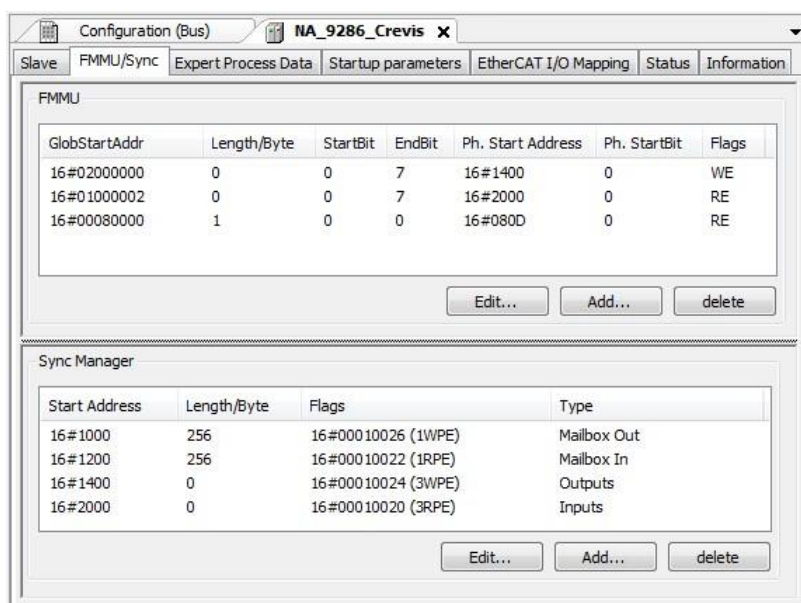


Рисунок 101: Диалоговое окно FMMU/синхронизации

5.5.12.4.3. FMMU/Синхронизация - FMMU

В этой таблице показаны блоки управления памятью Fieldbus (FMMU) ведомого устройства, которые используются для обработки данных процесса. Каждое отображение логического адреса GlobStartAddr на физический адрес Ph. Start Address определено. Возможно побитовое отображение. Новые единицы могут быть добавлены, а существующие могут быть отредактированы с помощью диалогового окна Edit FMMU, которое открывается с помощью кнопок Add и Редактировать... кнопки.

5.5.12.4.4. FMMU/Sync — диспетчер синхронизации

В этой таблице показаны диспетчеры синхронизации ведомого устройства. Для каждого доступного типа диспетчера синхронизации (входящий почтовый ящик, исходящий почтовый ящик, входы, выходы) определяются физический начальный адрес, тип доступа, буфер и физический адрес, на который должны быть отправлены прерывания. Новые диспетчеры синхронизации могут быть добавлены соответственно. существующие можно редактировать в диалоге Edit SyncMan, который открывается кнопками Add и Edit....

5.5.12.4.5. Данные процесса и экспертные данные процесса

На вкладке Process Data редактора конфигурирования EtherCAT Slave отображаются входные и выходные данные процесса подчиненного устройства, каждое из которых определяется именем, типом и индексом в файле описания устройства, как показано на рисунке ниже.

5. КОНФИГУРАЦИЯ

Выбранный вход (для чтения) и выход (для записи) устройства доступны в диалоговом окне EtherCAT Slave — I/O Mapping в качестве входов и выходов ПЛК, на которые могут быть сопоставлены переменные проекта.

Чтобы изменить текущий выбор, сначала необходимо установить флажок перед текущими выбранными данными, чтобы отменить выбор. После этого вы можете установить еще один.

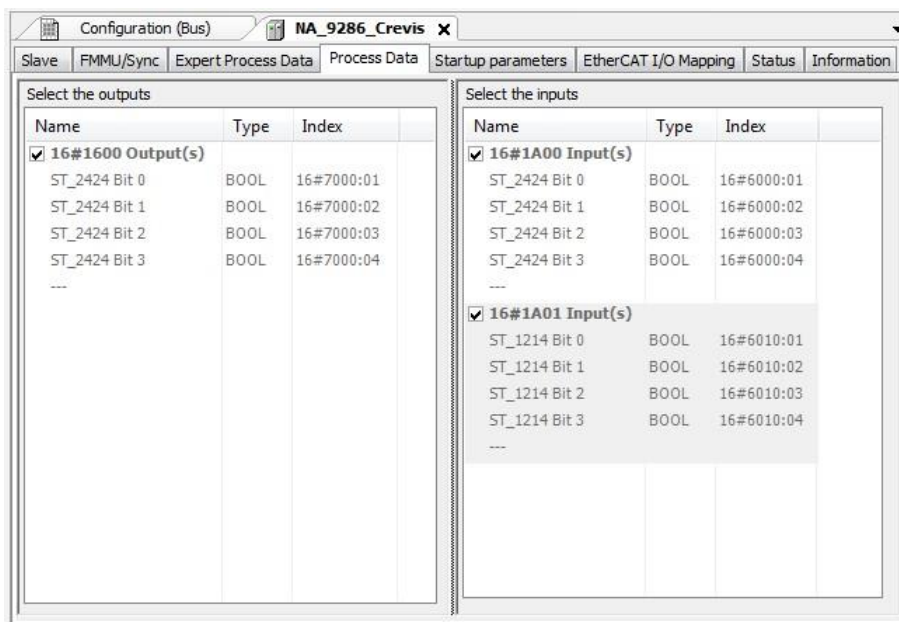


Рисунок 102: Диалоговое окно обработки данных

Диалоговое окно Expert Process Data будет доступно в редакторе конфигурации EtherCAT Slave, только если активирована опция Enable Expert Settings. Он предоставляет другое, более подробное представление о данных процесса, дополняя то, что представлено на вкладке «Данные процесса». Кроме того, в этом диалоговом окне можно активировать загрузку назначения PDO и конфигурации PDO.

ВНИМАНИЕ

Если ведомое устройство не принимает конфигурацию PDO, оно останется в предрабочем состоянии, и обмен данными в реальном времени будет невозможен.

5. КОНФИГУРАЦИЯ

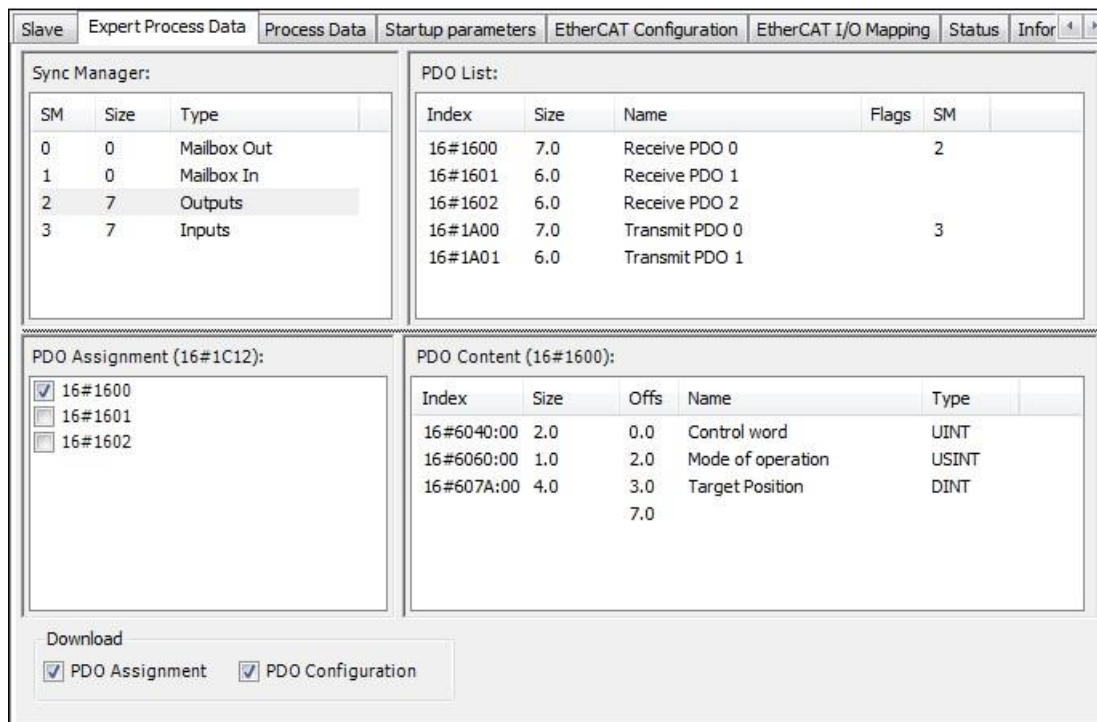


Рисунок 103: Диалоговое окно данных экспертного процесса

Это диалоговое окно разделено на четыре раздела и две опции:

Sync Manager: Список (Sync Manager) с размером данных и типом PDO.

Назначение PDO: список PDO, назначенных выбранному диспетчеру синхронизации. Флажок активирует PDO и создаются каналы ввода/вывода. Это похоже на простые окна конфигурации PDO. Здесь могут быть включены или отключены только PDO.

Список PDO: Список всех PDO, определенных в файле описания устройства. Отдельные PDO можно удалить, отредактировать или добавить, выполнив соответствующую команду из контекстного меню.

PDO Content: Отображает содержимое PDO, выбранного в разделе выше. Записи можно удалять, редактировать или добавлять, выполнив соответствующую команду из контекстного меню.

Назначение PDO: если активировано, команда записи CoE будет добавлена к индексу 0x1CXX для записи конфигурации PDO 0x16XX или 0x1A00.

Конфигурация PDO: если активировано, будет добавлено несколько команд записи CoE для записи отображения PDO на ведомое устройство.

ВНИМАНИЕ

Если ведомое устройство не поддерживает конфигурацию PDO, загрузка может привести к ошибке ведомого. Эта функция должна использоваться только экспертами.

5.5.12.4.6. Данные процесса и экспертные данные процесса — редактирование списка PDO

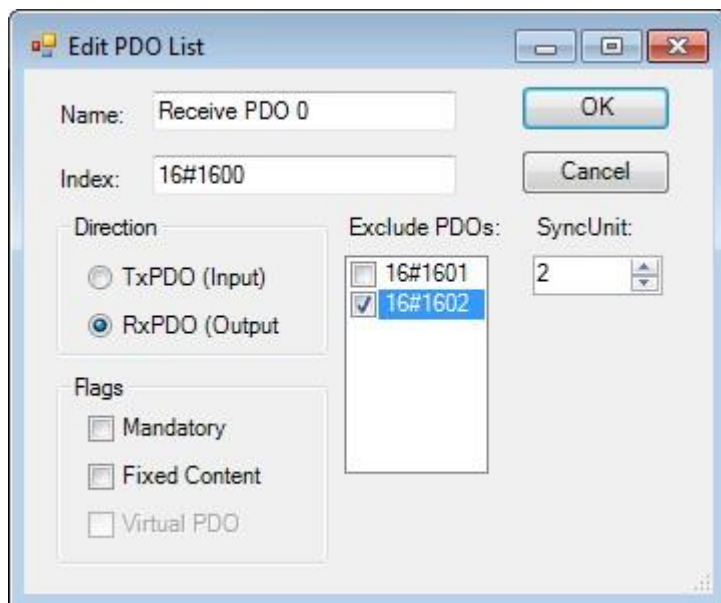


Рисунок 104: Диалоговое окно редактирования списка PDO

Это диалоговое окно открывается через контекстное меню из области списка PDO, представленной на рисунке 103. Ниже приведены некоторые пояснения по параметрам конфигурации, представленным в этом диалоговом окне.

Имя: Имя записи PDO.

Индекс: индекс редактируемого PDO.

TxPDO (вход): если активировано, PDO будет передаваться от ведущего к ведомому.

RxPDO (выход): если активировано, PDO будет передан от ведомого к ведущему.

Обязательно: PDO необходим и не может быть снят в области назначения PDO.

Фиксированное содержимое: содержимое PDO является фиксированным и не может быть изменено. Невозможно добавлять записи на панель содержимого PDO.

Виртуальный PDO: зарезервирован для использования в будущем.

Исключить PDO: можно определить список PDO, которые могут или не могут быть выбраны вместе с редактируемым PDO в области назначения PDO или на вкладке Process Data. Если объект PDO отмечен в этом списке, его нельзя выбрать, он становится серым в области назначения PDO, когда выбран объект PDO в редакции. SyncUnit: идентификатор PDO Sync Manager, которому должен быть назначен.

5.5.12.4.7. Данные процесса и экспертные данные процесса — определение содержимого PDO

Доступ к этому диалоговому окну осуществляется через контекстное меню в области содержимого PDO, и его содержимое, помимо возможности доступа к этим окнам, зависит от используемого ведомого устройства EtherCAT..

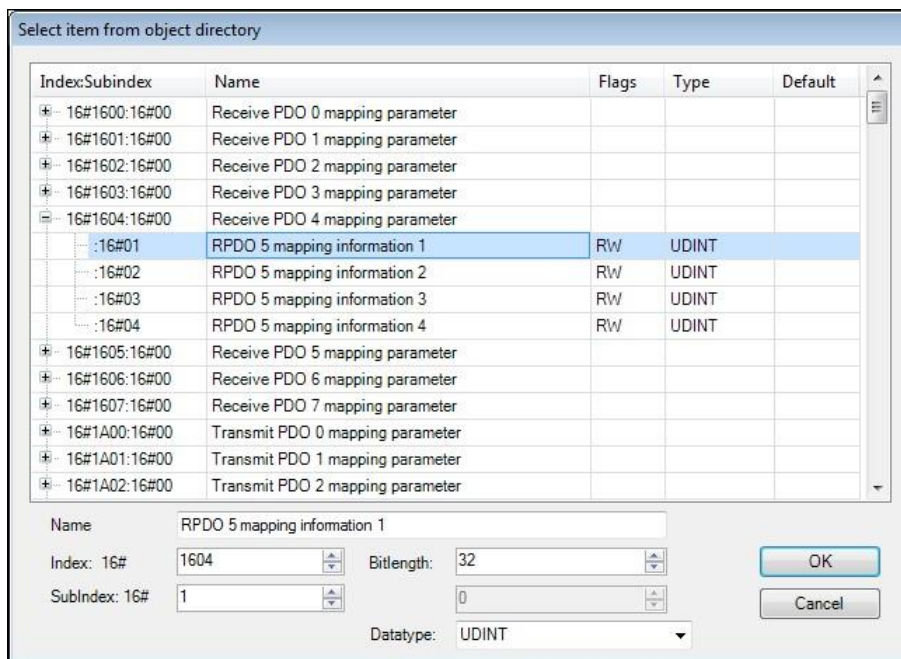


Рисунок 105: Выберите элемент из диалогового окна каталога объектов

5.5.12.4.8. Параметры запуска

На вкладке Startup Parameters можно определить параметры для устройства, которые будут передаваться SDO (Service Data Objects) или IDN при запуске системы. Параметры, доступные на этой вкладке, а также возможности доступа зависят от используемого подчиненного устройства EtherCAT и представлены в файле описания устройства.

5.5.12.4.9. В сети

Диалог Онлайн доступен только в редакторе конфигурации EtherCAT Slave, если активен параметр Включить экспертные настройки ведомого устройства и Приложение подключено к устройству. Он обеспечивает просмотр информации о состоянии ведомого устройства и функции для передачи файлов на ведомые устройства по EtherCAT (FoE).

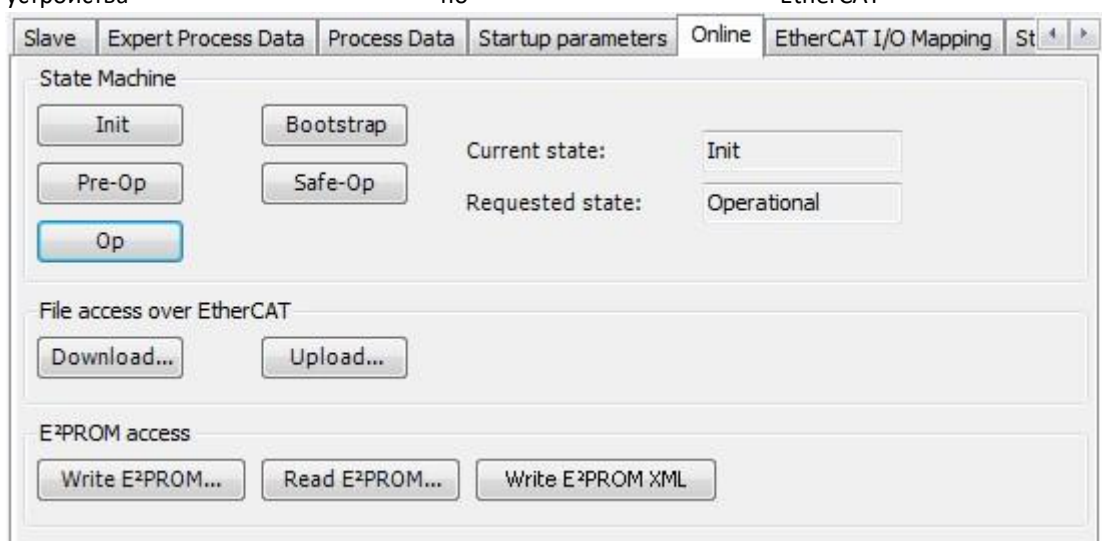


Рисунок 106: Онлайн диалог

Эта вкладка разделена на следующую функциональную группу:

State Machine: Кнопки Init, Pre-Op (предоперационный), Op (операционный) и Safe-Op (безопасный рабочий) можно использовать для целей отладки. Они делают ведомый переход в соответствующее состояние.

Доступ к файлам через EtherCAT: Если вы хотите передать файлы встроенного программного обеспечения на ведомое устройство или с него, вам нужно нажать кнопку Bootstrap, чтобы переключить ведомое устройство в режим начальной загрузки. Загрузка и загрузка файлов прошивки осуществляется соответствующими кнопками. Откроется диалоговое окно для сохранения или открытия файла прошивки. В этом диалоговом окне для выполнения передачи файла требуется строка и пароль. Эта информация предоставляется ведомым устройством и задокументирована в паспорте ведомого устройства.

Доступ к E2PROM: конфигурация ведомого устройства может быть прочитана или записана в E2PROM. Здесь, как и при переносе прошивки, откроется диалоговое окно для открытия или сохранения файлов. Команду Write E2PROM XML можно использовать для записи конфигурации Slave непосредственно из XML-файла в устройство. Эта команда доступна только при наличии данных конфигурации в файле XML (раздел <ConfigData>).

5.5.12.4.10. Ведомое устройство EtherCAT — сопоставление ввода/вывода

На этой вкладке редактора конфигурации EtherCAT Slave можно назначить переменные проекта входам или выходам EtherCAT. Таким образом, пользовательское приложение может управлять переменными EtherCAT Slave.

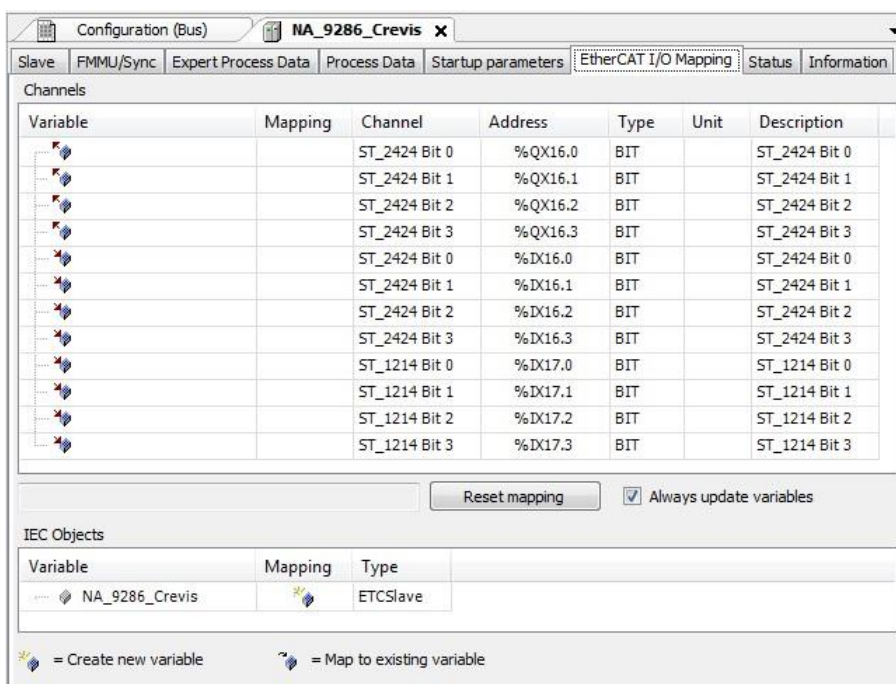


Рисунок 107: Диалоговое окно отображения ведомых входов/выходов

5.5.12.4.11. Вкладки «Статус» и «Информация»

На вкладке «Статус» ведомого устройства EtherCAT отображается информация о состоянии (например, «Работает», «Остановлен») и диагностические сообщения для конкретного устройства, а также на используемой плате и внутренней шинной системе.

Вкладка «Информация», представленная в редакторе конфигурации EtherCAT Slave, показывает, если она доступна, следующую общую информацию о модуле: Имя, Производитель, Тип, Версия, Категории, Номер заказа, Описание, Изображение.

5.5.13. Этернет/IP

EtherNet/IP — это протокол архитектуры ведущий-ведомый, который состоит из сканера EtherNet/IP (главного) и одного или нескольких адаптеров EtherNet/IP (ведомых). Редактор EtherNet/IP предоставляет диалоговые окна для настройки параметров и сопоставления входов/выходов с переменными.

5. КОНФИГУРАЦИЯ

Протокол Ethernet/IP основан на CIP (общем промышленном протоколе), который имеет две основные цели: передача данных, ориентированных на управление, связанных с устройствами ввода-вывода, и передача другой информации, относящейся к управляемой системе, такой как конфигурация параметров и диагностика. Первый выполняется с помощью неявных сообщений, а второй — с помощью явных сообщений.

Их система выполнения может действовать как сканер или адаптер. NET-интерфейс каждого ЦП поддерживает только один экземпляр EtherNet/IP, и его нельзя разместить на модуле расширения Ethernet.

Экземпляр адаптера EtherNet/IP поддерживает до 64 модулей ввода или вывода, ограниченных 505 байтами на входе и 509 байтами на выходе. Эти модули могут быть BYTE, WORD, SINT, INT, DINT, DWORD или REAL. Диапазон MainTask устройства, работающего как адаптер, должен быть меньше или равен RPI.

ВНИМАНИЕ

EtherNet/IP нельзя использовать вместе с Ethernet NIC-Teaming или с резервированием Half-Cluster.

ВНИМАНИЕ

Для EtherNet/IP требуется циклическая основная задача, а это означает, что ее нельзя использовать с какой-либо конфигурацией, которая устанавливает основную задачу как независимую. Например, проект Basic Profile не поддерживает EtherNet/IP.

ВНИМАНИЕ

Чтобы избежать проблем со связью, EtherNet/IP Scanner может иметь адаптеры, настроенные только в одной и той же подсети.

5.5.13.1. EtherNet/IP

Чтобы добавить сканер или адаптер EtherNet/IP, необходимо добавить интерфейс EtherNet/IP под интерфейсом NET. Это можно сделать через команду Add Device. В этот интерфейс EtherNet/IP можно добавить сканер или адаптер.

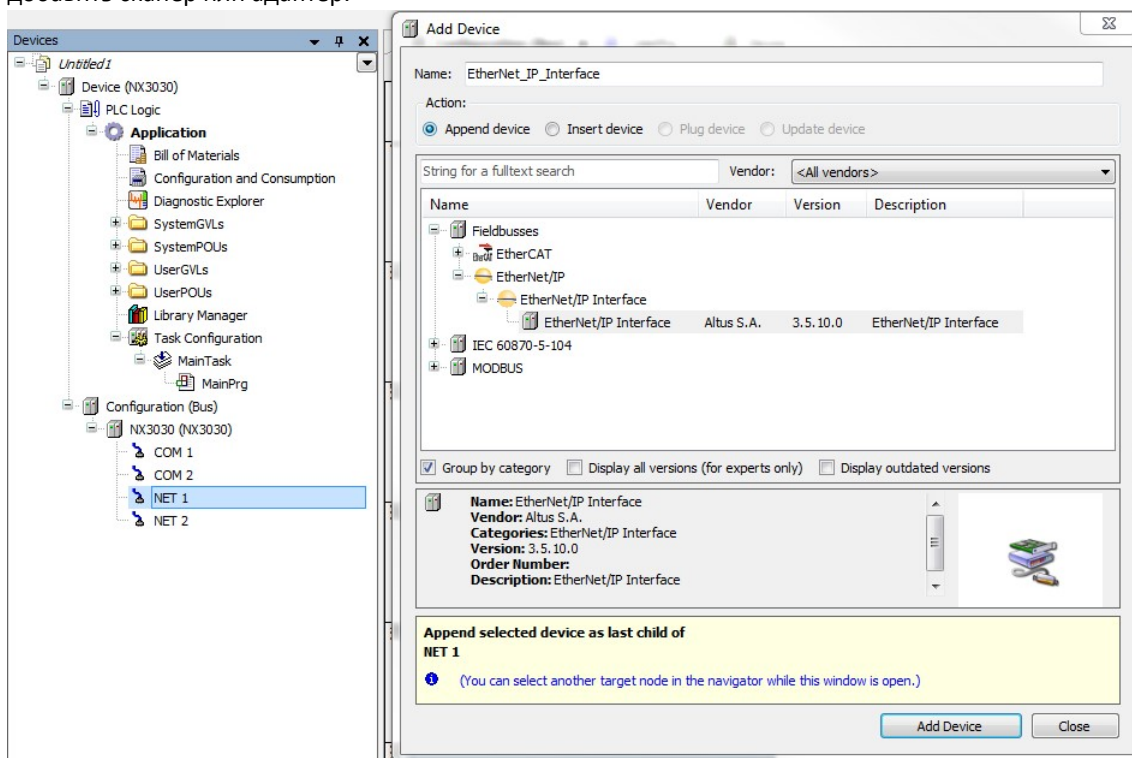


Рисунок 108: Добавление EtherNet/IP Interface

5. КОНФИГУРАЦИЯ

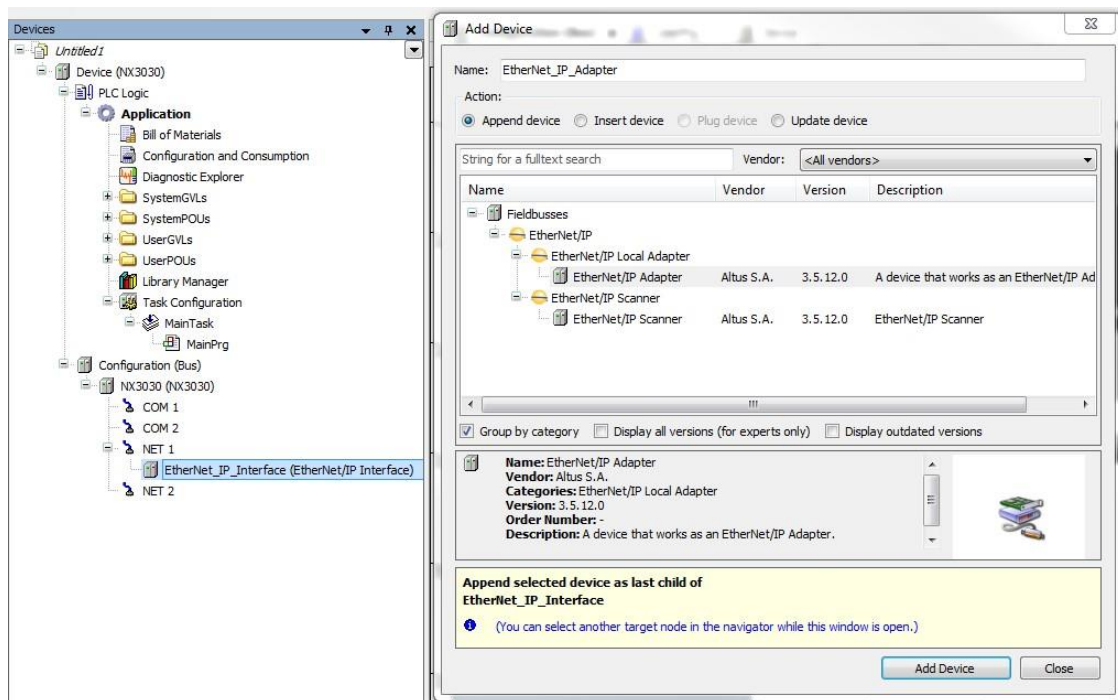


Рисунок 109: Добавление адаптера EtherNet/IP или сканера

5.5.13.2. Конфигурация сканера EtherNet/IP

Сканеру требуется как минимум один адаптер, с которым он будет обмениваться данными. Новые адаптеры можно установить на MasterTool с помощью файлов EDS и DCF. Параметры конфигурации могут отличаться в зависимости от файла описания устройства добавленного адаптера.

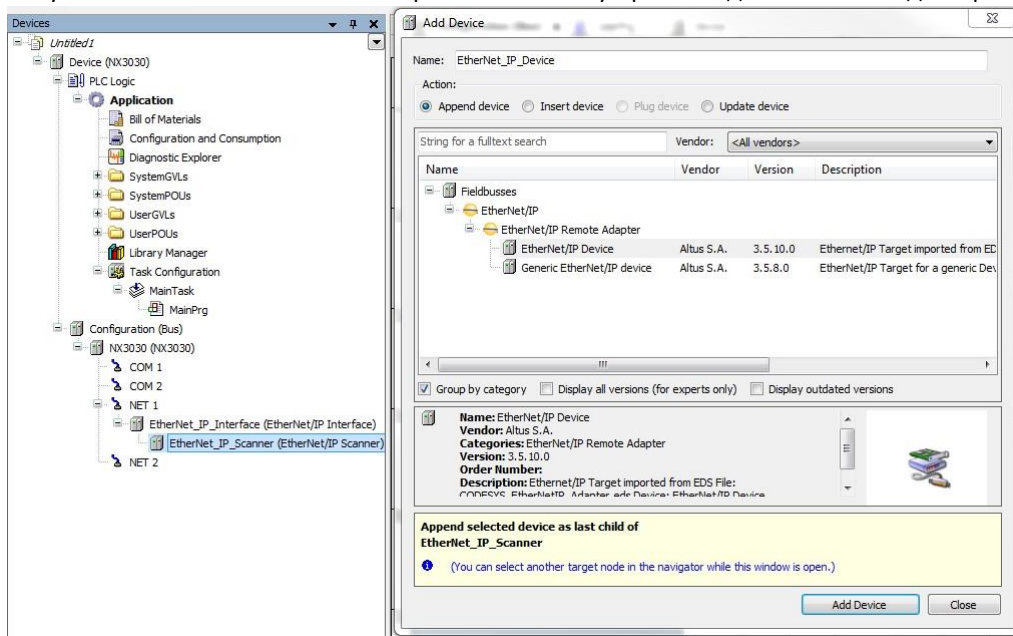


Рисунок 110: Добавление адаптера EtherNet/IP под сканер

5.5.13.2.1. Основная информация

5. КОНФИГУРАЦИЯ

После открытия адаптера, объявленного под сканером, его можно настроить по мере необходимости. Первая вкладка — «Общие», на ней можно настроить IP-адрес и параметры электронного ключа. Эти параметры должны быть отмечены или сняты, если используемый адаптер установлен на MasterTool. В противном случае, если используется тип адаптера Generic, то поля Проверить тип устройства, Проверить идентификатор поставщика, Проверить код продукта, Проверить основную версию и Проверить дополнительную версию должны быть заполнены правильной информацией, а флажки установлены столько, сколько необходимо. Проверка может быть переключена с проверки совместимости на строгую проверку личности.

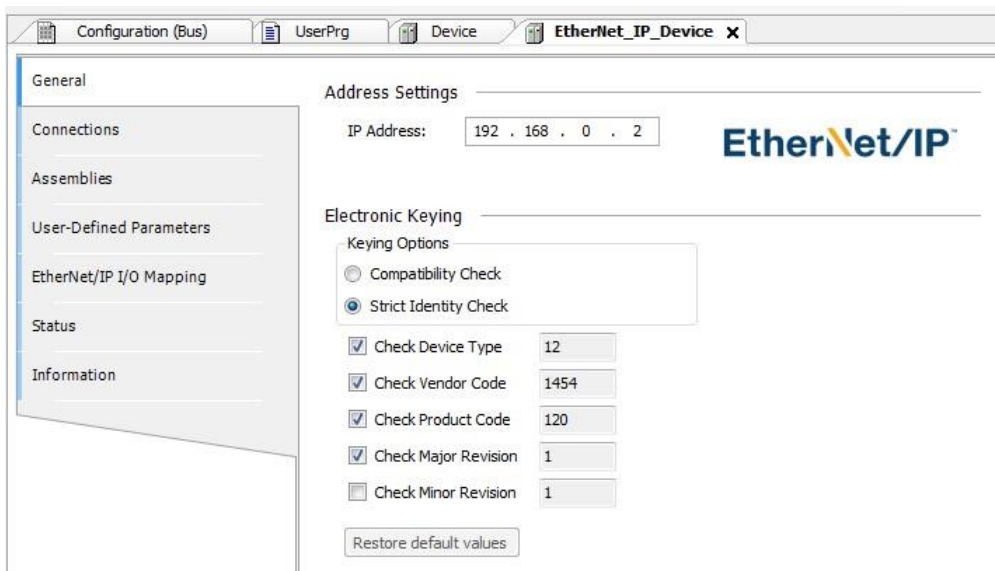


Рисунок 111: Вкладка «Общие» EtherNet/IP

5.5.13.2.2. Соединения

В верхней части вкладки «Подключения» отображается список всех настроенных подключений. Когда в файле EDS есть соединение с монопольным владельцем, оно вставляется автоматически при добавлении адаптера. Данные конфигурации для этих подключений можно изменить в нижней части экрана.

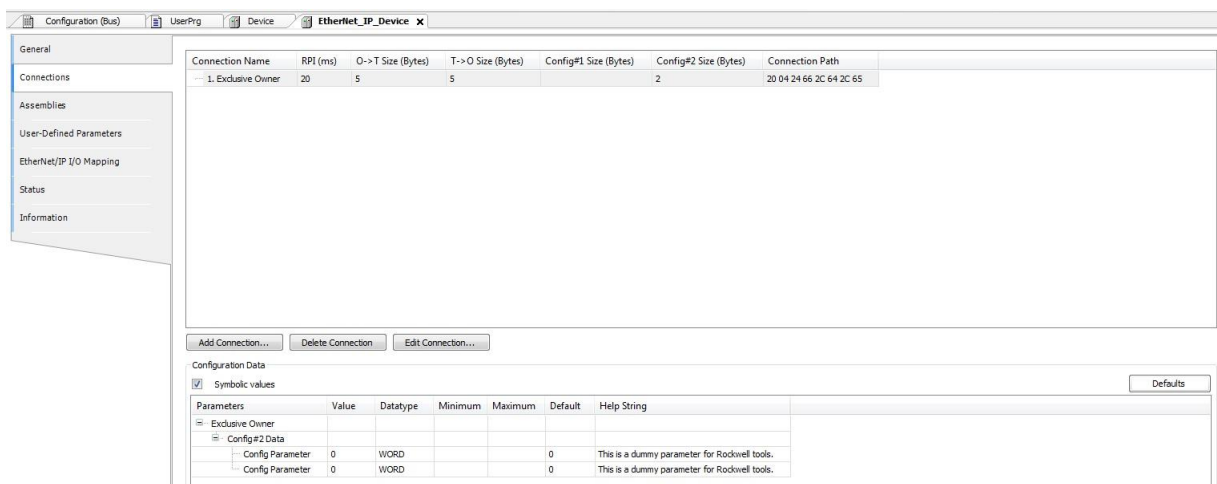


Рисунок 112: Вкладка «Соединение EtherNet/IP»

Примечания:

Для подключения двух или более сканеров EtherNet/IP к одному и тому же удаленному адаптеру:

1. Только один из Сканеров может установить соединение с монопольным владельцем.

5. КОНФИГУРАЦИЯ

2. То же значение RPI (мс) должно быть настроено для сканеров.

Данные конфигурации определяются в файле EDS. Данные передаются на удаленный адаптер при открытии соединения.

Конфигурация	Описание	Значение по умолчанию	Варианты
RPI (мс)	Request Packet Interval: интервал обмена входными и выходными данными.	10 мс	Кратность интервала MainTask
О -> Размер Т (байты)	Размер данных производителя от сканера до адаптера (оригинатор -> цель)	Зависит от ЭЦП адаптера	0 - 400
Т -> О Размер (байты)	Размер потребительских данных от адаптера до сканера (Т -> О)	Зависит от ЭЦП адаптера	0 - 400
Размер конфигурации №1 (байты)	Размер данных конфигурации 1.	Зависит от ЭЦП адаптера	-
Конфигурация № 2 Размер (байты)	Размер данных конфигурации 2.	Зависит от ЭЦП адаптера	-
Путь подключения	Адрес объектов конфигурации - объекты ввода, объекты вывода.	Зависит от ЭЦП адаптера	-

Таблица 139: Параметры соединения EtherNet/IP

Для добавления новых подключений есть кнопка Добавить подключение..., которая открывает окно Новое подключение. В этом окне можно настроить новый тип подключения из predeterminedных на ЭЦП адаптера или универсальное подключение с нуля.

Рисунок 113: Окно нового подключения EtherNet/IP

5.5.13.2.3. Сборки

В верхней части вкладки «Сборки» отображается список всех настроенных соединений. Когда соединение выбрано, связанные входы и выходы отображаются в нижней части вкладки.

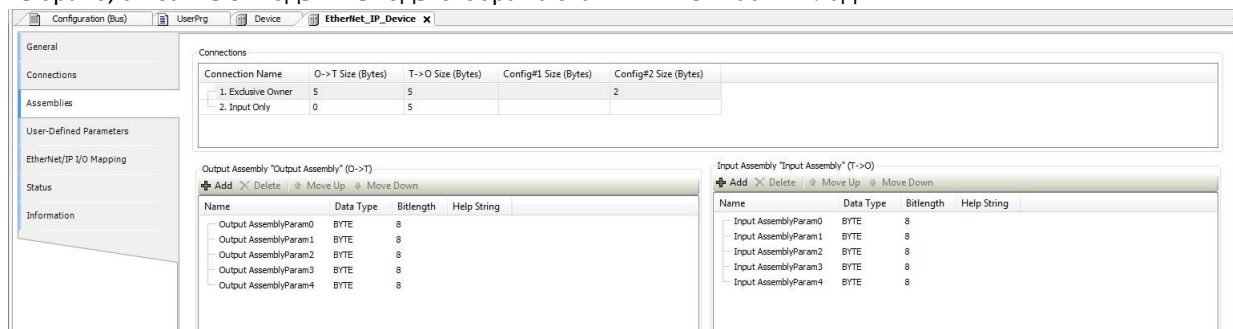


Рисунок 114: Сборки EtherNet/IP

Выходная сборка и входная сборка:

Конфигурация	Описание
Добавить	Открывает диалоговое окно «Добавить ввод/вывод».
Удалить	Удаляет все выбранные входы/выходы.
Конфигурация	Описание
Вверх	Перемещает выбранный вход/выход в списке.
Вниз	Порядок в списке определяет порядок отображения ввода/вывода.
Имя	Эти значения можно изменить, дважды щелкнув текстовое поле.
Строка справки	
Длина бита	Это значение нельзя редактировать..

Табл. 140: Вкладка "Сборки EtherNet/IP"

Диалоговое окно Добавить ввод/вывод:

Конфигурация	Описание
Имя	Имя входа/выхода, которое необходимо вставить.
Тип данных	Тип вставляемого входа/выхода. Этот тип также определяет его битовую длину.

Табл. 141: Окно EtherNet/IP «Добавить ввод/вывод»

5.5.13.2.4. Сопоставление ввода-вывода EtherNet/IP

На вкладке I/O Mapping в столбце «Переменная» отображается имя автоматически сгенерированного экземпляра адаптера в разделе «Объекты IEC». Таким образом, приложение может получить доступ к экземпляру. Здесь переменные проекта сопоставляются с входами и выходами адаптера. Опция Всегда обновлять переменные должна быть сохранена по умолчанию в Включить 1.

5.5.13.3. Конфигурация адаптера EtherNet/IP

Для адаптера EtherNet/IP требуются модули Ethernet/IP. Модули будут предоставлять сопоставления ввода-вывода, которыми пользовательское приложение может управлять через адреса %I или %Q в соответствии с его конфигурацией (ВХОДНОЙ БАЙТ, ВЫХОДНОЙ БАЙТ и т. д.).

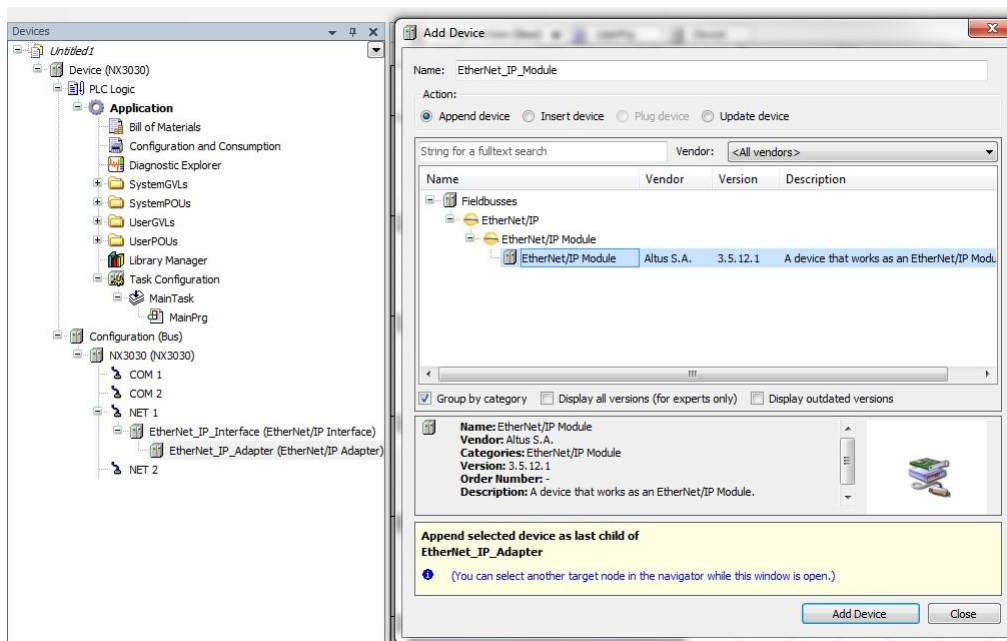


Figure 115: Adding an EtherNet/IP Module under the Adapter

5.5.13.3.1. Module Types

There are 18 different modules which can be added under the adapter. Nine outputs and Nine inputs. They are of type BYTE, WORD, DWORD, REAL, SINT, INT, DINT and BIG. These types can be chosen in the *General* tab of the module.

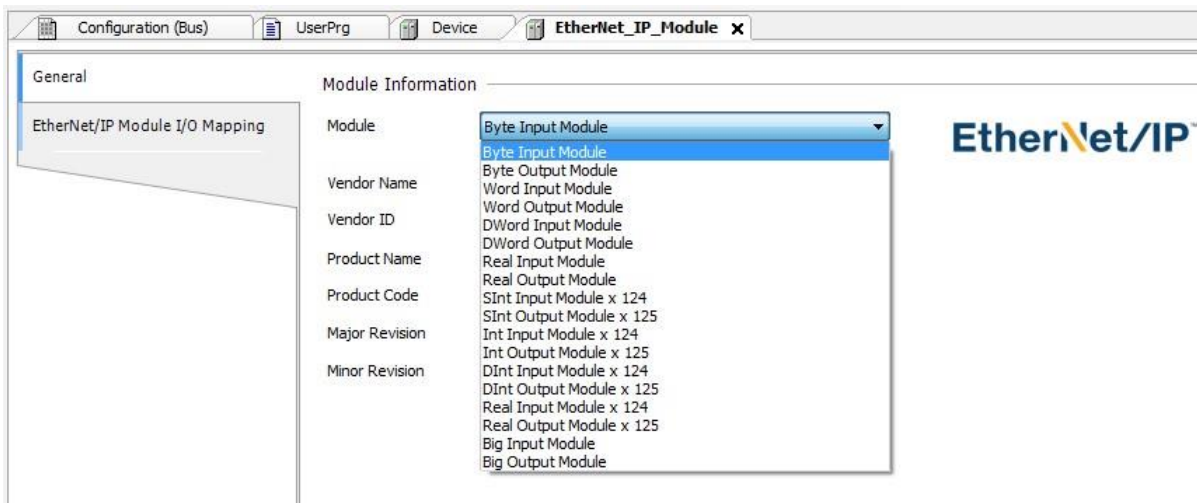


Рисунок 116: Тип модуля EtherNet/IP

5.5.13.3.2. Сопоставление входов/выходов модуля EtherNet/IP

В столбце «Переменная» отображается имя автоматически сгенерированного экземпляра модуля в разделе «Объекты IEC». Таким образом, приложение может получить доступ к экземпляру. Параметр «Всегда обновлять переменные» должен оставаться по умолчанию в разделе «Включить».

5.5.14. Сервер МЭК 60870-5-104

При выборе этой опции в MasterTool ЦП становится коммуникационным сервером IEC 60870-5-104, что позволяет подключать до трех клиентских устройств. Каждому клиенту драйверу принадлежит одна эксклюзивная очередь событий со следующими функциями:

Размер: 1000 событий

Сохраняемость: не сохраняет

Политика переполнения: сохраняйте самые новые

Для настройки этого протокола необходимо выполнить следующие шаги:

Добавьте экземпляр сервера протокола IEC 60870-5-104 в один из доступных каналов Ethernet. Чтобы реализовать эту процедуру, обратитесь к разделу «Вставка экземпляра протокола».

Настройте интерфейс Ethernet. Чтобы реализовать эту процедуру, обратитесь к разделу Конфигурация интерфейсов Ethernet.

Настройте общие параметры протокола IEC 60870-5-104 Сервер с режимом подключения Порт или IP и номер порта TCP, если выбран режим подключения IP

Добавляйте и настраивайте устройства, определяя нужные параметры

Добавьте и настройте сопоставления IEC 60870-5-104, указав имя переменной, тип объекта, адрес объекта, размер, диапазон, зону нечувствительности и тип зоны нечувствительности.

Настройте параметры канального уровня, указав адреса, тайм-ауты связи и параметры связи.

Настройте параметры прикладного уровня, конфигурацию синхронизации, команды, а также режим передачи объектов Integrated Totals.

Описание каждой конфигурации приведено ниже в этом разделе.

5.5.14.1. Тип данных

В таблице ниже показаны типы переменных, поддерживаемые ЦП серии Nexto для каждого типа данных протокола IEC 60870-5-104..

Тип объекта	Тип переменных МЭК
Информация об одной точке (M_SP_NA)	BOOL
	BIT
Информация о двойной точке (M_DP_NA)	DBP
Информация о позиции шага (M_ST_NA)	USINT
Измеренное значение, нормализованное значение (M_ME_NA)	INT
Измеренное значение, масштабированное значение (M_ME_NB)	INT
	INT
	UINT
Измеренное значение, короткое значение с плавающей запятой (M_ME_NC)	DINT
	UDINT
	REAL

5. КОНФИГУРАЦИЯ

Интегрированные итоги (M_IT_NA)	INT
	DINT
Одна команда (C_SC_NA)	BOOL
	BIT
Двойная команда (C_DC_NA)	DBP
Команда шага регулирования (C_RC_NA)	DBP
Команда установки точки, нормализованное значение (C_SE_NA)	INT
Команда установки точки, масштабированное значение (C_SE_NB)	INT
Команда настройки точки, короткое значение с плавающей запятой (C_SE_NC)	REAL

Таблица 142: Объявление переменных согласно IEC 60870-5-104

Примечания:

Команда шага регулирования: Нижний и Высший состояния объекта C_RC_NA связаны соответственно с состояниями внутреннего типа OFF и ON внутреннего типа DBP.

Информация о позиции шага: в соответствии с пунктом 7.3.1.5 стандарта IEC 60870-5-101 эта 8-битная переменная состоит из двух полей: значения (определяется 7 младшими битами) и переходного процесса (определяется как старший бит, который указывает, когда измеряемое устройство находится в состоянии перехода).

Ниже приведен пример кода для манипулирования полями в переменной типа USINT. Внимание, потому что этот код не работает, если значение находится внутри диапазона, поэтому эта согласованность остается на усмотрение пользователя.

```
PROGRAM UserPrg VAR usiVTI: USINT; // Значение с индикацией переходного состояния, отображаемое для клиента siValue: SINT; // Значение для преобразования в VTI. Должно быть между -64 и +63 bTransient: BOOL; // Transient to be converted to VTI END_VAR
```

```
usiVTI := SINT_TO_USINT(siValue) AND 16#3F; IF siValue < 0  
THEN usiVTI := usiVTI OR 16#40;  
END_IF  
IF bTransient THEN usiVTI := usiVTI OR  
16#80;  
END_IF
```

PROFIBUS: За исключением цифровых объектов, типы данных аналоговых объектов и счетчиков протокола IEC 60870-5-104 отличаются от типов данных модулей аналоговых и счетчиков PROFIBUS, поскольку невозможно напрямую сопоставить такие типы переменных PROFIBUS с IEC 60870-5-104. клиенты.

В этих случаях необходимо создать промежуточную переменную, которая будет отображена в клиенте IEC 60870-5-104, и должным образом преобразовать типы, как можно наблюдать в приведенном ниже примере кода.

```

PROGRAM UserPrg
VAR
  iAnalogIn: INT; iAnalogOut:
  INT; diCounter: DINT;
END_VAR

// Преобразование аналогового входа из WORD (PROFIBUS) в INT (IEC104) iAnalogIn:=
WORD_TO_INT(wNX6000in00);

// Преобразование аналогового выхода из INT(IEC104) в WORD (PROFIBUS) wNX6100out00:=
INT_TO_WORD(iAnalogOut);

// Преобразование счетчика из WORDs high+low (PROFIBUS) в DINT (IEC104) diCounter:=
DWORD_TO_DINT(ROL(WORD_TO_DWORD(wNX1005cnt00H), 16) OR wNX1005cnt00L );

```

5.5.14.2. Двойные точки

Двойные цифровые точки используются для обозначения положения оборудования, например, клапанов, автоматических выключателей и секционных секций, где переход между открытыми и закрытыми состояниями требует определенного времени. Таким образом, может указывать на промежуточное переходное состояние между обоими конечными состояниями.

Двойные цифровые точки также используются в качестве выходов и, аналогично, необходимо держать один из выходов включенным в течение определенного времени, чтобы завершить переход. Такое срабатывание осуществляется импульсами, также известными как команды отключения/включения, с определенной длительностью (достаточной для переключения управляемого устройства).

Обратитесь к разделу «Двойные точки» Руководства по использованию для получения информации о двойных цифровых точках через тип данных DBP.

После того, как цифровые модули ввода и вывода серии Nexto не поддерживают отображение точек ДАД, для того, чтобы это стало возможным, потребуются некоторые хитрости приложения. Помните, что также невозможно использовать функцию PulsedCommand, определенную в библиотеке LibRtuStandard, для управления цифровыми двойными точками серии Nexto.

5.5.14.2.1. Двойные точки цифрового ввода

Для цифровых модулей ввода необходимо объявление двух вспомогательных переменных, которые должны быть отображены на цифровом модуле ввода, помимо двойной точки, которую необходимо отобразить на сервере::

- The double point value Переменная значения
- Переменная значения простой точки OFF/TRIP: тип BOOL
- Переменная значения простой точки ON/CLOSE: тип BOOL



Рисунок 117: Пример объявления переменных Double Point

После объявления переменных необходимо создать связь между переменными-значениями и качеством цифрового модуля ввода через вкладку CPU Internal Points:

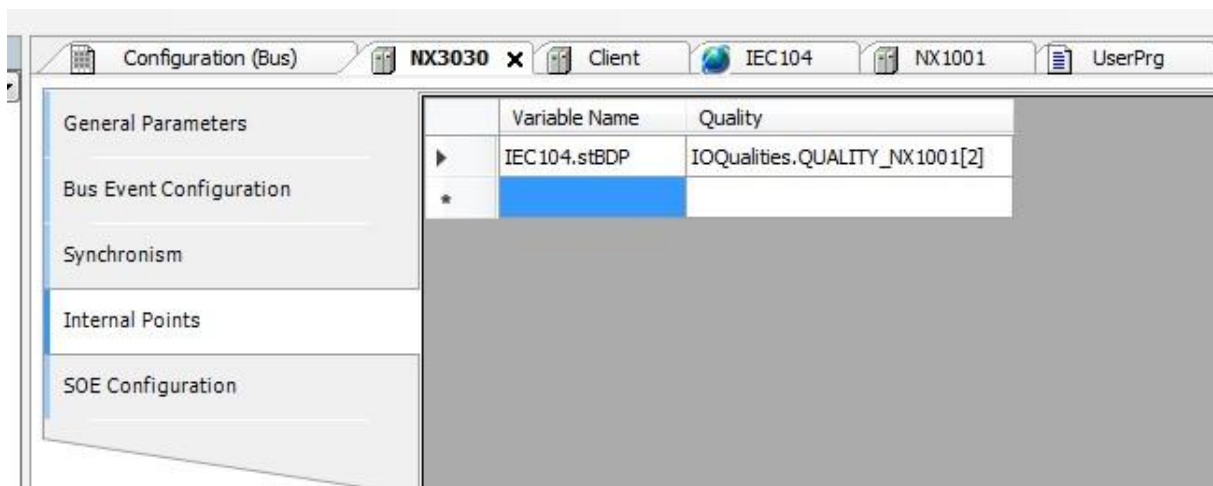


Рис. 118. Отнесение переменных Double Point к внутренним точкам

Переменная со значением двойной точки должна быть отображена в драйвере сервера IEC 60870-5-104, а обе простые переменные — в цифровом модуле ввода серии Nexto (в этом примере NX1001). Обычно состояние ВЫКЛ (ОТКЛЮЧЕНИЕ) отображается на четный вход, а состояние ВКЛ (ЗАКРЫТ) на нечетный вход.

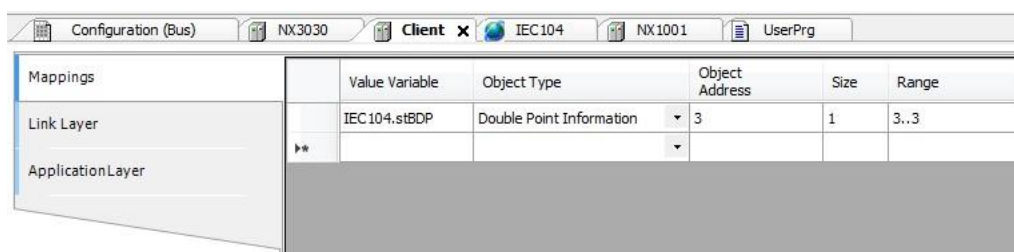


Рисунок 119: Отображение двухточечных переменных на клиенте IEC 60870-5-104

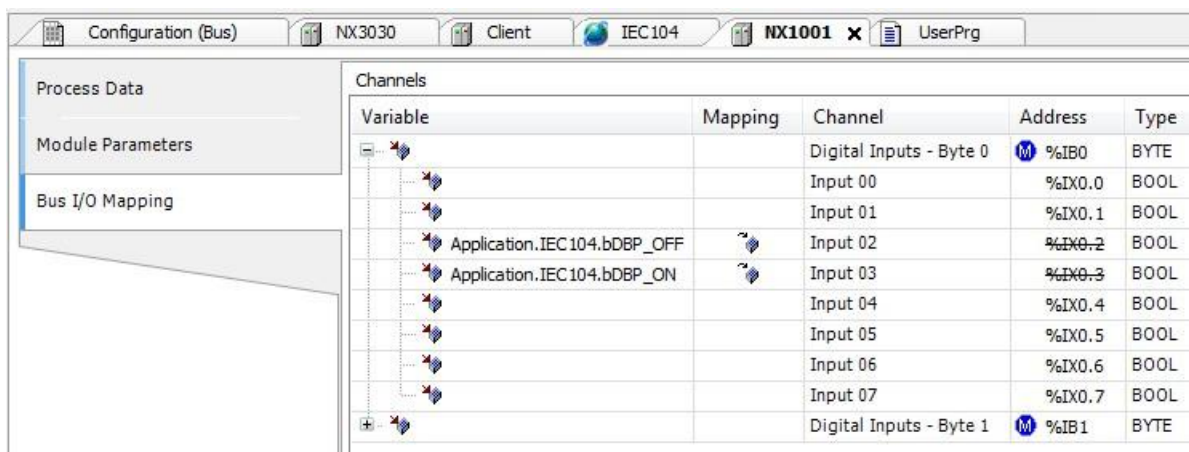


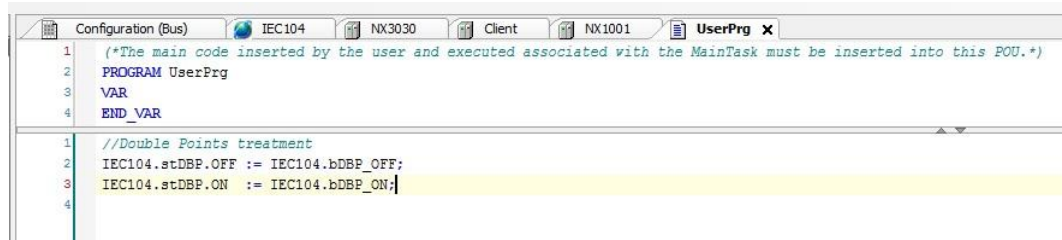
Рисунок 120: Отображение переменных на входах модуля

Наконец, пользователь должен вставить в свое приложение две строки кода, которые будут циклически выполняться, для присвоения значения простой переменной двойной точке:

Переменная значения ДАД, индекс ВКЛ, получение значения простой точки ВКЛ

5. КОНФИГУРАЦИЯ

Переменная значения ДАД, индекс ВЫКЛ, получение значения простой точки ВЫКЛ



```
1 (*The main code inserted by the user and executed associated with the MainTask must be inserted into this POU.*)
2 PROGRAM UserPrg
3 VAR
4 END_VAR

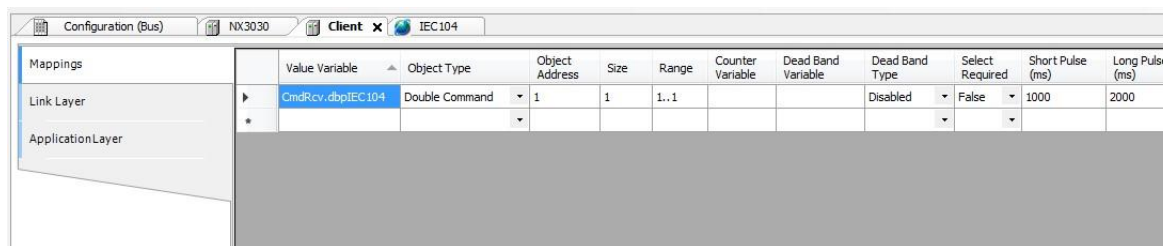
1 //Double Points treatment
2 IEC104.stDBP.OFF := IEC104.bDBP_OFF;
3 IEC104.stDBP.ON := IEC104.bDBP_ON;
```

Рисунок 121: Присвоение значений переменных двойной точке

5.5.14.2.2. Двойные точки цифрового выхода

Для цифровых модулей вывода необходимо использовать функциональный блок CommandReceiver для перехвата команд активации двойных точек, поступающих от клиентов IEC 60870-5-104. Дополнительную информацию см. в разделе «Перехват команд, поступающих из Центра управления».

Приведенный ниже пример кода, POU CmdRcv, обрабатывает импульсные команды, полученные от клиентов, для цифровой двойной точки, отображенной в модуле NX2020. Помимо следующего кода ST, необходимо сопоставить точку DBP на сервере Nexto IEC 60870-5-104.



Value Variable	Object Type	Object Address	Size	Range	Counter Variable	Dead Band Variable	Dead Band Type	Select Required	Short Pulse (ms)	Long Pulse (ms)
CmdRcv.dbpIEC104	Double Command	1	1	1..1			Disabled	False	1000	2000

Рисунок 122: Отображение переменных двойной точки цифрового выхода на клиенте IEC 60870-5-104

PROGRAM CmdRcv

VAR

CmdReceive: CommandReceiver; // Экземпляр интерцептора

5. КОНФИГУРАЦИЯ

```
fbPulsedCmd: PulsedCommandNexto; // Экземпляр импульсной команды byResult:
    BYTE; // Результат импульсной команды dbpIEC104:    DBP; //
Переменная, отображаемая в IEC 104 bSetup:    BOOL:= TRUE; // Первоначальная
настройка интцептора END_VAR

// Выполняет конфигурацию функции в первом цикле
IF bSetup THEN
CmdReceive.dwVariableAddr:= ADR(dbpIEC104);
CmdReceive.bExec:= TRUE;
CmdReceive.eCommandResult:= COMMAND_RESULT.NONE;
CmdReceive.dwTimeout:= 256 * 10; bSetup:= FALSE; END_IF

// В случае захвата команды:
IF CmdReceive.bCommandAvailable THEN

// Обрабатывает каждую из возможных команд CASE

CmdReceive.sCommand.eCommand OF
COMMAND_TYPE.NO_COMMAND:

    // Сообщаем, что есть недопустимая команда.
    // Ничего не делает и должен двигаться дальше по тайм-ауту.

COMMAND_TYPE.SELECT:

    // Обрабатывает только команды для двойных точек
    IF CmdReceive.sCommand.sSelectParameters.sValue.eParamType =
        DOUBLE_POINT_COMMAND THEN
        // Возвращает успешно завершённую команду //
        (управляется протоколом IEC104) byResult:= 7;
    ELSE
        // Команда возврата не поддерживается
    byResult:= 1; END_IF COMMAND_TYPE.OPERATE:

    // Обрабатывает только команды для двойных точек
    IF CmdReceive.sCommand.sOperateParameters.sValue.eParamType =
        DOUBLE_POINT_COMMAND THEN
        // Генерация импульсов на выходах
        IF CmdReceive.sCommand.sOperateParameters.sValue.sDoublePoint.bValue THEN
            // Выполняет функцию TRIP fbPulsedCmd( byCmdType:= 101, byPulseTime:=
                DWORD_TO_BYTE(CmdReceive.sCommand.sOperateParameters.
                    sValue.sDoublePoint.sPulseConfig.dwOnDuration/10), ptDbpVarAdr:= ADR(dbpIEC104),
                    stQuality:= IOQualities.QUALITY_NX2020[4], byStatus=> byResult);
        ELSE
            // Выполняет функцию ЗАКРЫТЬ
```

```

        fbPulsedCmd( byCmdType:= 102, byPulseTime:=
            DWORD_TO_BYTE(CmdReceive.sCommand.sOperateParameters.
                sValue.sDoublePoint.sPulseConfig.dwOffDuration/10), ptDbpVarAdr:=
                ADR(dbpIEC104), stQuality:= IOQualities.QUALITY_NX2020[5], byStatus=>
                byResult); END_IF
    ELSE
        // Команда возврата не поддерживается
byResult:= 1; END_IF COMMAND_TYPE.CANCEL:

        // Возвращает успешно завершенную команду //
        (управляется протоколом IEC104) byResult:= 7;

END_CASE

// Обрабатывает результат импульсной командной функции
// и формирует ответ на перехваченную команду
CASE byResult OF
1: // Неверный тип команды
    CmdReceive.eCommandResult:= COMMAND_RESULT.NOT_SUPPORTED; CmdReceive.bDone:= TRUE;
2: // Неверные входные параметры
    CmdReceive.eCommandResult:= COMMAND_RESULT.INCONSISTENT_PARAMETERS; CmdReceive.bDone:= TRUE;
3: // Изменение параметра во время работы
    CmdReceive.eCommandResult:= COMMAND_RESULT.PARAMETER_CHANGE_IN_EXECUTION; CmdReceive.bDone:=
    TRUE;
4: // Модуль не ответил на команду (отсутствует)
    CmdReceive.eCommandResult:= COMMAND_RESULT.HARDWARE_ERROR; CmdReceive.bDone:= TRUE;
5: // Команда запущена и выполняется (ничего не возвращает)
6: // В эту точку была отправлена другая команда, и она выполняется CmdReceive.eCommandResult:=
    COMMAND_RESULT.LOCKED_BY_OTHER_CLIENT; CmdReceive.bDone:= TRUE;
7: // Команда завершена успешно
    CmdReceive.eCommandResult:= COMMAND_RESULT.SUCCESS; CmdReceive.bDone:= TRUE;
END_CASE

END_IF

CmdReceive();

IF CmdReceive.bDone THEN
CmdReceive.bDone:= FALSE;
END_IF

```

Как видно из предыдущего кода, для облегчения генерации импульсов в цифровых двойных выходах Nexto был создан и использован функциональный блок, эквивалентный функции PulsedCommand библиотеки LibRtuStandard. Функциональный блок PulsedCommandNexto() показан в кодировке на языке ST.

5. КОНФИГУРАЦИЯ

```
FUNCTION_BLOCK PulsedCommandNexto VAR_INPUT
byCmdType:      BYTE;           // тип команды:
// 100 = статус
// 101 = закрыть/включить
// 102 = отключение/выключение
byPulseTime:   BYTE;           // Длительность импульса (в сотых долях секунды) ptDbpVarAdr:
POINTER TO DBP; // Адрес переменной DBP (может отображаться) stQuality:      QUALITY;
// Качество точки ДАД (цифровой модуль) END_VAR
VAR_OUTPUT
bON:   BOOL; // Нечетный вывод отображается на модуле Nexto DO bOFF:
BOOL; // Даже вывод отображается на модуле Nexto DO byStatus:
BYTE:= 7; // Возврат функции:
// 1 = неверная команда
// 2 = время вне допустимого диапазона (2..255)
// 3 = команда изменена во время работы
// 4 = модуль не ответил на команду (отсутствует)
// 5 = команда запущена или выполняется
// 6 = В этой точке уже есть активная команда
// 7 = импульсная команда успешно завершена
END_VAR
VAR
byState: BYTE; // Состояние функционального блока udiPulseEnd:
UDINT; // Мгновенное окончание импульса END_VAR

// Конечный автомат PulsedCommandNexto
CASE byState OF

0: // Состояние инициализации, готовность к приему команд:
CASE byCmdType OF

100:// Просто возвращает последний статус

101: // Выполнить импульс ВКЛ:
// Проверяет длительность импульса
IF byPulseTime > 1 THEN
// Проверить, есть ли уже активная команда в этой точке
IF ptDbpVarAdr^.ON OR ptDbpVarAdr^.OFF THEN
// Возвращает, что уже есть активная команда byStatus:= 6;
ELSE
// Включает выход ЗАКРЫТЬ ptDbpVarAdr^.ON:= TRUE;
ptDbpVarAdr^.OFF:= FALSE;
// Следующее состояние: выполнить
импульс ВКЛ. byState:= byCmdType; //
Возвращает запущенную команду byStatus:=
5; END_IF
ELSE
// Возвращает импульс вне диапазона
```



```

        byStatus:= 2; END_IF

102: // Выполнить импульс OFF
    // Проверяет длительность импульса
    IF byPulseTime > 1 THEN
        // Проверить, есть ли уже активная команда в этой точке
        IF ptDbpVarAdr^.ON OR ptDbpVarAdr^.OFF THEN // Возвращает,
            что уже есть активный byStatus:= 6;
        ELSE
            // Включает выход TRIP ptDbpVarAdr^.ON:=
            FALSE; ptDbpVarAdr^.OFF:= TRUE; //
            Следующий шаг: выполнить импульс OFF
            byState:= byCmdType; // Возвращает
            запущенную команду byStatus:= 5; END_IF
        ELSE
            // Возвращает импульс вне диапазона
            byStatus:= 2; END_IF
    ELSE
        // Возвращает неверную команду byStatus:= 1;
    END_CASE

// Запоминает момент окончания импульса udiPulseEnd:= SysTimeGetMs() +
BYTE_TO_UDINT(byPulseTime) * 10;

101, 102:// Продолжает выполнение импульса ВКЛ/ВЫКЛ //
Возвращает, что команда выполняется byStatus:= 5;
// Проверяет изменение рабочего параметра
IF byCmdType <> 100 AND byCmdType <> byState THEN // Возвращает
    текущее изменение параметра byStatus:= 3;
END_IF
// Проверяет окончание импульса
IF SysTimeGetMs() >= udiPulseEnd THEN // Отключить
    выходы TRIP и CLOSE ptDbpVarAdr^.ON:= FALSE;
    ptDbpVarAdr^.OFF:= FALSE;
    // Возвращает завершённую команду, только если команда не изменилась
    IF byCmdType = 100 OR byCmdType = byState THEN byStatus:= 7;
    END_IF
    // Следующее состояние: начальное
    byState:= 0;
END_IF

END_CASE

// Проверяет качество цифрового модуля (точки ДАД) IF stQuality.VALIDITY <> QUALITY_VALIDITY.VALIDITY_GOOD
THEN

```

5. КОНФИГУРАЦИЯ

```
// Отключить выходы TRIP и CLOSE
ptDbpVarAdr^.ON:= FALSE;
ptDbpVarAdr^.OFF:= FALSE; // Возвращает
отсутствующий модуль byStatus:= 4;
// Следующее состояние:
начальное byState:= 0; END_IF

// Скопируйте состояния выхода DBP в простые выходы bON:=
ptDbpVarAdr^.ON; bOFF:= ptDbpVarAdr^.OFF;
```

5.5.14.3. Основные параметры

Для конфигурации общих параметров сервера МЭК 60870-5-104 в соответствии с рисунком ниже следуйте таблице ниже параметров:

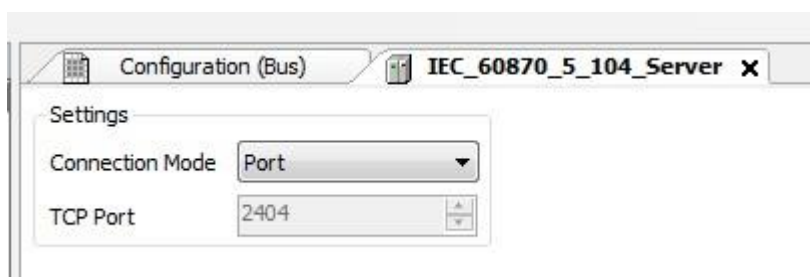


Рисунок 123: Экран общих параметров сервера IEC 60870-5-104

Параметр	Описание	по умолчанию	Возможности
Режим подключения	Установите режим соединения с модулями Connected Client.	Порт	Порт IP
TCP-порт	Определяет, какой номер порта TCP ПЛК будет использоваться для связи с подключенными клиентскими модулями. В случае, если в поле «Режим подключения» установлено значение «IP».	2404	1 до 65535

Таблица 143. Конфигурация общих параметров сервера IEC 60870-5-104.5.14.4.

Чтобы настроить отношение данных сервера IEC 60870-5-104, показанное на рисунке ниже, следуйте параметрам, описанным в таблице ниже:

Mappings	Value Variable	Object Type	Object Address	Size	Range	Counter Variable	Dead Band Variable	Dead Band Type	Select Required	Short Pulse (ms)	Long Pulse (ms)
Link Layer											
ApplicationLayer											

Рисунок 124: Экран сопоставления серверов IEC 60870-5-104

5. КОНФИГУРАЦИЯ

Параметр	Описание	по умолчанию	Возможности
Переменная значения	Символьное имя переменной	-	Имя переменной, объявленной в POU или GVL
Тип объекта	Конфигурация типа объекта IEC 60870-5-104	-	Информация об одной точке Информация о двойной точке Информация о позиции шага Измеренное значение (нормализованное) Измеренное значение (масштабированное) Измеренное значение (короткое) Плавающая точка) Интегрированные итоги Единая команда Двойная команда Регулирующая пошаговая команда Команда настройки точки (нормализованная) Команда настройки точки (в масштабе) Команда установки точки (короткая с плавающей запятой)
Адрес объекта	МЭК 60870-5-104 отображение индекса первой точки	-	1 до 65535
Размер	Определяет максимальное количество данных, к которым может получить доступ сопоставление IEC 608705-104.	-	1 до 86400000
Диапазон	Настроенный диапазон адресов данных	-	-
Переменная счетчика	Имя символической переменной, которая будет содержать значение переменной-счетчика	-	Имя переменной, объявленной в POU, GVL или модуле счетчика
Переменная зоны нечувствительности	Имя символической переменной, которая будет содержать значение зоны нечувствительности	-	Имя переменной, объявленной в POU или GVL

5. КОНФИГУРАЦИЯ

Тип зоны нечувствительности	Определяет тип зоны нечувствительности, который будет использоваться в отображении	Неполноценный	Абсолютный Неполноценный Интегрированный
Выберите обязательный	Определяет, требуется ли предыдущий выбор для запуска команды	Не правда	Правда Не правда
Параметр	Описание	по умолчанию	Возможности
Короткий импульс (мс)	Определяет длительность короткого импульса для цифровой команды IEC 60870-5-104.	1000	1 до 86400000
Длинный импульс (мс)	Определяет длительность длинного импульса для цифровой команды IEC 60870-5-104.	2000	1 до 86400000

Таблица 144. Конфигурация сопоставлений серверов IEC 60870-5-104

Примечания:

Переменная значения: когда отправляется команда чтения, результат, полученный в ответе, сохраняется в этой переменной. Когда это команда записи, записанное значение будет сохранено в этой переменной. Переменная может быть простой, массивом, элементом массива или может быть структурой.

Переменная счетчика: это поле применяется только при отображении объектов типа Интегрированные итоги, так как это переменная контроллера, которой нужно управлять в процессе. Он должен иметь тот же тип и размер переменной, объявленной в столбце Value Variable, значение которого будет считываться или сообщаться клиенту в случае событий.

ВНИМАНИЕ

Когда переменная счетчика имеет связанную с качеством переменную качества, которая должна быть передана в замороженную переменную по команде замораживания, она должна быть связана с переменной качества замороженной переменной. Эту процедуру необходимо выполнить через вкладку «Внутренние точки».

Переменная зоны нечувствительности: Это поле применяется только к отображениям входных аналоговых переменных (объекты типа измеряемого значения). Он должен иметь тот же тип и размер, что и переменная, объявленная в столбце «Значение переменной». Новые значения переменных зоны нечувствительности будут учитываться только тогда, когда входная аналоговая переменная изменит свое значение.

Тип зоны нечувствительности: Типы конфигурации, доступные для зоны нечувствительности:

Тип функции	Конфигурация	Описание
Тип зоны нечувствительности	Неполноценный	В этом варианте любое изменение значения в точке группы, чем оно меньше, генерирует событие для этой точки.
	Абсолютный	В этой опции, если абсолютное изменение значения точки группы больше, чем значение в поле «Мертвая зона», событие будет сгенерировано до этой точки.

5. КОНФИГУРАЦИЯ

	Интегрированный	В этой опции, если абсолютная величина интеграции изменения значения точки группы больше, чем значение в поле «Мертвая зона», событие будет сгенерировано для этой точки. Интервал интегрирования равен одной секунде.
--	-----------------	--

Таблица 145: IEC 60870-5-104 Сопоставление серверов Типы зон нечувствительности

Короткий импульс и длинный импульс: При определении продолжительности коротких и длинных импульсов необходимо учитывать пределы, поддерживаемые устройством, которое будет обрабатывать команду. Например, судьба - это выходная карта, которая не поддерживается Nexto Series. Необходимо проверить в техническом паспорте модуля, какое минимальное и максимальное время, а также разрешение для запуска импульсных команд.

5.5.14.5. Связующий слой

Для настройки параметров канального уровня сервера IEC 60870-5-104, показанной на рисунке ниже, следуйте параметрам, описанным в таблице ниже:

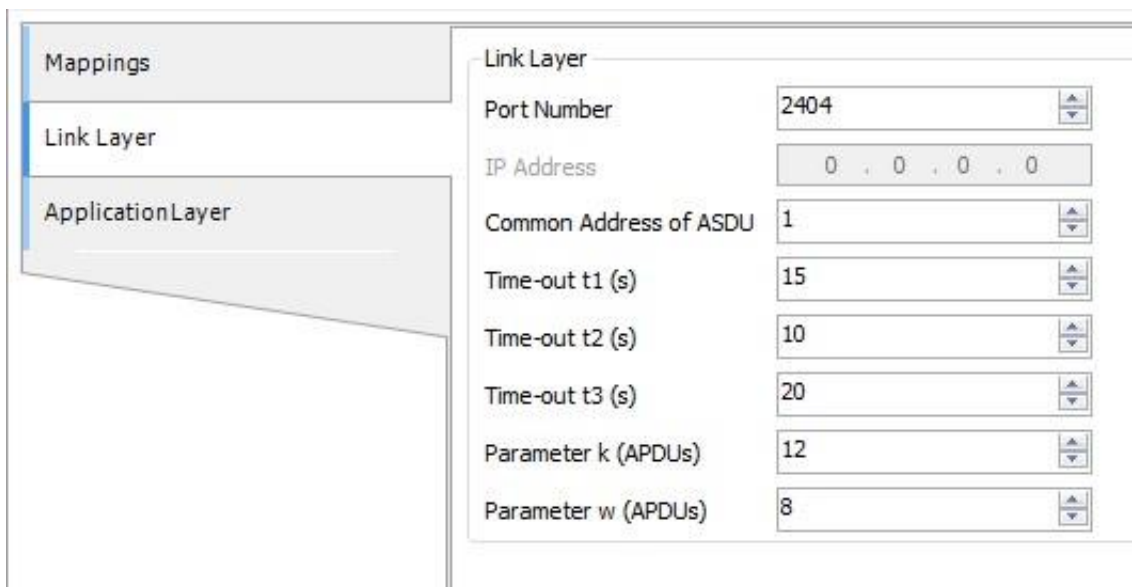


Рисунок 125: Экран конфигурации канального уровня IEC 60870-5-104 сервера

Параметры	Описание	По умолчанию	Возможности
Номер порта	Прослушиваемый адрес порта для подключения клиента. Используется, когда клиентское соединение не через IP	2404	1 до 65535
Айпи адрес	Подключенный клиентский IP-адрес, используемый, когда клиентское соединение осуществляется через IP-адрес.	0.0.0.0	1.0.0.1 до 223.255.255.254
Общий адрес	Адрес IEC 60870-5-104, если клиент подключен через IP	Адрес IEC 60870-5-104, если клиент	1 до 65534

5. КОНФИГУРАЦИЯ

		подключен через IP	
АСДУ	Период времени (в секундах), в течение которого устройство ожидает получения сообщения подтверждения после отправки сообщения APDU типа I или U (данные), прежде чем закрыть соединение	10	1 до 180
Тайм-аут t1 (с)	Период времени (в секундах), в течение которого устройство ожидает отправки контрольного сообщения (SFrame), подтверждающего получение кадра данных.	10	1 до 180
Тайм-аут t2 (с)	Период времени (в секундах), в течение которого будет отправлено сообщение для проверки связи в случае отсутствия передачи с обеих сторон.	20	1 до 180
Тайм-аут t3 (с)	Максимальное количество сообщений данных (I-Frame), переданных и не подтвержденных	12	1 до 12
Параметр k (APDU)	Параметр k (APDU)	По возможности	Возможности
Параметр	Параметр	8	1 до 8

Table 146: IEC 60870-5-104 Server Link Layer Configuration

Примечание:

Поля Time-out t1 (с), Time-out t2 (с) и Time-out t3 (с) являются зависимыми между собой и должны быть настроены таким образом, чтобы Time-out t1 (с) был больше, чем Time-out. t2 (с) и тайм-аут t3 (с) больше, чем тайм-аут t1 (с). Если какое-либо из этих правил не соблюдается, во время компиляции проекта будут генерироваться сообщения об ошибках.

ВНИМАНИЕ

Для медленных каналов связи (пример: спутниковая связь) параметры Тайм-аут t1 (с), Тайм-аут t2 (с) и Тайм-аут t3 (с) должны быть правильно настроены, например, удвоить значения этих полей по умолчанию.

5.5.14.6. Прикладной уровень

Для настройки прикладного уровня сервера IEC 60870-5-104, показанного на рисунке ниже, следуйте параметрам, описанным в таблице ниже:

5. КОНФИГУРАЦИЯ

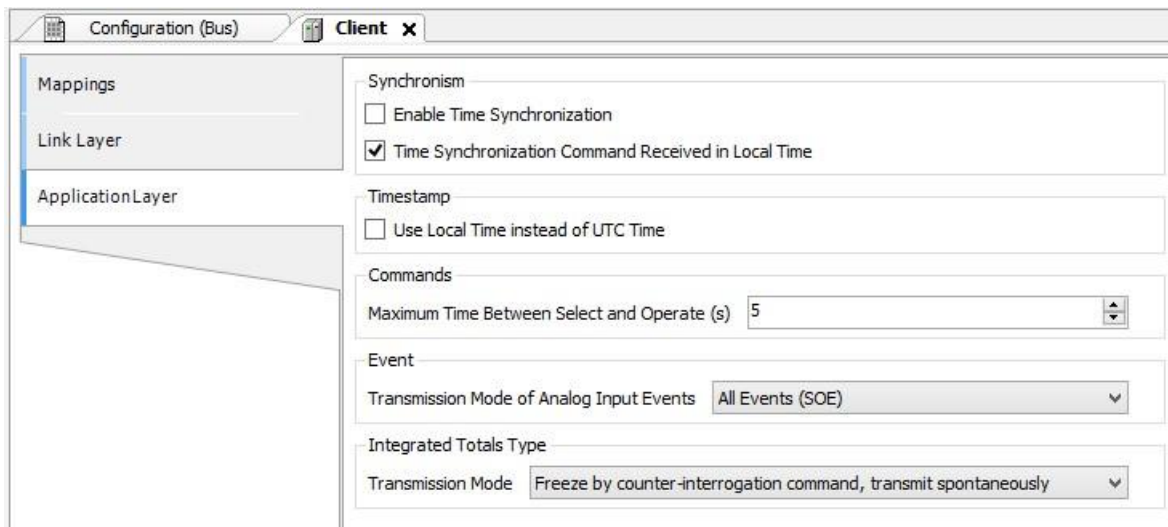


Рисунок 126: Сервер МЭК 60870-5-104 Экран конфигурации прикладного уровня

Параметр	Описание	По умолчанию	Возможности
Включить синхронизацию времени	Возможность включить/отключить запрос синхронизации времени	Выключено	Выключено Включено
Получена команда синхронизации времени по местному времени	Возможность включить/отключить обработку команды синхронизации по местному времени	Включено	Выключено Включено
Параметр	Описание	По умолчанию	Возможности
Используйте местное время вместо времени UTC	Возможность включить/отключить отметку времени по местному времени для событий.	Выключено	Выключено Включено
Максимальное время между выбором и срабатыванием (с)	Период времени, в течение которого команда выбора будет оставаться активной (отсчет начинается с полученного подтверждения команды выбора) в ожидании команды «Работать»	5	1 до 180
Режим передачи событий аналогового ввода	Режим передачи событий аналогового входа	Все события (SOE)	Все события (SOE) Наиболее частые события

5. КОНФИГУРАЦИЯ

Режим передачи	Режим передачи замороженных счетчиков (Интегрированные итоги)	Заморозить по команде встречного допроса, спонтанно передать	Заморозить по команде встречного опроса, спонтанно передать Заморозить и передать по команде встречного опроса
----------------	---	--	---

Таблица 147. Конфигурация серверного прикладного уровня IEC 60870-5-104

Примечания:

Включить синхронизацию времени: после включения сервер IEC 60870-5-104 позволяет настроить часы ЦП при получении команды синхронизации.

Команда синхронизации времени, полученная по местному времени: если эта функция включена, сервер IEC 60870-5-104 настраивает часы ЦП, рассматривая время, полученное в команде синхронизации, как местное время. В противном случае это время считается UTC.

Использовать местное время вместо времени UTC: после включения отметка времени событий, генерируемых сервером IEC 60870-5-104, будет отправляться в соответствии с местным временем ЦП.

ВНИМАНИЕ

Когда опция синхронизации времени включена более чем на одном сервере, полученное время с разных серверов будет перезаписано в системных часах за короткий период времени, что может вызвать нежелательное поведение из-за задержек во времени распространения сообщений и загрузки системы.

Режим передачи событий аналоговых входов: доступны следующие режимы передачи событий аналоговых входов:

Тип функций	Конфигурация	Описание
Режим передачи событий аналогового ввода	Все события (SOE)	Все сгенерированные аналоговые события будут отправлены.
	Самое последнее событие	Отправляется только самое последнее аналоговое событие.

Таблица 148: IEC 60870-5-104 Режимы передачи сервером событий аналоговых входов

Режим передачи: доступны следующие режимы передачи замороженных счетчиков (интегрированных сумм):

Тип функций	Конфигурация	Описание
Режим передачи	Заморозить по команде встречного допроса, спонтанно передать	Эквивалентен режиму сбора данных счетчиков D (интегрированные итоги), определенному стандартом IEC 60870-5-101. В этом режиме станции управления дают команды опроса счетчиков, замораживание счетчиков. Если замороженные значения были изменены, о них сообщается через события.

5. КОНФИГУРАЦИЯ

	Заморозить и передать по команде встречного допроса	Эквивалентен режиму сбора данных счетчиков С (интегрированные итоги), определенному стандартом IEC 60870-5-101. В этом режиме станции управления дают команды опроса счетчиков, замораживание счетчиков. Последующие команды опроса счетчиков (чтение) посылаются станцией управления для получения замороженных значений.
--	---	--

Таблица 149. Режимы передачи сервера IEC 60870-5-104 замороженных счетчиков

ВНИМАНИЕ

Стандарт IEC 60870-5-104, раздел Управление передачей с использованием Start/Stop, предусматривает использование команд STARTDT и STOPDT для управления трафиком данных между клиентом и сервером с использованием простых или множественных соединений. Несмотря на то, что Nexto поддерживает такие команды, его использование для управления передачей данных не рекомендуется, в основном с резервными ЦП, поскольку такие команды не синхронизируются между обоими ЦП. Вместо использования нескольких соединений между клиентом и сервером Nexto предлагается использовать ресурсы NIC Teaming для предоставления (физически) резервных каналов Ethernet и сохранения ресурсов ЦП (центры управления ЦП).

5.5.14.7. Диагностика сервера

Диагностика протокола сервера IEC 60870-5-104 хранится в переменных типа T_DIAG_IEC104_SERVER_1, которые описаны в таблице ниже:

Диагностика переменной типа T_DIAG_IEC104_SERVER_1.*	Размер	Описание
Командные биты, автоматически сбрасываются:		
tCommand.bStop	BOOL	Отключить драйвер
tCommand.bStart	BOOL	Включить драйвер
tCommand.bDiag_01_Reserved	BOOL	Сдержанный
tCommand.bDiag_02_Reserved	BOOL	Сдержанный
tCommand.bDiag_03_Reserved	BOOL	Сдержанный
tCommand.bDiag_04_Reserved	BOOL	Сдержанный
tCommand.bDiag_05_Reserved	BOOL	Сдержанный
tCommand.bDiag_06_Reserved	BOOL	Сдержанный
Диагностика:		
tClient_X.bRunning	BOOL	Сервер IEC 60870-5-104 работает
tClient_X.eConnectionStatus. CLOSED	ENUM(BYTE)	Канал связи закрыт. Сервер не принимает запрос на подключение. Значение ENUM (0)

5. КОНФИГУРАЦИЯ

tClient_X.eConnectionStatus. LISTENING		Сервер прослушивает настроенный порт и нет подключенных клиентов. Значение ENUM (1)
Диагностика переменной типа T_DIAG_IEC104_SERVER_1.*	Размер	Описание
tClient_X.eConnectionStatus. CONNECTED		Подключенный клиент. Значение ENUM (2)
tClient_X.tQueueDiags. bOverflow	BOOL	Очередь клиентов переполнена
tClient_X.tQueueDiags. wSize	WORD	Настроенный размер очереди
tClient_X.tQueueDiags. wUsage	WORD	Количество событий в очереди
tClient_X.tQueueDiags. dwReserved_0	DWORD	Сдержанный
tClient_X.tQueueDiags. dwReserved_1	DWORD	Сдержанный
tClient_X.tStats.wRXFrames	WORD	Количество полученных кадров
tClient_X.tStats.wTXFrames	WORD	Количество отправленных кадров
tClient_X.tStats.wCommErrors	WORD	Счетчик ошибок связи, включая ошибки физического уровня, канального уровня и транспортного уровня.
tClient_X.tStats.dwReserved_0	DWORD	Сдержанный
tClient_X.tStats.dwReserved_1	DWORD	Сдержанный

Таблица 150: Диагностика сервера IEC 60870-5-104

5.5.14.8. Классификатор команд

Стандарт IEC 60870-5-104 предусматривает четыре различных квалификатора команд для объектов Single Command, Double Command и Regulatory Step Command, все из которых поддерживаются сервером Nexto.

Каждый тип объекта имеет определенное поведение для каждого квалификатора команды, как показано в таблице ниже.

Квалификатор	Тип объекта протокола IEC 60870-5-104		
	Одиночная команда	Двойная команда	Регулирующая пошаговая команда
Без дополнительного определения (по умолчанию)	То же поведение постоянного квалификатора.	Такое же поведение квалификатора коротких импульсов.	Такое же поведение квалификатора коротких импульсов.
Короткая длительность импульса	Требуется перехват команды для обработки приложения. В противном случае он вернет сообщение об отрицательном подтверждении (сбой).	Требуется перехват команды для обработки приложения. В противном случае он вернет сообщение об отрицательном	Требуется перехват команды для обработки приложения. В противном случае он вернет сообщение об отрицательном подтверждении (сбой).
Длинная длительность импульса	Требуется перехват команды для обработки приложения. В противном случае он вернет сообщение об отрицательном подтверждении (сбой).	Требуется перехват команды для обработки приложения. В противном случае он вернет сообщение об отрицательном	Требуется перехват команды для обработки приложения. В противном случае он вернет сообщение об отрицательном подтверждении (сбой).

5. КОНФИГУРАЦИЯ

Постоянный вывод	Выход будет включен или выключен, и это будет оставаться до новой команды в соответствии со значением (ВКЛ или ВЫКЛ), заданным клиентом.	подтверждении (сбой).	
------------------	--	-----------------------	--

Таблица 151: Классификатор серверных команд IEC 60870-5-104

Примечание:

Перехват команд: Дополнительную информацию о перехвате команд клиентов IEC 60870-5-104 см. в разделе «Перехват команд, поступающих из Центра управления», реализуемый с помощью функционального блока CommandReceiver.

5.6. Коммуникационная производительность

5.6.1. MODBUS -сервер

Устройства MODBUS, настраиваемые в ЦП Nexto, работают в фоновом режиме с приоритетом ниже пользовательского приложения и циклически. Таким образом, их производительность варьируется в зависимости от оставшегося времени с учетом разницы между интервалом и временем, которое требуется приложению для запуска. Например, устройство MODBUS в приложении, которое запускается каждые 100 мс, с временем работы 50 мс, будет иметь более низкую производительность, чем приложение, работающее с интервалом от 50 до 200 мс. Это происходит потому, что в последнем случае у ЦП будет больше времени между каждым циклом MainTask для выполнения задач с более низким приоритетом.

Также необходимо учитывать количество циклов, которое требуется устройству, ведомому устройству или серверу для ответа на запрос. Для обработки и передачи ответа ведомому устройству MODBUS RTU требуются циклы (время цикла задачи MODBUS), тогда как задача сервера MODBUS Ethernet занимает только один цикл. Но это минимальное время между получением запроса и ответом. Если запрос отправляется сразу после выполнения задачи, время цикла MODBUS может быть в 2 или 3 раза больше времени цикла для ведомого устройства MODBUS и в 1-2 раза больше времени цикла для сервера MODBUS.

В этом случае: Максимальное время ответа = $3 * (\text{время цикла}) + (\text{время выполнения задач}) + (\text{время межкадровых символов}) + (\text{задержка отправки})$.

Например, для задачи сервера MODBUS Ethernet с циклом 50 мс, приложения, которое выполняется в течение 60 мс каждые 100 мс, сервер может выполнять только один цикл между каждым циклом приложения. С другой стороны, при использовании того же приложения, работающего в течение 60 мс, но с интервалом в 500 мс, MODBUS работает лучше, потому что пока приложение не запущено, оно будет запускаться каждые 50 мс и только каждый цикл MainTask его будет дольше бегать. В этих случаях наихудшей производительностью будет сумма времени выполнения пользовательского приложения со временем цикла задачи MODBUS.

Для ведущего и клиентского устройств принцип работы точно такой же, но с учетом времени опроса отношения MODBUS, а не времени цикла задачи MODBUS. В этих случаях наихудшая производительность связи будет достигнута после времени опроса вместе с временем выполнения пользовательского приложения.

Важно подчеркнуть, что количество запущенных устройств MODBUS также меняет его производительность. В пользовательском приложении со временем выполнения 60 мс и интервалом 100 мс у ЦП остается 40 мс для выполнения всех задач с более низким приоритетом. Следовательно, ЦП только с одним сервером Ethernet MODBUS будет иметь более высокую производительность, чем ЦП, использующий четыре таких устройства.

5.6.1.1. Локальные интерфейсы ЦП

Для устройства MODBUS Ethernet Server мы можем утверждать, что устройство способно отвечать на x запросов в секунду. Или, другими словами, Сервер способен передавать n байтов в секунду, в зависимости от размера каждой заявки. Чем меньше время цикла задачи сервера MODBUS, тем выше влияние количества соединений на его скорость ответов. Однако для времени цикла менее 20 мс это влияние не является линейным, и для информации необходимо просмотреть приведенную ниже таблицу.

В приведенной ниже таблице показано количество запросов, на которые способен ответить сервер MODBUS, подключенный к локальному интерфейсу Ethernet, в зависимости от времени цикла, настроенного для задачи MODBUS, и количества активных соединений:

5. КОНФИГУРАЦИЯ

Количество активных соединений	Отвеченных запросов в секунду с рабочим циклом MODBUS в 5 мс	Отвеченных запросов в секунду с рабочим циклом MODBUS в 10 мс	Отвеченных запросов в секунду с рабочим циклом MODBUS в 20 мс
1 соединение	185	99	50
2 соединения	367	197	100
4 соединения	760	395	200
7 соединения	1354	695	350
10 соединения	1933	976	500

Таблица 152: Скорость связи сервера MODBUS на локальном интерфейсе

ВНИМАНИЕ

Коммуникационные характеристики, упомянутые в этом разделе, являются лишь примерами использования ЦП только с одним устройством MODBUS TCP Server, без какой-либо логики, которая должна выполняться внутри приложения, что может задержать связь. Поэтому эти показатели необходимо принять за максимальные показатели.

Для времени цикла, равного или превышающего 20 мс, увеличение частоты ответов является линейным и может быть рассчитано с использованием уравнения:

$$N = C \times (1/T)$$

Где:

N — среднее количество ответов в секунду; C — количество активных подключений; T — интервал задачи MODBUS в секундах.

В качестве примера сервера MODBUS с одним активным соединением и временем цикла 50 мс мы получаем:

$$C = 1; T = 0,05 \text{ с};$$

$$N = 1 \times (1/(0,05))$$

$$N = 20$$

То есть в этой 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.

5.6.1.2. Удаленные интерфейсы

Производительность сервера MODBUS устройства в одном удаленном интерфейсе Ethernet аналогична производительности в локальных интерфейсах ЦП.

Однако из-за времени обмена данными между ЦП и модулями максимальная производительность ограничена. Только для одного активного соединения количество ответов ограничено максимум 18 ответами в секунду. При более активных подключениях количество ответов будет увеличиваться линейно, точно так же, как и для локальных интерфейсов, но с ограничением до 90 ответов в секунду. Итак, для удаленного интерфейса Ethernet у нас будут следующие формы для расчета его производительности:

Для $T \leq 55$ мс используется:

$$N = C \times (18,18 - (18,18 / (0,055 \times 1000)))$$
 А для $T \geq 55$ мс используется:

$$N = C \times (Z - (Z / (T \times 1000)))$$

Где N — среднее количество ответов в секунду, C — количество активных подключений, а T — время цикла задачи MODBUS (в секундах).

Пользователь должен обратить внимание на то, что максимальная производительность устройства MODBUS Server в одном удаленном интерфейсе Ethernet составляет 90 ответов на заявки в секунду.

5.6.2. OPC-DA-сервер

Производительность связи с сервером OPC DA была протестирована путем создания POU с 1000 переменными INT в каждом. Все сценарии были протестированы с одним профилем и интервалом основной задачи 100 мс. Связь была разрешена атрибутом 'symbol' := 'readwrite', чтобы сделать данные доступными для сервера OPC DA. Измерения проводились, когда MasterTool был отключен от ЦП, а продолжительность основной задачи составляла 5 %, 50 % и 80 % настроенного интервала, как показано в таблице ниже.

5. КОНФИГУРАЦИЯ

На стороне клиента OPC использовался системный драйвер SCADA. Настроенное время обновления составляло 50 мс. Результаты работы в этих условиях описаны в таблице ниже.

Общее количество переменных в проекте PLC's	Переменное время обновления на клиенте OPC DA		
	5% занятости процессора	50% занятости процессора	80% занятости процессора
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
15000	3200 мс	6400 мс	20000 мс
20000	4000 мс	8100 мс	25000 мс

Таблица 153: Скорость связи сервера OPC DA

5.6.3. Сервер OPC UA

Для тестов производительности связи с сервером OPC UA были созданы проекты для ПЛК путем объявления переменных типа INT.

Все тестовые сценарии имеют следующие общие характеристики:

Проекты с Machine Profile и MainTask Interval настроены на 100 мс;

Все переменные типа INT изменяются каждые 100 мс. В протоколе OPC UA клиентам OPC UA сообщается только об измененных переменных. В приложениях, где несколько переменных часто меняют производительность

Связь OPC UA значительно улучшается;

Продолжительность MainTask настроена на потребление 10 %, 30 % и 60 % настроенного интервала в 100 мс. Когда продолжительность MainTask очень велика, вычислительная мощность для обмена данными OPC UA мала, что может значительно снизить его производительность;

Параметр «Интервал публикации» был настроен на клиенте в 1000 мс, это означает, что переменные, изменяющие его значение, должны сообщаться клиенту каждые 1000 мс. Очень низкое значение этого параметра может значительно снизить производительность связи OPC UA;

Параметр «Интервал выборки» был настроен на клиенте в 500 мс, это означает, что контроллер (сервер) будет обнаруживать изменения значений в переменных каждые 500 мс, чтобы знать, о каких из них следует сообщать клиенту. Очень низкое значение этого параметра может значительно снизить производительность связи OPC UA. Рекомендуется, что он должен составлять не менее половины «интервала публикации» и не более «интервала публикации»;

Шифрование не настроено. Если настроить, производительность связи OPC UA немного снизится;

Был подключен только один клиент OPC UA. Подключение нескольких клиентов может значительно снизить производительность связи OPC UA.

В таблице ниже показан интервал между сеансами связи (значение которого должно составлять 1000 мс в соответствии с настроенным интервалом публикации):

Общее количество переменных INT в проекте ПЛК	Время обновления переменной в клиенте OPC UA		
	10% процессор занят	30% процессор занят	60% процессор занят
1000	1010 мс	1014 мс	1040 мс

5. КОНФИГУРАЦИЯ

2000	1013 мс	1017 мс	1472 мс
5000	1321 мс	1405 мс	2351 мс

Таблица 154: Скорость связи сервера OPC UA

ВНИМАНИЕ

В примечаниях по применению NAP165 более подробно анализируется производительность связи OPC UA, в том числе рассматривается потребление пропускной способности связи Ethernet. В этом примечании по применению также обсуждаются концепции работы протокола OPC UA.

5.6.4. Сервер IEC60870-5-104

Драйвер сервера МЭК 60870-5-104 выполняется ЦП так же, как и другие коммуникационные драйверы серверов, то есть в фоновом режиме, с приоритетом ниже пользовательского приложения и циклически. Задача этого драйвера конкретно выполняется каждые 50 мс, а для обработки и ответа на запросы достаточно 1 цикла выполнения драйвера. Однако, поскольку это задача с низким приоритетом, ее выполнение на этой частоте не гарантируется, поскольку зависит от процента свободного ЦП (разница между интервалом MainTask и временем, которое требуется для выполнения пользовательского приложения), а также параллелизм с задачами из других протоколов, настроенных в ЦП.

Чтобы помочь в понимании драйвера производительности сервера IEC 60870-5-104, представлены результаты некоторых тестов, проведенных с помощью симулятора клиента IEC 60870-5-104, подключенного к NX3030, на котором работает сервер IEC 60870-5-104. Настроенная база данных состояла из 900 цифровых точек и 100 аналоговых точек (все с отметками качества и времени), а основная задача использовала 70 мс (из 100-мс интервала).

Время выполнения общей команды допроса: менее одной секунды

Время передачи 900 цифровых событий + 100 аналоговых событий: 6 секунд

5.7. Производительность системы

В случаях, когда приложение имеет только одну пользовательскую задачу MainTask, отвечающую за выполнение одной программной единицы типа «Программа», называемой MainPrg (как в случае с одним профилем), ПЛК потребляет определенное количество времени для обработки задачи. В то время мы называем это временем выполнения.

В приложении среднее время выполнения приложения можно узнать с помощью MasterTool IEC XE в элементе «Устройство» его дерева устройств следующим образом:

Логика ПЛК->Приложение->Конфигурация задачи на вкладке Монитор, столбец Среднее время цикла.

Пользователь должен обращать внимание на время цикла, чтобы оно не превышало 80% интервала, установленного в пользовательской задаче MainTask. Например, в приложении, где интервал составляет 100 мс, соответствующее время цикла составляет до 80 мс. Это связано с тем, что ЦП требуется время для выполнения других задач, таких как обработка связи, обработка дисплея и карты памяти, и эти задачи выполняются в пределах диапазона (оставшиеся 20% времени цикла).

ВНИМАНИЕ

При очень большом времени цикла (обычно более 300 мс) даже при соблюдении значения 80 % может произойти сокращение времени отклика дисплея и диагностической клавиши. В случае несоблюдения 80-процентного значения и времени выполнения пользовательской задачи ближе или превышает интервал, установленный для основной задачи, экран и кнопка диагностики не могут реагировать, поскольку их приоритет в системе ниже, чем у пользовательских задач. Если в ЦП загружено приложение с ошибками, может потребоваться его перезапуск без загрузки этого приложения, как показано в разделе «Системный журнал».

5. КОНФИГУРАЦИЯ

Системные журналы ЦП серии Nexto, начиная с версии прошивки 1.4.0.33, теперь перезагружаются в случае сброса ЦП или перезагрузки системы выполнения, то есть вы можете просмотреть более старые журналы при возникновении одного из этих условий.

5.7.1. Время сканирования ввода/вывода

Для проекта, в котором используются цифровые модули ввода-вывода, поскольку они вставлены в шину и объявлены в проекте, время основной задачи будет увеличиваться в соответствии с количеством модулей. В таблице ниже показано среднее время, которое добавляется к основной задаче:

Объявленные модули в шине	Добавленное время во время цикла MainTask (мкс)
5	300
10	700
20	1000

Таблица 155: Время сканирования ввода/вывода

В проектах, использующих удаленный ввод-вывод, например, с использованием ведущего модуля NX5001 PROFIBUS-DP, необходимо обращаться к руководству по соответствующему модулю для получения информации о производительности и влиянии модуля на выполнение пользовательских задач.

5.7.2. Карта памяти

Передача данных с использованием карты памяти выполняется ЦП в фоновом режиме, так как приоритет отдается выполнению пользовательского приложения и обработке связи. Таким образом, передача файлов на карту может страдать дополнительным значительным временем, зависящим от Cycle Time пользовательского приложения.

Время, необходимое для чтения/записи файлов на карту, будет напрямую зависеть от времени цикла пользовательского приложения, поскольку это приложение имеет приоритет при выполнении.

Дополнительную информацию об использовании карты памяти см. в разделе «Карта памяти».

5.8. Часы реального времени

ЦП серии Nexto имеют внутренние часы, которые можно использовать с помощью библиотеки NextoStandard.lib. Эта библиотека автоматически загружается при создании нового проекта (процедуру вставки библиотеки см. в разделе «Библиотеки»). На рисунке ниже показано, как включить блоки в проект:

5. КОНФИГУРАЦИЯ

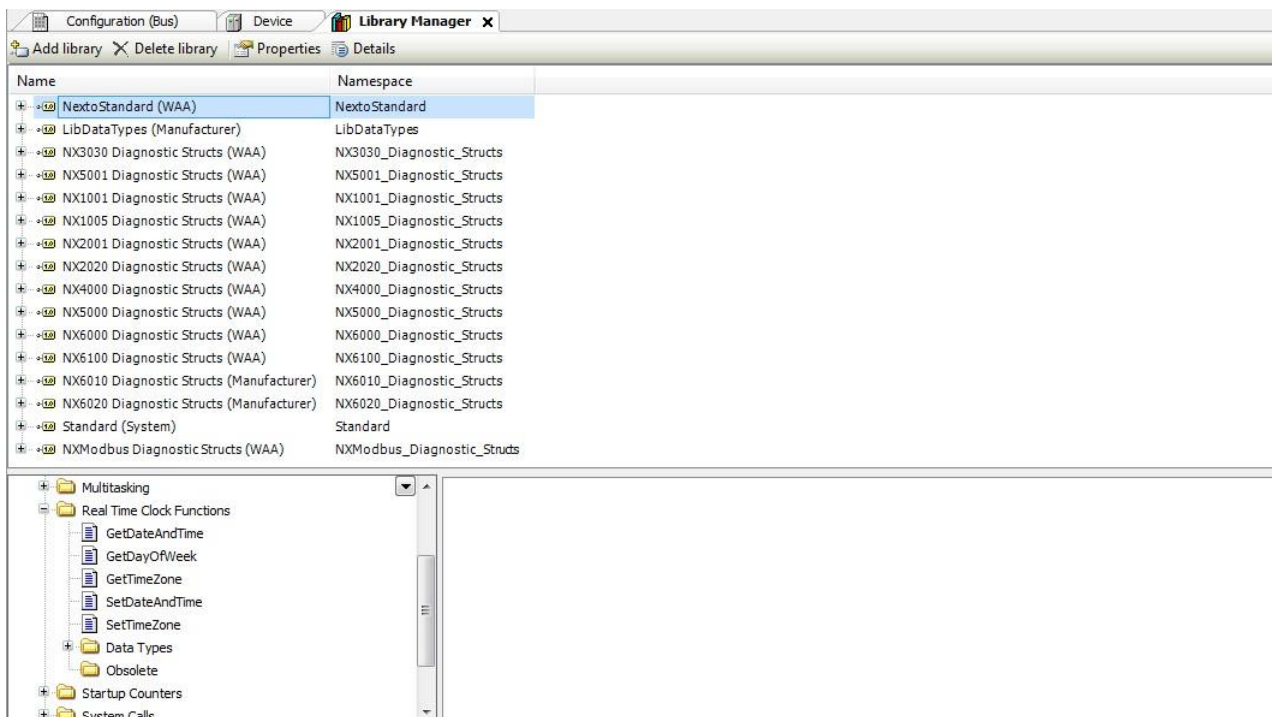


Рисунок 127: Блоки чтения и записи часов

ВНИМАНИЕ

Функциональные блоки чтения и записи RTC, ранее доступные в версии 2.00 MasterTool IEC XE или более ранней версии, устаревают с версии 2.00 или более новой, следующие блоки больше не используются: NextoGetDateAndTime, NextoGetDateAndTimeMs, NextoGetTimeZone, NextoSetDateAndTime, NextoSetDateAndTimeMs и NextoSetTimeZone.

5.8.1. Функциональные блоки для чтения и записи RTC

Среди других функциональных блоков есть очень важные, используемые для чтения часов (GetDateAndTime, GetDayOfWeek и GetTimeZone) и для настройки новых данных даты и времени (SetDateAndTime и SetTimeZone). Эти функции всегда используют местное время, то есть учитывают значение, определенное часовым поясом.

Процедура настройки этих двух блоков описана ниже.

ВНИМАНИЕ

Устаревшие функциональные блоки для чтения и записи RTC (NextoGetDateAndTime, NextoGetDateAndTimeMs, NextoSetDateAndTime и NextoSetDateAndTimeMs) нельзя использовать в области избыточных данных в избыточных проектах. Они должны использоваться в неизбыточных POU, таких как NonSkippedPrg POU. Более подробную информацию о том, как работает POU NonSkippedPrg, можно найти в программе NonSkippedPrg.

5.8.1.1. Функциональные блоки для чтения RTC

Считывание часов может быть выполнено с помощью следующих функций:

5. КОНФИГУРАЦИЯ

5.8.1.1.1. GetDateAndTime



Рисунок 128: Чтение даты и времени

Внутренние параметры	Тип	Описание
DATEANDTIME	EXTENDED_DATE_AND_TIME	Эта переменная возвращает значение даты и часа RTC в формате, показанном в таблице 165.

Таблица 156: Внутренние параметры GetDateAndTime

Внешние параметры	Тип	Описание
GETDATEANDTIME	RTC_STATUS	Возвращает состояние ошибки функции, см. Таблицу 167.

Таблица 157: Внешние параметры GetDateAndTime

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR
Result : RTC_STATUS;
DATEANDTIME : EXTENDED_DATE_AND_TIME; xEnable : BOOL;
END_VAR
-----
IF xEnable = TRUE THEN
Result := GetDateAndTime(DATEANDTIME); xEnable := FALSE;
END_IF
    
```

5.8.1.1.2. GetTimeZone

Следующая функция считывает конфигурацию часового пояса, эта функция напрямую связана со временем в часовом поясе в службе синхронизации SNTP:

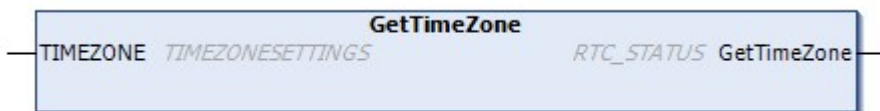


Рисунок 129: Чтение конфигурации часового пояса

Внутренние параметры	Тип	Описание
----------------------	-----	----------

5. КОНФИГУРАЦИЯ

TIMEZONE	TIMEZONESETTINGS	Эта переменная представляет чтение конфигурации часового пояса.
----------	------------------	---

Таблица 158: Внутренние параметры GetTimeZone

Внешние параметры	Тип	Описание
GetTimeZone	RTC_STATUS	Возвращает состояние ошибки функции, см. Таблицу 167.

Таблица 159: Внешние параметры GetTimeZone

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR
GetTimeZone_Status : RTC_STATUS; TimeZone      :
TIMEZONESETTINGS; xEnable : BOOL;
END_VAR
-----
IF xEnable = TRUE THEN
GetTimeZone_Status := GetTimeZone(TimeZone); xEnable := FALSE;
END_IF
```

5.8.1.1.3. GetDayOfWeek

Функция *GetDayOfWeek* используется для чтения дня недели.



Рисунок 130: Чтение дней недели

Внешние параметры	Тип	Описание
GetDayOfWeek	DAYS_OF_WEEK	Возвращает день недели. См. Раздел 166.

Таблица 160: Выходные параметры GetDayOfWeek

При вызове функция считывает день недели и заполняет структуру DAYS_OF_WEEK.

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR
DayOfWeek : DAYS_OF_WEEK;
END_VAR
-----
DayOfWeek := GetDayOfWeek();
```

5. КОНФИГУРАЦИЯ

5.8.1.2. Функции записи RTC

Настройки часов выполняются через функции и функциональные блоки следующим образом:

5.8.1.2.1. SetDateAndTime

Функция SetDateAndTime используется для записи настроек часов. Обычно точность составляет порядка сотен миллисекунд.



Рисунок 131: Установка времени и даты

Внутенние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, когда получает нарастающий фронт, разрешает запись часов.
DATEANDTIME	EXTENDED_DATE_AND_TIME	Получает значения даты и часа в миллисекундах. См. раздел 165.

Таблица 161: Внутренние параметры SetDateAndTime

Внутенние параметры	Тип	Описание
DONE	BOOL	Если эта переменная имеет значение true, это означает, что действие было успешно завершено.
EXEC	BOOL	Если эта переменная имеет значение true, это означает, что функция обрабатывает значения.
ERROR	BOOL	Если эта переменная имеет значение true, это указывает на ошибку во время записи.
STATUS	RTC_CMD_STATUS	Возвращает ошибку, возникшую во время настройки. См. Таблицу 167.

Таблица 162: Внешние параметры SetDateAndTime

Когда на входе REQUEST появляется нарастающий фронт, функциональный блок записывает новые значения DATEANDTIME в часы. Если запись выполнена успешно, вывод DONE будет равен TRUE. В противном случае вывод ERROR будет равен TRUE, и ошибка появится в переменной STATUS.

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR
SetDateAndTime : SetDateAndTime; xRequest : BOOL;
DateAndTime : EXTENDED_DATE_AND_TIME; xDone :
BOOL; xExec : BOOL; xError : BOOL; xStatus :
RTC_STATUS;
END_VAR
-----
IF xRequest THEN
    SetDateAndTime.REQUEST:=TRUE;
    SetDateAndTime.DATEANDTIME:=DateAndTime; xRequest:= FALSE;
END_IF
SetDateAndTime();
SetDateAndTime.REQUEST:=FALSE;
IF SetDateAndTime.DONE THEN xExec:=SetDateAndTime.EXEC;
    xError:=SetDateAndTime.ERROR; xStatus:=SetDateAndTime.STATUS;
END_IF
    
```

ВНИМАНИЕ

Если вы попытаетесь записать значения времени за пределами диапазона RTC, значения будут преобразованы в допустимые значения при условии, что они не превышают допустимый диапазон от 01.01.2000 до 31.12.2035. Например, если пользователь попытается записать значение 2000 мс, оно будет преобразовано в 2 секунды, напишите значение 100 секунд, оно будет преобразовано в 1 минуту и 40 секунд. Если значение типа 30 часов, оно преобразуется в 1 день и 6 часов и т. д..

5.8.1.2.2. *SetTimeZone*

Следующий функциональный блок выполняет запись настроек часового пояса:

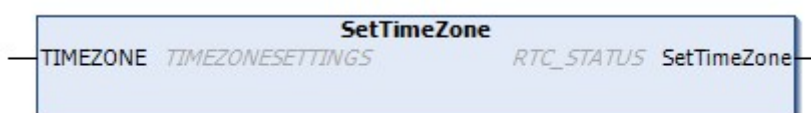


Рисунок 132: Запись настроек часового пояса

Внутренние параметры	Тип	Описание
TIMEZONE	TIMEZONESETTINGS	Структура с часовым поясом для настройки. См. Таблицу 168.

Таблица 163: Внутренние параметры SetTimeZone

Внешние параметры	Тип	Описание
SetTimeZone	RTC_STATUS	Возвращает ошибку, возникшую во время чтения/настройки. См. Таблицу 167.

Table 164: Внешние параметры SetTimeZone

При вызове функция настроит TIMEZONE с новой конфигурацией системного часового пояса. Результаты конфигурации возвращаются функцией.

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR
Status : RTC_STATUS;
TimeZone : TIMEZONESETTINGS; xWrite : BOOL;
END_VAR
-----//FB SetTimeZone
IF (xWrite = TRUE) THEN
Status := SetTimeZone(TimeZone);
  IF Status = RTC_STATUS.NO_ERROR THEN xWrite :=
  FALSE; END_IF
END_IF
```

ВНИМАНИЕ

Для выполнения часов должны использоваться значения времени и даты в следующем допустимом диапазоне: с 00:00:00 часов 01.01.2000 до 31.12.2035 23:59:59 часов, в противном случае сообщается об ошибке через Выходной параметр СТАТУС. Подробнее о выходном параметре STATUS см. в разделе [RTC_STATUS](#).

5.8.2. Структуры данных RTC

Функциональные блоки чтения и настройки ЦП серии Nexto RTC используют в своей конфигурации следующие структуры данных:

5.8.2.1. EXTENDED_DATE_AND_TIME

Эта структура используется для хранения даты RTC при использовании функциональных блоков для чтения/установки даты с точностью до миллисекунд. Это описано в таблице ниже:

Структура	Тип	Переменная	Описание
EXTENDED_DATE_AND_TIME	BYTE	byDayOfMonth	Сохраняет день установленной даты.
	BYTE	ByMonth	Сохраняет месяц установленной даты.
	WORD	wYear	Сохраняет год установленной даты.
	BYTE	byHours	Сохраняет час установленной даты.
	BYTE	byMinutes	Сохраняет минуты установленной даты.
	BYTE	bySeconds	Сохраняет секунды установленной даты.
	WORD	wMilliseconds	Сохраняет миллисекунды установленной даты.

Таблица 165: EXTENDED_DATE_AND_TIME

5.8.2.2. DAYS_OF_WEEK

Эта структура используется для хранения дня недели:

Исчисляемый	Значение	Описание

5. КОНФИГУРАЦИЯ

DAYS_OF_WEEK	0	INVALID_DAY
	1	SUNDAY
	2	MONDAY
	3	TUESDAY
	4	WEDNESDAY
	5	THURSDAY
	6	FRIDAY
	7	SATURDAY

Таблица 166: Структура DAYS_OF_WEEK

5.8.2.3. RTC_STATUS

Этот перечислитель используется для возврата типа ошибки в настройке или чтении RTC, и он описан в таблице ниже:

Исчисляемый	Значение	Описание
RTC_STATUS	NO_ERROR (0)	Нет ошибки.
	UNKNOWN_COMMAND (1)	Неизвестная команда.
	DEVICE_BUSY (2)	Устройство занято.
	DEVICE_ERROR (3)	Устройство с ошибкой.
	ERROR_READING_OSF (4)	Ошибка чтения действительного флага даты и часа.
	ERROR_READING_RTC (5)	Ошибка в чтении даты и часа.
	ERROR_WRITING_RTC (6)	Ошибка в написании даты и часа.
	ERROR_UPDATING_SYSTEM_TIME (7)	Ошибка в обновлении системной даты и часа.
	INTERNAL_ERROR (8)	Внутренняя ошибка.
	INVALID_TIME (9)	Неверная дата и час.
	INPUT_OUT_OF_RANGE (10)	Вышли за пределы допустимой даты и часа для системы.
Исчисляемый	Значение	Описание
	SNTP_NOT_ENABLE (11)	Ошибка генерируется, когда служба SNTP не включена и делается попытка изменить часовой пояс.

Таблица 167: RTC_STATUS

5.8.2.4. TIMEZONESETTINGS

5. КОНФИГУРАЦИЯ

Эта структура используется для хранения значения часового пояса в запросах на чтение/установку функциональных блоков RTC и описана в таблице ниже:

Структура	Тип	Переменная	Описание
TIMEZONESETTINGS	INT	iHour	Установить час часового пояса.
	INT	iMinutes	Установите минуты часового пояса.

Таблица 168: TIMEZONESETTINGS

Примечание:

Функциональные блоки записи и чтения даты и часа: различные библиотеки NextoStandard, которые имеют функциональные блоки или функции, которые могут осуществлять доступ чтения и записи даты и часа в системе, не указаны. Библиотека NextoStandard имеет соответствующие интерфейсы для записи и чтения системной даты и часа соответственно и для информирования правильной диагностики.

5.9. Память пользовательских файлов

ЦП серии Nexto имеют область памяти, предназначенную для общего хранения данных, другими словами, пользователь может хранить в памяти ЦП несколько файлов проекта любого формата. Эта область памяти зависит от используемой модели процессора (см. раздел «Память»).

Чтобы использовать эту область, пользователь должен получить доступ к проекту в программном обеспечении MasterTool IEC XE и щелкнуть дерево устройств, расположенное слева от программы. Дважды щелкните элемент «Устройство» и, после выбора ЦП на вкладке «Настройки связи», которая будет открыта, выберите вкладку «Файлы» и нажмите «Обновить» как в столбце файлов компьютера (слева), так и в столбце файлов ЦП (справа). показано на рисунке ниже.

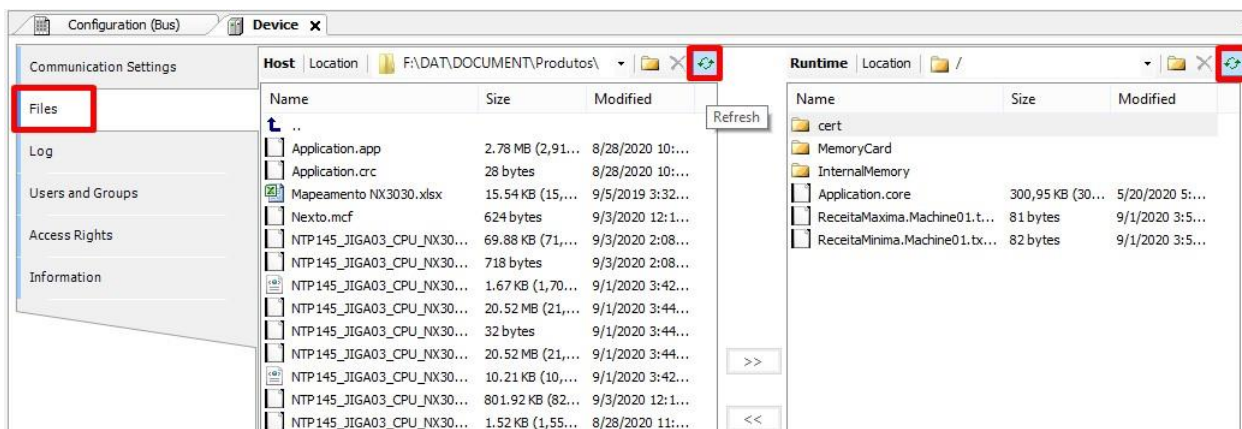


Рисунок 133: Доступ к пользовательским файлам

После обновления столбца файлов ЦП будет показан корневой каталог файлов, хранящихся в ЦП. Затем можно будет выбрать папку, в которую будут перенесены файлы. Папка «InternalMemory» — это папка по умолчанию, которая используется для хранения файлов во внутренней памяти ЦП, поскольку передача файлов в корневой каталог невозможна. При необходимости пользователь может создавать другие папки в корневом каталоге или подпапки внутри папки «InternalMemory».

Папка «MemoryCard» — это каталог, в который монтируется карта памяти, если она вставлена в ЦП. Файлы, передаваемые на «MemoryCard», передаются непосредственно на карту памяти. По мере добавления новых функций в продукт могут появиться некоторые папки, которые пользователь должен игнорировать.

ВНИМАНИЕ

В случае, когда карта памяти вставлена после запуска ЦП, будут запрошены имя пользователя и пароль для выполнения доступа MasterTool IEC XE и/или передачи файлов на карту памяти или наоборот. Стандартный пользователь с правами доступа к ЦП — «Владелец», а пароль по умолчанию для этого пользователя — «Владелец».

5. КОНФИГУРАЦИЯ

Чтобы выполнить передачу файла с микрокомпьютера на ЦП, просто выберите нужный файл в левой колонке и нажмите кнопку «>>», расположенную в центре экрана, как показано на рисунке ниже. Время загрузки зависит от размера файла и времени цикла (выполнения) текущего приложения ЦП и может занять несколько минут.

Пользователю не нужно находиться в режиме выполнения или подключаться к ЦП для выполнения передачи, поскольку он имеет возможность автоматически подключаться, когда пользователь выполняет передачу.

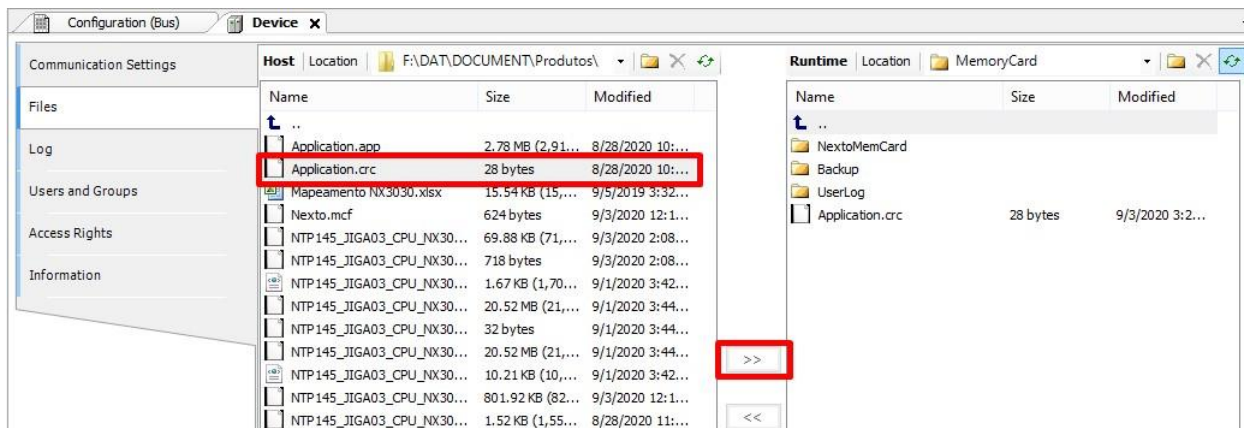





Рисунок 134: Передача файлов

ВНИМАНИЕ

Файлы, содержащиеся в папке проекта, созданного MasterTool IEC XE, имеют специальные имена, зарезервированные системой, поэтому их нельзя передать через вкладку «Файлы». Если пользователь желает перенести проект в пользовательскую память, необходимо сжать папку, а затем загрузить сжатый файл (например, *.zip).

В случае, если необходимо передать документы с ЦП на ПК, на котором установлено программное обеспечение MasterTool IEC XE, пользователь должен выполнить процедуру, очень похожую на описанную ранее, так как файл должен быть выбран из правого столбца и кнопка Нажата «<<», расположенная в центре экрана.

Кроме того, у пользователя есть некоторые параметры работы в области хранения файлов, а именно:

-  Новый каталог: позволяет создать новую папку в пользовательской области памяти.
-  Удалить пункт: разрешает исключать файлы из папок в пользовательской области памяти.
-  Обновить: позволяет обновлять на экране MasterTool IEC XE файлы в области памяти пользователя и на компьютере.

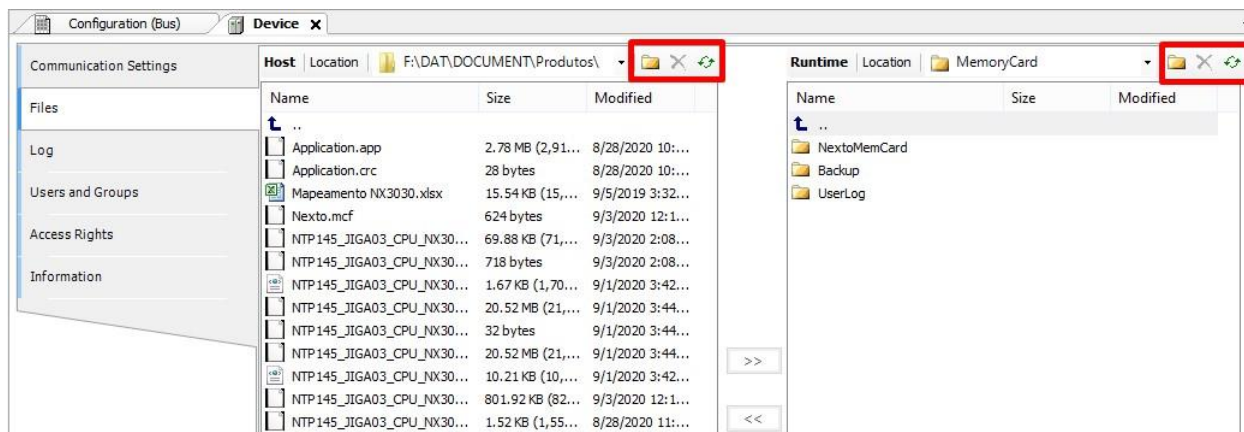


Рисунок 135: Варианты использования

ВНИМАНИЕ

Для процессора в режиме остановки или без приложения скорость передачи во внутреннюю память составляет примерно 150 Кбайт/с.

5.10. Карта памяти

Помимо прочего, процессоры серии Nexto позволяют пользователю использовать карту памяти. Он определяется в соответствии с функциями, описанными в разделе «Интерфейс карты памяти», который хранит, среди прочих файлов, проект и приложение во внутренней памяти ЦП.

Когда карта вставляется в ЦП и представляет тип файла, отличный от FAT32, она автоматически идентифицирует эти файлы и спрашивает пользователя, хочет ли он отформатировать файлы. В отрицательном случае пользователь не может использовать карту, так как она не смонтирована. Появляется сообщение о том, что формат не распознан, и наличие карты также не отображается. Если пользователь решает отформатировать файлы, ЦП требуется несколько минут для выполнения операции, в зависимости от времени цикла (выполнения) приложения, работающего в ЦП. Как только карта памяти установлена, ЦП считывает ее общую информацию, что замедляет доступ к карте памяти в первые несколько минут. Эта процедура выполняется только при вставленной карте или в случае сброса ЦП.

ВНИМАНИЕ

Карту памяти рекомендуется форматировать непосредственно в процессоре Nexto во избежание возможных проблем при использовании, увеличения времени монтажа или даже некорректной работы.

Не рекомендуется извлекать карту памяти или обесточивать процессор во время форматирования или передачи файлов, так как это может привести к потере данных, а также к необратимым повреждениям.

5.10.1. Подготовка проекта

Чтобы использовать эту функциональность, во время конфигурации проекта в программном обеспечении MasterTool IEC XE пользователь должен включить опцию копирования проекта ЦП на карту памяти и/или карты памяти в ЦП и настроить пароли. Эти пароли будут запрошены CPU при выполнении соответствующей передачи. Информацию о таблице см. в разделе Параметры проекта.

ВНИМАНИЕ

Если ЦП не имеет приложения, будет доступно меню «Карта памяти», позволяющее перенести проект с карты памяти на ЦП без какой-либо предварительной подготовки ЦП.

Для использования функции необходимо выполнить следующие шаги.

Перейдите в меню «Онлайн» и выполните команду «Создать загрузочное приложение», помня, что вы не можете войти в ЦП для выполнения этой процедуры. После выполнения этой команды в папке, в которой сохранен проект, создаются два файла. Один с расширением «приложение» и один с расширением «.src».

После создания файлов в предыдущем пункте необходимо перейти к настройкам общих параметров ЦП и нажать кнопку Карта памяти.... Откроется новый экран, как показано на рисунке ниже. На этом экране вы должны активировать желаемую(ые) операцию(ии) передачи и, при необходимости, установить пароль(и) только с числовыми символами. Использование пароля не требуется.

Чтобы завершить операцию установки, вы должны нажать кнопку «Найти файл...», а затем найти файл с расширением «.src», созданный на предыдущем шаге.

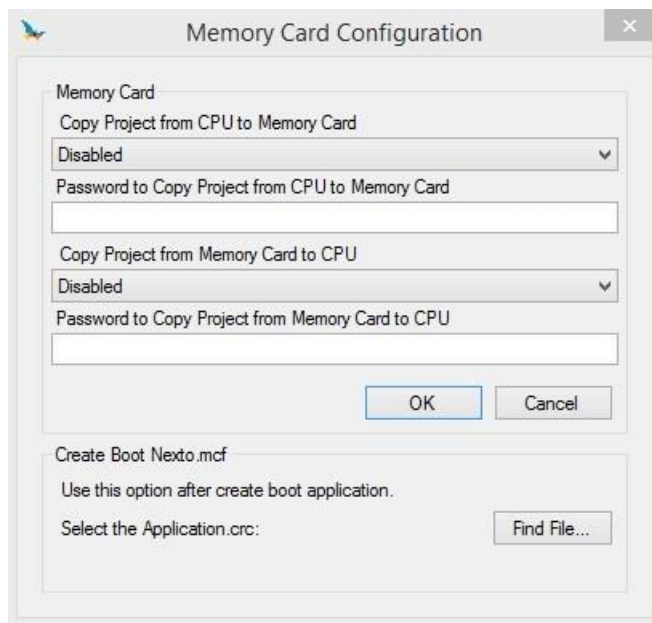


Рисунок 136: Настройки карты памяти

Следуя этим шагам, MasterTool IEC XE отправит все файлы, необходимые для выполнения операций отправки и получения проектов, через карту памяти. Если карта смонтирована, пароль будет записан на нее. В противном случае пароль, установленный в MasterTool, будет запрошен, если пользователь попытается передать проект CPU на карту.

5.10.2. Передача проекта

Чтобы перенести проект с ЦП на карту памяти или наоборот, пользователь, помимо включения в ПО MasterTool IEC XE возможности использования функциональности, должен будет войти в меню Карта памяти в ЦП, используя диагностический ключ, и выбрать желаемый вариант трансфера.

ВНИМАНИЕ

Перенос проекта на карту памяти должен производиться только с помощью диагностического ключа ЦП.

После этого вам будет предложено ввести пароль, если пользователь установил его во время установки приложения. Затем при коротком нажатии на клавишу диагностики цифры увеличиваются, а при длительном нажатии подтверждаются. В подтвержденной шестой цифре ЦП будет состоять из пароля и запускать процесс.

После переноса карты памяти в ЦП, если имеется приложение RUN, оно будет оставаться в состоянии STOP по соображениям безопасности. Чтобы перевести ЦП в режим RUN, его необходимо перезагрузить.

Когда пароли как приложения, которое находится в ЦП, так и приложения, которое находится на карте памяти, одинаковы, не требуется вводить пароли в меню ЦП для выполнения переноса приложений. Дополнительные сведения об использовании клавиши диагностики см. в разделе Диагностика одним нажатием.

Чтобы извлечь карту памяти, просто нажмите и удерживайте клавишу MS и подождите, пока значок карты памяти не исчезнет с экрана состояния графического дисплея.

ВНИМАНИЕ

Если карта памяти извлечена без размонтирования через меню ЦП, во время передачи файла этот процесс может привести к потере данных карты, а также повреждению файлов на ней. Этот процесс может вызвать необходимость форматирования другой карты, когда она будет снова вставлена в ЦП.

ВНИМАНИЕ

Если в корне карты памяти есть какой-либо файл с именем «NextoMemCard» или «Backup», он будет удален для создания системных папок с таким же именем, используемых ЦП для

5. КОНФИГУРАЦИЯ

хранения приложения проекта и архива проекта. Папки с такими именами не будут перезаписаны.

5.10.3. Доступ MasterTool

Доступ к карте памяти связан с тем же экраном пользовательской памяти в программном обеспечении MasterTool IEC XE, поскольку он смонтирован в папке под названием MemoryCard. Папки NextoMemCard и Backup создаются на карте памяти каждый раз, когда последняя вставляется в ЦП. Если эти папки уже существуют, система распознает их и не будет перезаписывать папки.

В NextoMemCard на карте памяти вы найдете файлы приложения, в этом окне у вас еще есть возможность сохранить ваш проект в предпочитаемом каталоге (если вы отправили исходный код). В MasterTool в опции "Файл / Архив проекта / Извлечь архив..." вы можете открыть в MasterTool сохраненное приложение, которое находится в выбранной ранее директории.

Папка Backup не используется пользователем.

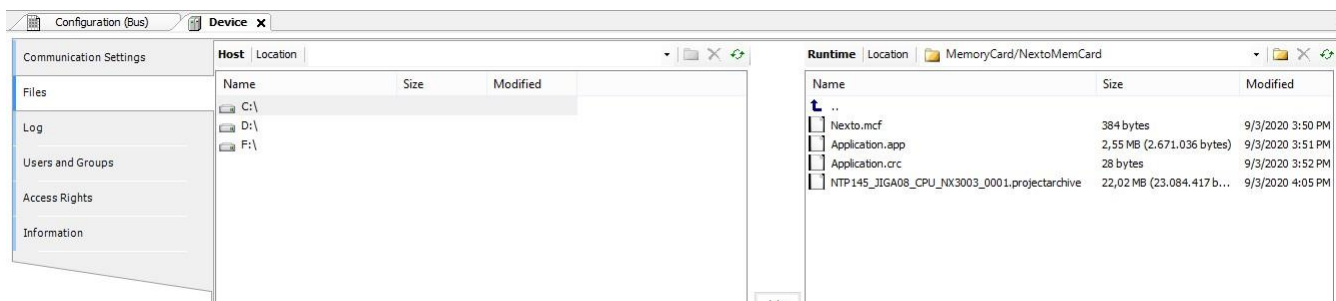


Рисунок 137: Каталог с картой памяти, вставленной вместе с проектом

ВНИМАНИЕ

Время передачи файлов зависит от разницы во времени интервала за вычетом среднего времени выполнения задачи(ов) в исполнении (доступное время до следующего цикла задачи), то есть чем больше эта разница для каждой задачи в приложении, тем быстрее будет быть переносом данных с карты памяти на CPU/MasterTool IEC XE или наоборот. Передача файлов на карту памяти будет медленнее, чем передача во внутреннюю память ЦП. Для ЦП в режиме остановки или без приложения скорость передачи близка к 100 Кбайт/с.

5.11. Информационное и конфигурационное меню процессора

Доступ к информационному меню, конфигурации процессора Nexto и подробной диагностике доступен через уровни, а для доступа к информации меню, изменения уровня и изменения любой конфигурации требуется долгое нажатие на кнопку диагностики и для навигации по элементам на того же уровня требуется короткое нажатие на кнопку диагностики. См. раздел One Touch Diag, чтобы проверить работу и различия между типами касания кнопки диагностики.

В таблице ниже показаны уровни меню и каждый тип экрана, доступный в ЦП, если они являются информативными, конфигурируемыми или возвращают уровень.

Уровень 1	Уровень 2	Уровень 3	Тип
HARDWARE	TEMPERATURE	-	Информативный
	CONTRAST	CONTRAST LEVEL	Настраиваемый
	DATE AND TIME	-	Информативный

5. КОНФИГУРАЦИЯ

	BACK	-	Уровень возврата
LANGUAGES	ENGLISH	>ENGLISH	Настраиваемый
	PORTUGUES	>PORTUGUES	Настраиваемый
	ESPANOL	>ESPANOL	Настраиваемый
	BACK	-	Уровень возврата
NETWORK	NET 1 IP ADDR.		Информативный
	NET 1 MASK		Информативный
	NET 2 IP ADDR.	-	Информативный
	NET 2 MASK		Информативный
	BACK		Уровень возврата
REDUNDANCY	PLC ID		Информативный
	REMOTE STATE	-	Информативный
	PROJ.SYNC.		Информативный
	BACK		Уровень возврата
SOFTWARE	FIRMWARE		Информативный
	BOOTLOADER	-	Информативный
	AUX. PROC.		Информативный
	BACK		Уровень возврата
CARTAO DE MEM.	MEMCARD > CPU	CPU PASSWORD	Настраиваемый
	CPU > MEMCARD	MC PASSWORD	Настраиваемый
	FORMAT	CONFIRM ?	Настраиваемый
	BACK	-	Уровень возврата
BACK	-	-	Уровень возврата

Таблица 169: Уровни меню процессора

Примечания:

Карта памяти: Карта памяти доступна в меню только в том случае, если она подключена к процессору Nexto.

Резервирование: меню «РЕЗЕРВИРОВАНИЕ» будет доступно только в том случае, если ЦП NX3030 идентифицирован как резервный.

Пароль: пароль доступа к данным карты памяти необходим только в том случае, если он настроен в программном обеспечении MasterTool IEC XE. Вы не можете редактировать пароль через меню.

Как показано в таблице 169, среди доступных опций для визуализации и изменения находятся основные данные, необходимые пользователю, такие как:

Информация об аппаратных ресурсах:

- ТЕМПЕРАТУРА – внутренняя температура ЦП (пример: 36 C 97 F)
- CONTRAST – настройка контрастности фронтального дисплея ЦП.
- ДАТА И ВРЕМЯ – Дата и время, установленные в ЦП (пример: 2001.01.31 00:00) Изменение языка меню в ЦП:
- PORTUGUES – меняет язык на португальский.

5. КОНФИГУРАЦИЯ

- АНГЛИЙСКИЙ – меняет язык на английский.
- ESPANOL – меняет язык на испанский.

Визуализация информации о сети, установленной в устройстве:

- СЕТЕВОЙ 1 IP-АДРЕС. – Адрес (пример: 192.168.0.1)
- NET 1 MASK – маска подсети (пример: 255.255.255.0)
- NET 2 IP ADDR – адрес (пример: 192.168.0.2)
- NET 2 MASK – маска подсети (пример: 255.255.255.0) Доступ к информации о резервировании ПЛК:
- Идентификатор ПЛК – информирует об идентификации ПЛК при резервировании. Возможная информация:
 - ПЛК А
 - ПЛК Б
- REMOTE STATE – информирует о состоянии удаленного резервного ПЛК. Возможные состояния:
 - АКТИВНЫЙ
 - ОЖИДАНИЕ
 - НЕАКТИВНО
 - НЕ КОНФИГУР.
 - ЗАПУСК
 - НЕДОСТУПНО
- ПРОЕКТ. СИНХР. – Сообщает, включена ли синхронизация проектов
 - ПОДКЛЮЧЕНО
 - НЕ ПОДКЛЮЧЕНО.
 - ОТКЛЮЧЕНО
 - НАЧАТЬ СИНХР.
 - СИНХРОНИЗИРОВАННЫЙ

Информация о версиях ПО:

- ПРОШИВКА – версия программного обеспечения ЦП (пример: 1.0.0.0)
- BOOTLOADER – версия загрузчика процессора (пример: 1.0.0.0)
- ДОП. ПРОК. – Версия вспомогательного процессора ЦП (пример: 1.0.0.0) Доступ к данным карты памяти:
- MEMCARD > CPU – Передача проекта карты памяти в CPU
- CPU > MEMCARD – Перенос проекта CPU на карту памяти
- FORMAT – форматирует карту в файловую систему FAT32.

На приведенном ниже рисунке показан пример работы с меню ЦП Nexto с помощью процедуры настройки меню контрастности на экране состояния. Кроме того, чтобы упростить настройку, можно идентифицировать все уровни экрана и тип касания для навигации по ним, а также изменять другие параметры, такие как язык и карта памяти, используя ту же логику доступа. Короткое касание показывает, что контрастность увеличивается (более четкая), а при следующем касании после максимального значения она возвращается к минимальному значению (менее четкому). Длительное касание показывает подтверждение желаемого контраста и его возврат на предыдущий уровень.

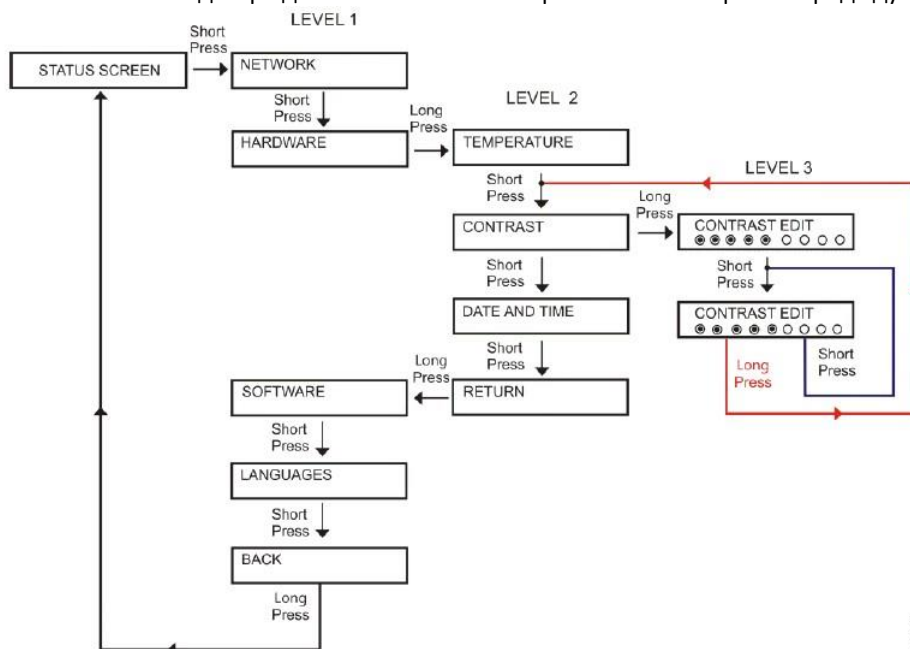


Рисунок 138: Настройка контрастности

Помимо возможности закрытия меню процессоров Nexto долгим касанием экранной диагностической кнопки НАЗАД с уровня 1, существуют и другие условия вывода, описанные ниже:

5. КОНФИГУРАЦИЯ

- Короткое прикосновение в любой момент к другим модулям, существующим на шине, приводит к отключению ЦП от меню и отображению диагностики желаемого модуля. Время простоя на любом уровне превосходит 5 с
-

5.12. Функциональные блоки и функции

5.12.1. Специальные функциональные блоки для последовательных интерфейсов

Специальные функциональные блоки для последовательных интерфейсов обеспечивают локальный доступ (COM 1 и COM 2), а также доступ к удаленным последовательным портам (модули расширения). Таким образом, пользователь может создавать свои собственные протоколы и управлять последовательными портами по своему усмотрению, следуя языкам IEC 61131-3, доступным в программном обеспечении MasterTool IEC XE. Блоки доступны внутри библиотеки NextoSerial, которую необходимо добавить в проект, чтобы можно было их использовать (о выполнении процедуры вставки библиотеки см. Руководство по программированию MasterTool IEC XE – MP399609, раздел Библиотека).

Специальные функциональные блоки для последовательных интерфейсов могут занять несколько циклов (последовательных вызовов) для завершения выполнения задачи. Иногда блок можно выполнить за один цикл, но в общем случае требуется несколько циклов. Выполнение задачи, связанной с блоком, может состоять из множества шагов, некоторые из которых зависят от внешних событий, которые могут быть значительно задержаны. Функциональный блок не может реализовать подпрограммы, занимающие время ожидания этих событий, потому что это загрузит ЦП. Решением может быть создание блокирующих функциональных блоков, но это нецелесообразно, поскольку это увеличивает сложность пользовательского приложения, поскольку обычно многозадачное программирование недоступно. Следовательно, когда ожидается внешнее событие, последовательные функциональные блоки завершают работу, и управление возвращается к основной программе. Обработка задачи продолжается в следующем цикле, другими словами, при следующем вызове блока.

Перед описанием специальных функциональных блоков для последовательных интерфейсов важно знать типы данных, то есть тип данных, используемый блоками.

Тип данных	Варианты	Описание
SERIAL_BAUDRATE	BAUD200	Перечисляет все возможные скорости передачи данных (бит в секунду)
	BAUD300	
	BAUD600	
	BAUD1200	
	BAUD1800	
	BAUD2400	
	BAUD4800	
	BAUD9600	
	BAUD19200	
	BAUD38400	
	BAUD57600	
BAUD115200		
SERIAL_DATABITS	DATABITS_5	Перечисляет все возможности биты данных.
	DATABITS_6	
	DATABITS_7	
	DATABITS_8	
SERIAL_HANDSHAKE	Определяет все возможности сигнала модема для конфигураций:	

5. КОНФИГУРАЦИЯ

RS232_RTS	Управляет портом RS-232C процессора Nexto. Передатчик Включено для начала передачи и Выключено как можно скорее после завершения передачи. Например, может использоваться для управления внешним преобразователем RS232/RS-485.
RS232_RTS_OFF	Управляет портом RS-232C процессора Nexto. Сигнал RTS всегда выключен

Тип данных	Варианты	Описание
	RS232_RTS_ON	Управляет портом RS-232C процессора Nexto. Сигнал RTS всегда включен.
	RS232_RTS_CTS	Управляет портом RS-232C процессора Nexto. В случае, если CTS — Выключено, RTS — Включено. Затем ждет, когда CTS будет Включено, чтобы получить передачу, и RTS перезапустится как можно скорее, в конце передачи. Пример: Управление радиомодемами с помощью одного и того же модемного сигнала.
	RS232_MANUAL	Управляет портом RS-232C процессора Nexto. Пользователь несет ответственность за управление всеми сигналами (RTS, DTR, CTS, DSR, DCD).
SERIAL_MODE	NORMAL_MODE	Последовательная связь Нормальный режим работы.
	EXTENDED_MODE	Serial Communication Extended Режим работы, в котором предоставляется информация о полученном кадре 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

5. КОНФИГУРАЦИЯ

	UART_RX_THRESHOLD	SERIAL_HANDSHAKE.
	MODE	Количество байтов, которое должно быть получено, чтобы сгенерировать новое прерывание UART. Меньшие значения делают TIMESTAMP более точным при использовании РАСШИРЕННОГО РЕЖИМА и минимизируют ошибки переполнения. Однако слишком низкие значения могут вызвать слишком много прерываний и задержку процессора.
	ENABLE_RX_ON_TX	Определено в SERIAL_MODE.
	ENABLE_DCD_EVENT	При значении true весь полученный байт во время передачи будет выгружен, а не отправлен на линию RX. Используется для отключения полнодуплексного режима в интерфейсе RS-422.
	ENABLE_CTS_EVENT	При значении true генерирует внешнее событие при изменении DCD.
SERIAL_PARITY	PARITY_NONE	Перечислите все возможности четности.
	PARITY_ODD	
	PARITY_EVEN	
	PARITY_MARK	

Тип данных	Варианты	Описание
	PARITY_SPACE	
SERIAL_PORT	COM 1	Список всех доступных последовательных портов (COM10, COM11, COM12, COM13, COM 14, COM 15, COM 16, COM

5. КОНФИГУРАЦИЯ

	COM 2	17, COM 18 и COM 19 – модули расширения).
SERIAL_RX_CHAR_EXTENDED	Определяет символ в очереди RX в расширенном режиме.	
	RX_CHAR	Байт данных.
	RX_ERROR	Код ошибки.
	RX_TIMESTAMP	Молчание из-за предыдущего символа или из-за другого события, которое произошло до этого символа (конфигурация последовательного порта, завершение передачи).
SERIAL_RX_QUEUE_STATUS	В нем есть несколько полей, которые предоставляют информацию о статусе/ошибке очереди RX, используемую при использовании обычного формата (без информации об ошибках и метках времени):	
	RX_FRAMING_ERRORS	Счетчик ошибок кадра: неправильное формирование символов – нет стопового бита, неправильная скорость передачи данных, среди прочего – после настройки последовательного порта. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_PARITY_ERRORS	Счетчик ошибок четности, начиная с конфигурации последовательного порта. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_BREAK_ERRORS	Счетчик ошибок прерывания, поскольку конфигурация последовательного порта, другими словами, активная строка выше, чем время символа. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_FIFO_OVERRUN_ERRORS	Счетчик ошибок переполнения FIFO RX, поскольку конфигурация последовательного порта, другими словами, ошибка в настроенном пороге FIFO RX. Возвращается к нулю, когда достигает максимального значения (65535).

5. КОНФИГУРАЦИЯ

	RX_QUEUE_OVERRUN_ERRORS	Счетчик ошибок переполнения очереди RX, другими словами, превышено максимальное количество символов (1024) и происходит перезапись данных. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_ANY_ERRORS	Суммируйте последние 5 счетчиков ошибок (кадр, четность, прерывание, переполнение RX FIFO, переполнение очереди RX).
	RX_REMAINING	Количество символов в очереди RX.

Тип данных	Варианты	Описание
SERIAL_STATUS		Список кодов критических ошибок, которые могут быть возвращены последовательным функциональным блоком. Каждый блок возвращает определенные ошибки, которые будут описаны ниже:
	NO_ERROR	Нет ошибки.
	ILLEGAL_*	Вернуть параметры с недопустимыми значениями или вне допустимого диапазона: - SERIAL_PORT - SERIAL_MODE - BAUDRATE - DATA_BITS - PARITY - STOP_BITS - HANDSHAKE - UART_RX_THRESHOLD - TIMEOUT - TX_BUFF_LENGTH - HANDSHAKE_METHOD - RX_BUFF_LENGTH
	PORT_BUSY	Указывает, что последовательный порт используется другим экземпляром
	HW_ERROR_UART	Обнаружена аппаратная ошибка в UART.
	HW_ERROR_REMOTE	Аппаратная ошибка при обмене данными с удаленным последовательным портом.

5. КОНФИГУРАЦИЯ

	CTS_TIMEOUT_ON	Тайм-аут ожидания включения CTS в рукопожатии RS-232 RTS/CTS в блоке SERIAL_TX.
	CTS_TIMEOUT_OFF	Тайм-аут ожидания отключения CTS, в рукопожатии RS-232 RTS/CTS, в блоке SERIAL_TX.
	TX_TIMEOUT_ERROR	Тайм-аут ожидания окончания передачи в SERIAL_TX.
	RX_TIMEOUT_ERROR	Тайм-аут при ожидании всех символов в блоке SERIAL_RX или блоке SERIAL_RX_EXTENDED.
	FB_SET_CTRL_NOT_ALLOWED	Блок SET_CTRL нельзя использовать, если квитирование отличается от RS232_MANUAL.
	FB_GET_CTRL_NOT_ALLOWED	Блок GET_CTRL нельзя использовать, если рукопожатие отличается от RS232_MANUAL.
	FB_SERIAL_RX_NOT_ALLOWED	SERIAL_RX недоступен для очереди RX, расширенный режим.
	FB_SERIAL_RX_EXTENDED_NOT_ALLOWED	SERIAL_RX_EXTENDED недоступен для очереди RX в обычном режиме.
	DCD_INTERRUPT_NOT_ALLOWED	Прерывание по сигналу DCD не может быть Включено, если последовательный порт не имеет соответствующего контакта.
Тип данных	Варианты	Описание
	CTS_INTERRUPT_NOT_ALLOWED	Прерывание по сигналу CTS не может быть Включено, если рукопожатие отличается от RS232_MANUAL или если последовательный порт не имеет соответствующего контакта.
	DSR_INTERRUPT_NOT_ALLOWED	Прерывание по сигналу DSR не может быть Включено, если последовательный порт не имеет соответствующего контакта. (ЦП Nexto не имеют этого сигнала в локальных портах)
	NOT_CONFIGURED	Функциональный блок нельзя использовать до настройки последовательного порта.
	INTERNAL_ERROR	Указывает, что в последовательном порту возникла внутренняя проблема.
SERIAL_STOPBITS	STOPBITS_1	

5. КОНФИГУРАЦИЯ

STOPBITS_2	Перечислите все возможности Стоп-биты.
STOPBITS_1_5	

Таблица 170: Типы данных последовательных функциональных блокам

5.12.1.1. SERIAL_CFG

Этот функциональный блок используется для настройки и инициализации нужного последовательного порта. После вызова блока каждая очередь RX и TX, связанная с последовательными портами, и FIFO RX и TX перезапускаются.



Рисунок 139: Блок последовательной конфигурации

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
PARAMETERS	SERIAL_PARAMETERS	Эта структура определяет параметры конфигурации последовательного порта, как описано в типе данных SERIAL_PARAMETERS.

Таблица 171: Внутренние параметры SERIAL_CFG

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

5. КОНФИГУРАЦИЯ

STATUS	SERIAL_STATUS	<p>- В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS:</p> <ul style="list-style-type: none"> - NO_ERROR - ILLEGAL_SERIAL_PORT - ILLEGAL_SERIAL_MODE - ILLEGAL_BAUDRATE - ILLEGAL_DATA_BITS - ILLEGAL_PARITY - ILLEGAL_STOP_BITS - ILLEGAL_HANDSHAKE - ILLEGAL_UART_RX_THRESHOLD - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - DCD_INTERRUPT_NOT_ALLOWED - CTS_INTERRUPT_NOT_ALLOWED - DSR_INTERRUPT_NOT_ALLOWED
--------	---------------	---

Таблица 172: Параметры вывода SERIAL_CFG

Пример использования на языке ST, после вставки библиотеки Nexto Serial в проект:

```

PROGRAM UserPrg
VAR
Config: SERIAL_CFG;
Port: SERIAL_PORT := COM1;
Parameters: SERIAL_PARAMETERS := (BAUDRATE := BAUD9600,
DATABITS := DATABITS_8,
STOPBITS := STOPBITS_1,
PARITY := PARITY_NONE,
HANDSHAKE := RS232_RTS,
UART_RX_THRESHOLD := 8, MODE
:=NORMAL_MODE,
ENABLE_RX_ON_TX := FALSE,
ENABLE_DCD_EVENT := FALSE,
ENABLE_CTS_EVENT := FALSE);
Status: SERIAL_STATUS;
END_VAR //ВХОДЫ:
Config.REQUEST := TRUE; Config.PORT := Port;
    
```

5. КОНФИГУРАЦИЯ

```

Config.PARAMETERS := Parameters; //ФУНКЦИЯ:
Config(); //ВЫХОДЫ:
Config.DONE;
Config.EXEC;
Config.ERROR;
Status := Config.STATUS; // Если нужно исправить ошибку.

```

5.12.1.2. SERIAL_GET_CFG

Функциональный блок используется для захвата желаемой конфигурации последовательного порта.



Рисунок 140: Блок для захвата последовательной конфигурации

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 173: Внутренние параметры SERIAL_GET_CFG

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
Внешние параметры	Тип	Описание

5. КОНФИГУРАЦИЯ

STATUS	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
PARAMETERS	SERIAL_PARAMETERS	Эта структура получает параметры конфигурации последовательного порта, как описано в типе данных SERIAL_PARAMETERS.

Таблица 174: Внешние параметры SERIAL_GET_CFG

Пример использования на языке ST, после вставки библиотеки в проект:

```

PROGRAM UserPrg
VAR
  GetConfig: SERIAL_GET_CFG;
  Port: SERIAL_PORT := COM1;
  Parameters: SERIAL_PARAMETERS;
  Status: SERIAL_STATUS;
END_VAR // ВХОДЫ:
  GetConfig.REQUEST := TRUE;
  GetConfig.PORT := Port; //ФУНКЦИЯ:
  GetConfig(); // ВЫХОДЫ:
  GetConfig.DONE;
  GetConfig.EXEC;
  GetConfig.ERROR;
  Status := GetConfig.STATUS; // Если надо исправить ошибку.
  Parameters := GetConfig.PARAMETERS; // Получить параметры желаемого последовательного порта.

```

5.12.1.3. SERIAL_GET_CTRL

Этот функциональный блок используется для считывания управляющих сигналов CTS, DSR и DCD, если они доступны в последовательном порту. При отсутствии управляющих сигналов будет возвращено ложное значение



Рисунок 141: Блок, используемый для визуализации управляющих сигналов

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 175: Внутренние параметры SERIAL_GET_CTRL

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: <ul style="list-style-type: none"> - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - FB_GET_CTRL_NOT_ALLOWED - NOT_CONFIGURED
CTS_VALUE	BOOL	Значение, считанное в управляющем сигнале CTS.
DSR_VALUE	BOOL	Значение, считанное в управляющем сигнале DSR.
DCD_VALUE	BOOL	Значение, считанное в управляющем сигнале DCD.

Таблица 176: Внешние параметры SERIAL_GET_CTRL

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```
PROGRAM UserPrg
```


5. КОНФИГУРАЦИЯ

```

VAR
Get_Control: SERIAL_GET_CTRL;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR //ВХОДЫ:
Get_Control.REQUEST := TRUE;
Get_Control.PORT := Port; //ФУНКЦИЯ:
Get_Control(); //ВЫХОДЫ:
Get_Control.DONE;
Get_Control.EXEC;
Get_Control.ERROR;
Status := Get_Control.STATUS;
Get_Control.CTS_VALUE;
Get_Control.DSR_VALUE;
Get_Control.DCD_VALUE;

// Если нужно исправить ошибку.
    
```

5.12.1.4. SERIAL_GET_RX_QUEUE_STATUS

Этот блок используется для чтения некоторой информации о статусе очереди RX, специально разработанной для нормального режима, но его также можно использовать в расширенном режиме.



Рисунок 142: Блок, используемый для визуализации статуса очереди RX

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 177: Внутренние параметры SERIAL_GET_RX_QUEUE_STATUS

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
Внешние параметры	Тип	Описание

5. КОНФИГУРАЦИЯ

ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
RXQ_STATUS	SERIAL_RX_QUEUE_STATUS	Возвращает статус/ошибку очереди RX, как описано в типе данных SERIAL_RX_QUEUE_STATUS.

Таблица 178. Выходные параметры SERIAL_GET_RX_QUEUE_STATUS

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```

PROGRAM UserPrg
VAR
Get_Status: SERIAL_GET_RX_QUEUE_STATUS;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
Status_RX: SERIAL_RX_QUEUE_STATUS;
END_VAR //ВХОДЫ:
Get_Status.REQUEST := TRUE;
Get_Status.PORT := Port; //ФУНКЦИЯ:
Get_Status(); //ВЫХОДЫ:
Get_Status.DONE;
Get_Status.EXEC;
Get_Status.ERROR;
Status := Get_Status.STATUS; //Если нужно обработать ошибку.
Status_RX := Get_Status.RXQ_STATUS; //Если нужно исправить ошибку RX.

```

5.12.1.5. SERIAL_PURGE_RX_QUEUE

Этот функциональный блок используется для очистки очереди RX последовательного порта, локальной и удаленной. UART RX FIFO также перезапускается.

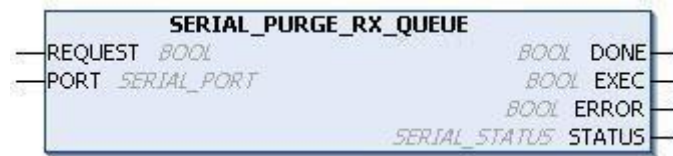


Рисунок 143: Блок, используемый для очистки очереди RX

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 179: Входные параметры SERIAL_PURGE_RX_QUEUE

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED

Таблица 180: Внешние параметры SERIAL_PURGE_RX_QUEUE

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

5. КОНФИГУРАЦИЯ

```

PROGRAM UserPrg
VAR
Purge_Queue: SERIAL_PURGE_RX_QUEUE;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR //ВХОДЫ:
Purge_Queue.REQUEST := TRUE;
Purge_Queue.PORT := Port; //ФУНКЦИЯ:
Purge_Queue(); //ВЫХОДЫ:
Purge_Queue.DONE;
Purge_Queue.EXEC;
Purge_Queue.ERROR;
Status := Purge_Queue.STATUS; //Если нужно исправить ошибку
    
```

5.12.1.6. SERIAL_RX

Этот функциональный блок используется для получения буфера последовательного порта в обычном режиме очереди приема. В этом режиме каждый символ в очереди RX занимает один байт, который содержит полученные данные, сохраняя 5, 6, 7 или 8 бит, в соответствии с конфигурацией последовательного интерфейса.



Рисунок 144: Блок, используемый для чтения значений буфера приема

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
RX_BUFFER_POINTER	POINTER TO BYTE	Указатель массива байтов для получения значений буфера.
RX_BUFFER_LENGTH	UINT	Укажите ожидаемое количество символов в массиве байтов. В случае, если доступно больше ожидаемых байтов, из массива байтов будет прочитано только ожидаемое количество, остальное будет оставлено в очереди RX (максимальный размер равен 1024 символам).

RX_TIMEOUT	UINT	Укажите время ожидания для получения ожидаемого количества символов. В случае, если он меньше необходимого для приема символов, будет указан вывод RX_TIMEOUT_ERROR из параметра STATUS. Когда указанное значение в мс равно нулю, функция вернет данные из буфера.
------------	------	---

Таблица 181: Внутренние параметры SERIAL_RX

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: <ul style="list-style-type: none"> - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_RX_BUFF_LENGTH - RX_TIMEOUT_ERROR - FB_SERIAL_RX_NOT_ALLOWED - NOT_CONFIGURED
RX_RECEIVED	UINT	Возвращает полученное количество символов. Это число может быть в пределах нуля и настроенного значения в RX_BUFFER_LENGTH. Если оно меньше, функциональный блок укажет на ошибку.
RX_REMAINING	UINT	Возвращает количество символов, которые все еще находятся в очереди RX после выполнения функционального блока.

Таблица 182: Внешние параметры SERIAL_RX

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```

PROGRAM UserPrg
VAR
Receive: SERIAL_RX;
Port: SERIAL_PORT := COM1;
    Buffer_Pointer: ARRAY [0..1023] OF BYTE;           //Макс размер.
Status: SERIAL_STATUS;
END_VAR //ВХОДЫ:
Receive.REQUEST := TRUE;
Receive.PORT := Port;
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;                       //Необходимо исправить ошибку
    Receive.RX_RECEIVED;
    Receive.RX_REMAINING;
    
```

5.12.1.7. SERIAL_RX_EXTENDED

Этот функциональный блок используется для получения буфера последовательного порта с использованием расширенного режима очереди приема, как показано в разделе «Конфигурация последовательных интерфейсов»



Рисунок 145: Блок, используемый для чтения буфера приема

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
RX_BUFFER_POINTER	POINTER TO SERIAL_RX_CHAR_EXTENDED	Указатель массива SERIAL_RX_CHAR_EXTENDED для получения значений буфера.

RX_BUFFER_LENGTH	UINT	Укажите ожидаемый номер символа в массиве SERIAL_RX_CHAR_EXTENDED. В случае, если доступно больше ожидаемых байтов, из массива байтов будет прочитано только ожидаемое количество, остальное будет оставлено в очереди RX (максимальный размер равен 1024 символам).
RX_TIMEOUT	UINT	Укажите время ожидания для получения ожидаемого количества символов. В случае, если он меньше необходимого для приема символов, будет указан вывод RX_TIMEOUT_ERROR из параметра STATUS. Когда указанное значение в мс равно нулю, функция вернет данные из буфера.

Таблица 183: Внутренние параметры SERIAL_RX_EXTENDED

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
STATUS	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: <ul style="list-style-type: none"> - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_RX_BUFF_LENGTH - RX_TIMEOUT_ERROR - FB_SERIAL_RX_EXTENDED_NOT_ALLOWED - NOT_CONFIGURED

5. КОНФИГУРАЦИЯ

RX_RECEIVED	UINT	Возвращает полученное количество символов. Это число может быть в пределах нуля и настроенного значения в RX_BUFFER_LENGTH. Если оно меньше, функциональный блок укажет на ошибку.
RX_REMAINING	UINT	Возвращает количество символов, которые все еще находятся в очереди RX после выполнения функционального блока.
RX_SILENCE	UINT	Возвращает время молчания в очереди RX, измеренное с момента завершения приема последнего символа. Единица времени составляет 10 мкс. Этот тип выходного параметра важен для определения времени молчания в таких протоколах, как MODBUS RTU. Это может быть не время молчания после последнего полученного символа этим функциональным блоком, так как это верно только в том случае, если

Таблица 184: Внешние параметры SERIAL_RX_EXTENDED

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```

PROGRAM UserPrg
VAR
Receive_Ex: SERIAL_RX_EXTENDED;
Port: SERIAL_PORT := COM1; Buffer_Pointer: ARRAY [0..1023] OF
SERIAL_RX_CHAR_EXTENDED;
Status: SERIAL_STATUS;
END_VAR //ВХОДЫ:
Receive_Ex.REQUEST := TRUE;
Receive_Ex.PORT := Port;
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;
Status := Receive_Ex.STATUS; //При необходимости исправить ошибку
Receive_Ex.RX_RECEIVED;
Receive_Ex.RX_REMAINING;
Receive_Ex.RX_SILENCE;

```

5.12.1.8. SERIAL_SET_CTRL

Этот блок используется для записи управляющих сигналов (RTS и DTR), когда они доступны в последовательном порту. Он также может установить состояние занятости для передачи с помощью параметра BREAK, и его можно использовать только в том случае, если сигнал модема настроен для RS232_MANUAL.



Рисунок 146: Блок записи управляющих сигналов

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
RTS_VALUE	BOOL	Значение, записываемое в сигнал RTS.
RTS_EN	BOOL	Разрешает запись параметра RTS_VALUE.
DTR_VALUE	BOOL	Значение, записываемое в сигнал DTR.
DTR_EN	BOOL	Разрешает запись параметра DTR_VALUE.
BREAK	BOOL	Если это так, включает логический 0 (занято) в линии передачи.

Таблица 185: Внутренние параметры SERIAL_SET_CTRL

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

5. КОНФИГУРАЦИЯ

STATUS	SERIAL_STATUS	<p>В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы:</p> <ul style="list-style-type: none"> - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - FB_SET_CTRL_NOT_ALLOWED - NOT_CONFIGURED
--------	---------------	--

Таблица 186: Внешние параметры SERIAL_SET_CTRL

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```

PROGRAM UserPrg
VAR
Set_Control: SERIAL_SET_CTRL;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR

//ВХОДЫ:
Set_Control.REQUEST := TRUE;

Set_Control.PORT := Port;
Set_Control.RTS_VALUE := FALSE;
Set_Control.RTS_EN := FALSE;
Set_Control.DTR_VALUE := FALSE;
Set_Control.DTR_EN := FALSE;
Set_Control.BREAK := FALSE; //Функция:
Set_Control(); //ВЫХОДЫ:
Set_Control.DONE;
Set_Control.EXEC;
Set_Control.ERROR;
Status := Set_Control.STATUS; //При необходимости исправить ошибку
    
```

5.12.1.9. SERIAL_TX

Этот функциональный блок используется для передачи буфера данных через последовательный порт и завершается только после того, как все байты были переданы или по истечении времени ожидания (генерация ошибок).

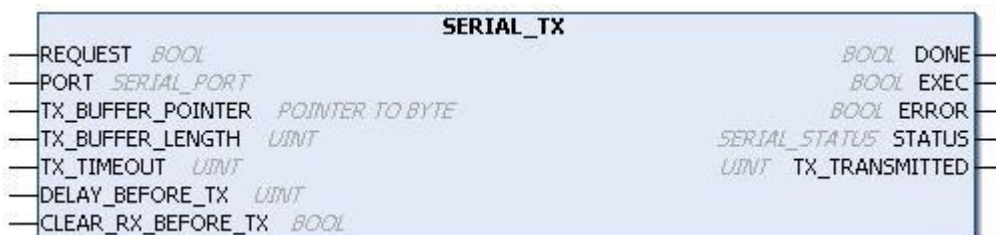


Рисунок 147: Блок передачи значений по серийному номеру

Внутренние параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
TX_BUFFER_POINTER	POINTER TO BYTE	Указатель массива байтов для передачи значений буфера.
TX_BUFFER_LENGTH	UINT	Укажите ожидаемое количество символов в передаваемом массиве байтов (максимальный размер очереди TX — 1024 символа).
TX_TIMEOUT	UINT	Укажите тайм-аут для завершения передачи, включая фазу квитирования. 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. Эта переменная может использоваться при
Внутренние параметры	Тип	Описание
CLEAR_RX_BEFORE_TX	BOOL	значении «TRUE» очередь RX и UART FIFO RX стираются перед началом передачи. Такое поведение типично для полудуплексных протоколов ведущий/ведомый.

Таблица 187: Внутренние параметры SERIAL_TX

Внешние параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

5. КОНФИГУРАЦИЯ

STATUS	SERIAL_STATUS	<p>В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы::</p> <ul style="list-style-type: none"> - 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	<p>Возвращает номер переданного байта, который должен быть равен TX_BUFFER_LENGTH, но может быть меньше, если во время передачи произошла ошибка.</p>

Таблица 188: Внешние параметры SERIAL_TX

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```

PROGRAM UserPrg
VAR
Transmit: SERIAL_TX;
Port: SERIAL_PORT := COM1;
Buffer_Pointer: ARRAY [0..9] OF BYTE := [0,1,2,3,4,5,6,7,8,9];
Status: SERIAL_STATUS;
END_VAR //ВХОДЫ:
Transmit.REQUEST := TRUE;
Transmit.PORT := Port;
Transmit.TX_BUFFER_POINTER := ADR(Buffer_Pointer);
Transmit.TX_BUFFER_LENGTH := 10;
Transmit.TX_TIMEOUT := 10000;
Transmit.DELAY_BEFORE_TX := 1000;
Transmit.CLEAR_RX_BEFORE_TX := TRUE; //ФУНКЦИЯ:
Transmit(); //ВЫХОДЫ:
Transmit.DONE;
Transmit.EXEC;
Transmit.ERROR;
Status := Transmit.STATUS; //При необходимости исправить ошибку
Transmit.TX_TRANSMITTED;

```

5.12.2. Обновление входов и выходов

5. КОНФИГУРАЦИЯ

По умолчанию локальная шина и встроенный ввод-вывод ЦП обновляются при каждом цикле выполнения MainTask. Функции обновления позволяют асинхронно обновлять эти точки ввода-вывода в любой точке кода пользовательского приложения.

Когда функциональные блоки для обновления входов и выходов не используются, обновление выполняется в каждом цикле MainTask.

ВНИМАНИЕ

При запуске ЦП этой серии входы и выходы только обновляются для чтения и подготавливаются для записи при выполнении MainTask. Все другие системные задачи, которые выполняются до MainTask, будут иметь недопустимые входы и выходы.

5.12.2.1. REFRESH_INPUT

Этот функциональный блок используется для обновления заданных входов модуля без необходимости ожидания завершения цикла. Важно отметить, что фильтры, настроенные в MasterTool IEC XE, и время обновления входов модуля должны учитываться во времени эффективного обновления входов в приложении, разработанном пользователем.

ВНИМАНИЕ

Функция REFRESH_INPUT должна использоваться только в MainTask. Для обновления входов в других задачах необходимо выбрать параметр Включить обновление ввода-вывода для каждой задачи. Для получения дополнительной информации об этом параметре см. Таблицу 42.

ВНИМАНИЕ

Функция REFRESH_INPUT не поддерживает входные данные, которые были сопоставлены с символьными переменными. Для правильной работы необходимо, чтобы ввод отображался на переменную в прямом представлении в памяти входных переменных (%I).

ВНИМАНИЕ

Функция REFRESH_INPUT обновляет только прямые переменные %I, которые объявлены на вкладке «Шина: I/O Mapping» модуля, адресуемого в соответствующей стойке/слоте функции. В случае коммуникационных модулей/интерфейсов (MODBUS, Profibus и т. д.) обновление не включает прямые переменные сопоставления устройств.

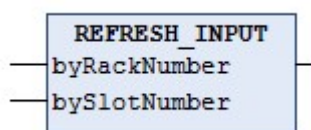


Рисунок 148: Блок для обновления ввода

Внутренний параметр	Тип	Описание
byRackNumber	BYTE	Номер стойки.
bySlotNumber	BYTE	Номер позиции, к которой подключается модуль.

Таблица 189: Входные параметры REFRESH_INPUT

Возможный TYPE_RESULT:

- OK_SUCCESS: успешное выполнение.

5. КОНФИГУРАЦИЯ

- ERROR_FAILED: эта ошибка возвращается, если функция вызывается для модуля, который имеет только выходы, или если параметр Всегда обновлять переменные (находится на экране конфигурации модуля, вкладка Сопоставление ввода-вывода) не отмечен.
- ERROR_NOTSUPPORTED: Вызванная подпрограмма не поддерживается продуктом.
- ERROR_PARAMETER: неверный/неподдерживаемый параметр.
- ERROR_MODULE_ABSENT: Модуль отсутствует в шине.
- ERROR_MODULE_NOTCONFIGURED: Модуль не настроен в приложении.
- ERROR_MODULE_NOTRUNNING: Модуль не запущен (не в рабочем состоянии).
- ERROR_MODULE_COMMFAIL: сбой связи с модулем.
- ERROR_MODULE_NOTFOUND: Модуль не найден в приложении или не поддерживается.

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR
Info: TYPE_RESULT; byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR //ВХОДЫ:
byRackNumber := 0; bySlotNumber := 10;
//ФУНКЦИЯ:
Info := REFRESH_INPUT (byRackNumber, bySlotNumber); //Функция звонка.
// Переменная «Информация» получает возможные ошибки функции.
```

5.12.2.2. REFRESH_OUTPUT

Этот функциональный блок используется для обновления указанных выходов модуля. Нет необходимости ждать окончания цикла. Важно отметить, что время обновления выходов модуля должно быть учтено в эффективном времени обновления выходов в приложении, разработанном пользователем.

ВНИМАНИЕ Функция REFRESH_OUTPUT должна использоваться только в MainTask. Для обновления выходов в других задачах необходимо выбрать параметр Включить обновление ввода-вывода для каждой задачи. Для получения дополнительной информации об этом параметре см. Таблицу 42.
ВНИМАНИЕ <i>REFRESH_OUTPUT</i> функция не поддерживает входные данные, которые были сопоставлены с символьными переменными. Для правильной работы необходимо, чтобы ввод отображался на переменную в прямом представлении в памяти входных переменных
ВНИМАНИЕ <i>REFRESH_OUTPUT</i> функция обновляет только прямые переменные %Q, которые объявлены на вкладке «Шина: сопоставление ввода/вывода модуля, адресованного в соответствующей стойке/слоте функции. В случае коммуникационных модулей/интерфейсов (MODBUS, Profibus и т. д.) обновление не включает

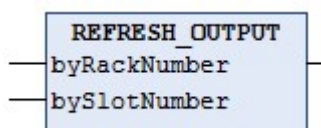


Рисунок 149: Блок для обновления вывода

5. КОНФИГУРАЦИЯ

Внутренние параметры	Тип	Описание
byRackNumber	BYTE	Номер стойки.
bySlotNumber	BYTE	Номер позиции, к которой подключается модуль.

Таблица 190: Входные параметры REFRESH_OUTPUT

Возможный TYPE_RESULT:

OK_SUCCESS: успешное выполнение.

ERROR_FAILED: Эта ошибка возвращается, если функция вызывается для модуля, который имеет только входы, или если параметр Всегда обновлять переменные (находится на экране конфигурации модуля, вкладка Сопоставление ввода-вывода) не отмечен.

ERROR_NOTSUPPORTED: Вызванная подпрограмма не поддерживается продуктом.

ERROR_PARAMETER: неверный/неподдерживаемый параметр.

ERROR_MODULE_ABSENT: Модуль отсутствует в шине.

ERROR_MODULE_NOTCONFIGURED: Модуль не настроен в приложении.

ERROR_MODULE_NOTRUNNING: Модуль не запущен (не в рабочем состоянии).

ERROR_MODULE_COMMFAIL: сбой связи с модулем.

ERROR_MODULE_NOTFOUND: Модуль не найден в приложении или не поддерживается.

Пример использования на языке ST:

```
PROGRAM UserPrg VAR
Info: TYPE_RESULT; byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR //ВХОДЫ:
byRackNumber := 0; bySlotNumber := 10;
//ФУНКЦИЯ:
//Функция звонка
Info := REFRESH_OUTPUT (byRackNumber, bySlotNumber); // Переменная
«Информация» получает возможные ошибки функции.
```

5.12.3. Функциональный блок ПИД-регулятора

ВНИМАНИЕ

Функциональный блок PID, описанный до предыдущей версии L данного руководства, устарел и был удален из этого руководства.

Функциональные блоки PID, PID_INT и PID_REAL, описанные до версии C MP399609, также устарели и были удалены из более новых версий этого руководства. Пользователи, которым требуется описание этих устаревших функциональных блоков по причинам технического обслуживания, должны использовать версию C MP399609.

Функциональные блоки PID, PID_INT и PID_REAL не должны использоваться в новых проектах. Эти функциональные блоки были заменены более новыми функциональными блоками с дополнительными ресурсами, такими как автонастройка и поддержка каскадных, переопределяющих и упреждающих элементов управления. Эти новые функциональные блоки описаны в MU214610 и доступны после версии 1.1.0.0 *NextoPID*.

5.12.4. Сохранение таймера

Сохранение таймера — это функциональный блок, разработанный для приложений, таких как часы производственной линии, которые должны сохранять свое значение и перезапускать отсчет с той же точки в случае

5. КОНФИГУРАЦИЯ

сбоя питания. Значения, сохраненные функциональным блоком, равны нулю только в случае холодного сброса, сброса источника или загрузки нового приложения (см. Руководство пользователя MasterTool IEC XE - MU299609), когда счетчики продолжают работать, даже когда приложение остановлено (Режим остановки).

ВНИМАНИЕ

Важно подчеркнуть, что для правильного функционирования блоков Timer Retain переменные должны быть объявлены как Retain (VAR RETAIN). Также важно отметить, что в режиме симуляции функциональные блоки Timer Retain не работают должным образом из-за того, что для правильного поведения требуется ЦП Nexto.

Три блока, уже имеющиеся в библиотеке NextoStandard программного обеспечения MasterTool IEC XE, описаны ниже (о порядке вставки библиотеки см. Руководство по программированию MasterTool IEC XE — MP399609, раздел «Библиотека»).

5.12.4.1. TOF_RET

Функциональный блок TOF_RET реализует временную задержку для отключения выхода. Когда состояние входа IN изменяется с (TRUE) на (FALSE) или спадающий фронт, будет отсчитываться заданное время PT, и по его окончании выход Q будет переведен в состояние (FALSE). Когда вход IN находится в состоянии логической 1 (TRUE), выход Q остается в том же состоянии (TRUE), даже если это произошло в середине процесса счета. Время PT может быть изменено во время подсчета, так как блок принимает новое значение, если подсчет еще не завершен. На рисунке 150 показан блок TOF_RET, а на рисунке 151 показано его графическое поведение.



Рисунок 150: Блок TOF_RET

Внутренние параметры	Тип	Описание
IN	BOOL	Эта переменная, когда получает задний фронт, включает подсчет блоков.
PT	TIME	Эта переменная определяет предел счета блоков (временную задержку).

Таблица 191: Входные параметры TOF_RET

Внешние параметры	Тип	Описание
Q	BOOL	Эта переменная выполняет задний фронт, когда переменная PT (временная задержка) достигает своего максимального значения.
ET	TIME	Эта переменная показывает текущую временную задержку.

Таблица 192: Выходные параметры TOF_RET

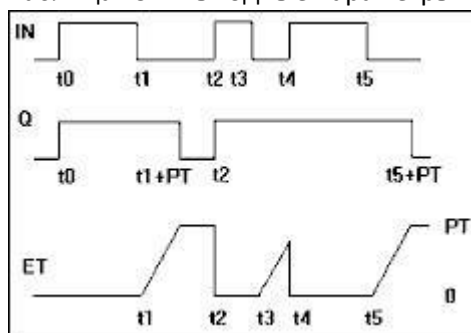


Рисунок 151: Графическое поведение блока TOF_RET

Пример использования на языке ST:

```
PROGRAM UserPrg VAR RETAIN
bStart : BOOL := TRUE;
TOF_RET : TOF_RET; END_VAR

// Когда bStart=FALSE начинает отсчет
TOF_RET( IN := bStart,
PT := T#20S);

// Действия, выполняемые в конце подсчета IF (TOF_RET.Q =
FALSE) THEN bStart := TRUE;
END_IF
```

5.12.4.2. TON_RET

TON_RET реализует временную задержку для включения выхода. Когда состояние входа IN изменяется с (FALSE) на (TRUE) или нарастающий фронт, будет отсчитываться заданное время PT, и в конце этого времени выход Q будет переведен в состояние (TRUE). Когда вход IN находится в состоянии логического 0 (FALSE), выход Q остается в том же состоянии (FALSE), даже если это происходит в середине процесса счета. Время PT может быть изменено во время подсчета, так как блок принимает новое значение, если подсчет еще не завершен. На рис. 152 показан блок TON_RET, а на рис. 153 показано его графическое поведение.



Рисунок 152: Функциональный блок TON_RET

Внутренние параметры	Тип	Описание
IN	BOOL	Эта переменная, когда получает нарастающий фронт, включает подсчет функциональных блоков.
PT	TIME	Эта переменная определяет предел счета блоков (временную задержку).

Таблица 193: Внутренние параметры TON_RET

Внешние параметры	Тип	Описание
Q	BOOL	Эта переменная выполняет нарастающий фронт, когда переменная PT (временная задержка) достигает своего максимального значения.
ET	TIME	Эта переменная показывает текущую временную задержку.

Таблица 194: Внешние параметры TON_RET

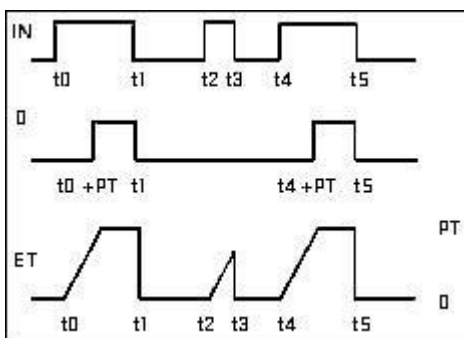


Рисунок 153: TON_RET Графическое поведение блока

Пример использования на языке ST:

```

PROGRAM UserPrg VAR
RETAIN bStart : BOOL;
TON_RET : TON_RET;
END_VAR

// Quando bStart=TRUE начинает отсчет
TON_RET( IN := bStart,
PT := T#20S);

// Действия, выполняемые в конце подсчета IF (TON_RET.Q =
TRUE) THEN bStart := FALSE;
END_IF

```

5.12.4.3. TP_RET

Функциональный блок TP_RET работает как триггер. Таймер, который запускается, когда состояние входа IN изменяется с (FALSE) на (TRUE), то есть передний фронт, он увеличивается до тех пор, пока не будет достигнут предел времени PT. Во время подсчета выход Q равен (TRUE), в противном случае — (FALSE). Время PT может быть изменено во время подсчета, так как блок принимает новое значение, если подсчет еще не завершен. Рисунок 154 изображает TP_RET, а Рисунок 155 показывает его графическое поведение.



Рисунок 154: Функциональный блок TP_RET

Внутренние параметры	Тип	Описание
IN	BOOL	Эта переменная, когда получает нарастающий фронт, включает подсчет функциональных блоков.
PT	TIME	Эта переменная определяет предел счета функциональных блоков (временную задержку).

Таблица 195: Входные параметры TP_RET

Внешние параметры	Тип	Описание
Q	BOOL	Эта переменная истинна во время подсчета, иначе ложна.
ET	TIME	Эта переменная показывает текущую временную задержку.

Таблица 196: Выходные параметры TP_RET

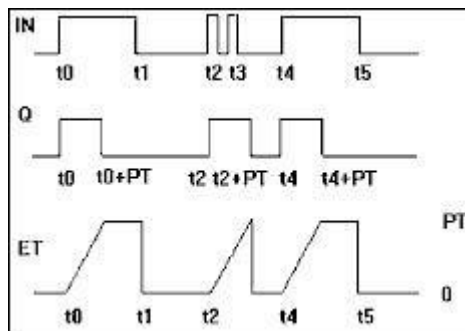


Рисунок 155: Графическое поведение блока TP_RET

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR RETAIN bStart :
BOOL; TP_RET :
TP_RET; END_VAR

// Настроить TP_RET
TP_RET(IN := bStart, PT := T#20S);
бСтарт := FALSE;
// Действия, осуществляемые во время подсчета
ЕСЛИ (TP_RET.Q = TRUE), ТО
// Выполняется, пока счетчик активирован
    
```

5. КОНФИГУРАЦИЯ

ЕЩЕ

```
// Выполняется, когда счетчик деактивирован
```

```
END_IF
```

5.12.5. Нерезервированный таймер

Таймер без резервирования используется в приложениях для резервированного ЦП NX3030, которым требуется таймер в нерезервированной программе полукластера. Этот таймер не использует таймер IEC, поэтому он не будет синхронизирован в случае, если резервный полукластер примет активное состояние, а активный станет резервным.

Три типа блоков, уже имеющихся в библиотеке NextoStandard программного обеспечения MasterTool IEC XE, описываются следующим образом (для выполнения процедуры включения библиотеки см. Руководство по программированию MasterTool IEC XE – MP399608, раздел Библиотека).

5.12.5.1. TOF_NR

Функциональный блок TOF_NR реализует время задержки для отключения выхода, его функционирование и конфигурация аналогичны функциональному блоку TOF_RET, отличаясь только тем, что он не является ни избыточным, ни запоминающим.



Рисунок 156: Функциональный блок TOF_NR

Пример использования на языке ST:

```
PROGRAM NonSkippedPrg VAR
bStart : BOOL := TRUE;
TOF_NR : TOF_NR; END_VAR

// Когда bStart=FALSE начинает подсчет
TOF_NR( IN := bStart,
PT := T#20S);

// Действия, выполняемые в конце подсчета IF (TOF_NR.Q =
FALSE) THEN bStart := TRUE;
END_IF
```

5.12.5.2. TON_NR

Функциональный блок TON_NR реализует время задержки для включения выхода, его функционирование и конфигурация аналогичны функциональному блоку TON_RET, отличие только в том, что он не является ни избыточным, ни сохраняющим.



Рисунок 157: Функциональный блок TON_NR

Пример использования на языке ST:

5. КОНФИГУРАЦИЯ

```

PROGRAM NonSkippedPrg
VAR bStart : BOOL; TON_NR :
TON_NR; END_VAR

// Когда bStart=TRUE начинает подсчет
TON_NR( IN := bStart,
PT := T#20S);

// Действия, выполняемые в конце отсчета IF (TON_NR.Q =
TRUE) THEN bStart := FALSE;
END_IF

```

5.12.5.3. TP_NR

Функциональный блок TP_NR работает как триггер, его функционирование и конфигурация аналогичны функциональному блоку TP_RET, отличие только в том, что он не является ни избыточным, ни запоминающим.



Рисунок 158: Функциональный блок TP_NR

Пример использования на языке ST:

```

PROGRAM NonSkippedPrg VAR bStart : BOOL; TP_NR : TP_NR;
END_VAR

// Настроить TP_NR
TP_NR( IN := bStart,
PT := T#20S);
bStart := FALSE;

// Действия, выполняемые во время подсчета
IF (TP_NR.Q = TRUE) THEN
// Выполняется, пока счетчик активирован
ELSE
// Выполняется, когда счетчик деактивирован END_IF

```

5.12.6. Журнал пользователя

Функция, позволяющая пользователю создавать собственные записи и записывать файлы журнала на карту памяти, установленную в ЦП. Файлы создаются в определенном каталоге карты памяти в формате CSV, что позволяет просматривать их в текстовых редакторах и электронных таблицах. Разделителем был символ точки с запятой. Дополнительные сведения об использовании карты памяти см. в разделе Карта памяти.

Доступны две функции: одна для информации журнала, а другая для удаления всех записей. Ниже приводится описание типов данных, используемых функциями:

Тип данных	Вариант	Описание
USER_LOG_EVENT_TYPES	USER_LOG_EVENT_ERROR	Пользователь может использовать наилучшую индикацию в соответствии с
	USER_LOG_EVENT_DEBUG	
	USER_LOG_EVENT_INFO	

5. КОНФИГУРАЦИЯ

	USER_LOG_EVENT_WARN	серьезностью сообщения журнала.
USER_LOG_MESSAGE		Сообщение журнала длиной не более 150 символов.
USER_LOG_ERROR_CODES	USER_LOG_OK	Операция была проведена успешно.
	USER_LOG_FAILED	Операция не была проведена успешно. Причину сбоя можно проверить в системных журналах – см. раздел Системный журнал.
	USER_LOG_BUFFER_FULL	Сообщения добавляются за пределами возможностей обработки.
	USER_LOG_NO_MEMORY	В то время не было ресурсов для проведения операции.
	USER_LOG_FILE_SYSTEM_ERROR	Произошла ошибка при доступе к карте памяти или нет свободного места. Информацию об ошибках можно проверить в журналах системы – см. раздел Системный журнал.
	USER_LOG_NO_MEMORY_CARD	В процессоре есть карта памяти.
	USER_LOG_MEMORY_CARD_FULL	На карте памяти нет свободного места.
	USER_LOG_PROCESSING	Ресурс занят выполнением последней операции, например удалением всех лог-файлов.

Таблица 197: Тип данных для журнала пользователя

Ниже описаны две функции, доступные в библиотеке LibLogs в MasterTool IEC XE. Чтобы выполнить процедуру вставки библиотеки, см. Руководство по программированию MasterTool IEC – MP399609, раздел Библиотеки.

ВНИМАНИЕ

Пользовательские журналы доступны только до версии 1.3.0.20 процессоров серии Nexto. Так же для использования этой функции необходима версия 1.40 или выше MasterTool IEC XE.

5.12.6.1. UserLogAdd

Эта функция используется для добавления нового сообщения пользовательского журнала, добавляя новую строку в файл журнала на карте памяти. Сообщение должно иметь максимальную длину 150 символов и тип события сообщения. Переменные приложения могут быть зарегистрированы с помощью преобразования в строку и конкатенации с основным сообщением. Информация о дате и времени в формате UTC (отметка времени) автоматически добавляется в сообщение с разрешением в миллисекунды, в котором было зарегистрировано событие. Информация о дате и времени также используется при формировании имен файлов журнала.

Функцию UserLogAdd можно использовать для ввода нескольких сообщений в рамках одной задачи, а также в разных прикладных задачах. Однако независимо от каждого выполнения функции в приложении, выполнения одной и той же задачи или разных задач, все они используют одну и ту же функцию для записи нужных сообщений. По этой причине рекомендуется, чтобы добавление сообщений с использованием функции UserLogAdd в приложении удерживалось каждые 50 мс, чтобы предотвратить повторную перегрузку буфера. Если функция выполняется в периоды короче, чем указано, но соблюдается среднее время 50 мс между каждым добавлением

5. КОНФИГУРАЦИЯ

сообщения в конце интервала для задачи, также предотвращается возврат перегрузки буфера. Чтобы журналы добавлялись корректно, важно соблюдать ограничения по времени при вставке карты и при запуске ЦП, указанные в разделе Карта памяти. После операции функция возвращает параметры для данного типа USER_LOG_ERROR_CODES в виде таблицы 197. Если функция возвращает не USER_LOG_OK, сообщение не было зарегистрировано и функция UserLogAdd должна быть повторно выполнена с желаемым сообщением. В случае возврата последовательных сбоях записи карта памяти может быть повреждена. Замена на исправную карту памяти гарантирует, что последние зарегистрированные сообщения будут записываться на неповрежденную карту, поскольку ЦП не перезагружается.

На рисунке ниже представлена функция UserLogAdd и таблица под входными параметрами:



Рисунок 159: Функция UserLogAdd

Внутренние параметры	Тип	Описание
byEventType	BYTE	Эта переменная указывает тип события журнала, добавляемого в качестве опций для типа данных USER_LOG_EVENT_TYPES.
pszMessage	USER_LOG_MESSAGE	Эта переменная должна содержать набор символов, составляющих сообщение, которое будет добавлено в файл журнала. Сообщение должно содержать не более 150 символов.

Таблица 198: Входные параметры UserLogAdd

Файлы журналов создаются и размещаются на карте памяти в определенном каталоге в зависимости от серийного номера ЦП и установленной версии микропрограммы. Например, для ЦП с серийным номером 445627 и версией микропрограммы 1.4.0.4 расположение, в которое файлы журнала должны быть записаны на карту памяти, — MemoryCard/UserLog/445627/1.4.0.4/.

Имена лог-файлов формируются по дате и времени (метке времени) первого сообщения. За исключением случаев, когда есть проблема с использованием этого имени, например, другой существующий файл с таким же именем, в этой ситуации используются мгновенные дата и время. Имя файла соответствует следующему шаблону: год/месяц/день/час/минута/секунда/миллисекунда.CSV. В случае проблем с доступом к файлу из-за дефектного сектора, не позволяющего продолжить запись, к имени этого файла будет добавлено расширение «поврежден» и будет создан новый файл. Количество журналов на файл не фиксировано и зависит от размера сообщений. Количество создаваемых файлов ограничено 1024 файлами с максимальным размером 1 МБ каждый, поэтому на карте памяти требуется 1 Гб свободного места. При достижении предела в 1024 файла, созданного на карте памяти, во время работы ЦП удаляются самые старые файлы, чтобы сохранить файлы с последними журналами, даже в случаях частичного ручного удаления файлов в каталоге, где файлы находятся. написано.

Просмотр файлов журнала можно выполнять с помощью рабочих листов или обычных текстовых редакторов. Объединенная информация для улучшения визуализации может использовать точку с запятой между строками сообщения для их разделения. Нужно быть осторожным при форматировании ячеек со значениями с плавающей запятой.

Пример использования на языке ST:

5. КОНФИГУРАЦИЯ

```
PROGRAM UserPrg
VAR eLogError : USER_LOG_ERROR_CODES;
    sMessage :USER_LOG_MESSAGE;
END_VAR

IF (m_rTemperature > MAX_TEMPERATURE_ACCEPT) THEN sMessage := 'Temperature
higher than expected: '; sMessage := concat(sMessage,
REAL_TO_STRING(m_rTemperature)); sMessage := concat(sMessage, '°'); eLogError :=
UserLogAdd(USER_LOG_EVENT_WARN, sMessage); // Переменная eLogError получает
возможные ошибки функции.
END_IF
```

Пример содержимого файла журнала: (UserLog-201308271506245738.csv)

```
Model; NX3030
Serial number; 445627
Firmware version; 1.4.0.4
```

```
27/08/2013 15:06:24.5738; WARN; Overtemperature: 25°
27/08/2013 16:37:45.3476; WARN; Overtemperature: 25°
28/08/2013 09:10:55.4201; WARN; Overtemperature: 26°
```

5.12.6.2. UserLogDeleteAll

Функция UserLogDeleteAll выполняет удаление файлов журнала, находящихся в каталоге, созданном специально для ЦП, в который вставлена карта памяти, т. е. удаляются только журналы, содержащиеся в каталоге, названном версией микропрограммы ЦП, который существует в каталоге с ЦП. серийная версия. Удаляемые лог-файлы — это только те файлы, которые существуют на момент монтирования карты памяти и генерируются функцией UserLogAdd. Журналы других процессоров и файлы, добавленные пользователем вручную во время выполнения, не удаляются. На рисунке ниже представлена функция UserLogDeleteAll.

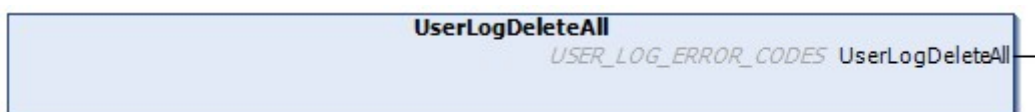


Рисунок 160: Функция UserLogDeleteAll

Пример использования на языке ST:

```
PROGRAM UserPrg VAR
eLogError : USER_LOG_ERROR_CODES;
END_VAR

IF (m_DeleteLogs = TRUE) THEN eLogError :=
UserLogDeleteAll(); m_DeleteLogs := FALSE;
// Переменная eLogError получает возможные ошибки функции.
END_IF
```


5. КОНФИГУРАЦИЯ

Возврат функции UserLogDeleteAll не указывает на завершение операции, а только на подтверждение выполнения, которое может занять много времени, если в каталоге есть сотни лог-файлов. Функция для записи нового пользовательского журнала сейчас недоступна, возвращая параметр USER_LOG_PROCESSING для любой операции. Результат операции также можно проверить в системном журнале.

5.12.7. ClearRtuDiagnostic

Эта функция не поддерживается этой серией процессоров.

5.12.8. ClearEventQueue

Функцию ClearEventQueue, доступную в библиотеке LibRtuStandard, можно использовать, когда необходимо очистить очередь событий ЦП и все установленные драйверы.

Согласно таблице ниже результат выполнения функции будет отображаться в ее выходной переменной..

ВНИМАНИЕ

Функция ClearEventQueue не применяется к очистке очереди службы журнала событий (SOE), описанной в разделе Конфигурация SOE. Функция очищает только очереди событий драйверов, сконфигурированных для коммуникационных интерфейсов (COM и NET) ЦП.

Наименование	ENUM (UDINT)	Описание результата
OK_SUCCESS	0	Успешно
ERROR_FAILED	1	Общая ошибка
ERROR_NOTSUPPORTED	2	Вызванная подпрограмма не поддерживается продуктом
ERROR_PARAMETER	3	Недопустимый/неподдерживаемый параметр
ERROR_MODULE_ABSENT	16	Модуль отсутствует в шине
ERROR_MODULE_NOTCONFIGURED	17	Модуль не настроен в приложении
ERROR_MODULE_NOTRUNNING	18	Модуль не запущен (не в рабочем состоянии)
ERROR_MODULE_COMMFAIL	19	Сбой связи с модулем
ERROR_MODULE_NOTFOUND	20	Модуль не найден в приложении или не поддерживается

Таблица 199: Результаты функции ClearEventQueue

Используя пример на языке ST, где вызов функции будет очищать очередь событий и, следовательно, сбрасывать диагностику использования очереди событий драйверов связи T_DIAG_DNP_SERVER_1.tClient_*.tQueueDiags.wUsage:

```
PROGRAM UserPrg
VAR
    ClearEventQueueStatus : TYPE_RESULT;
END_VAR

ClearEventQueueStatus := ClearEventQueue();
```

5.13. Простой протокол управления сетью

5.13.1. Введение

SNMP (*Простой протокол управления сетью*) это протокол, широко используемый сетевыми администраторами для предоставления важной информации и диагностического оборудования, присутствующего в данной сети Ethernet.

Этот протокол использует концепцию агента и менеджера, в которой менеджер отправляет агенту запросы на чтение или запись определенных объектов. Через MIB (Информационная база управления) менеджер знает о существующих объектах в агенте и, таким образом, может делать запросы к этим объектам, соблюдая разрешения на чтение или запись. MIB представляет собой набор информации, организованной иерархически, при этом каждый объект этого дерева называется OID (идентификатор объекта).

Для всего оборудования с SNMP обязательна поддержка MIB-II. В этой MIB описывается ключевая информация для управления сетями Ethernet.

5.13.2. SNMP NAS UCP Nexto

Центральные процессоры серии Nexto ведут себя как агенты при обмене данными по протоколу SNMP. Информация, доступная через SNMP, не может быть обработана или доступна через пользовательское приложение, для доступа к которой требуется внешний менеджер SNMP. В таблице ниже перечислены объекты, доступные в процессорах Nexto. Эта функция доступна после версии прошивки 1.4.0.33 процессоров серии Nexto, поддерживающей протоколы SNMPv1, SNMPv2c и SNMPv3, а также поддерживающей MIB-II, где необходимые объекты описаны в RFC-1213.

OID	Наименование	Описание
1.3.6.1.2.1.1	Система	Содержит название, описание, местонахождение и другую идентификационную информацию об оборудовании.
1.3.6.1.2.1.2	Интерфейсы	Содержит информацию о сетевых интерфейсах машины. Таблица ifTable (OID 1.3.6.1.2.1.2.2) имеет доступные индексы 6 и 7, которые можно просмотреть по статистике сетевых интерфейсов NET 1 и NET 2, соответственно, процессоров серии Nexto.
1.3.6.1.2.1.3	At	Содержит информацию о последних необходимых соединениях с агентом.
1.3.6.1.2.1.4	IP	Содержит статистические соединения с использованием IP-протокола.
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 подключения по протоколу SNMP.
1.3.6.1.2.1.31	ifMib	Расширение для интерфейсов, OID 1.3.6.1.2.1.2.

Таблица 200: Объекты MIB II — агент SNMP серии Nexto

По умолчанию агент SNMP активирован, т. е. служба инициализируется в момент запуска ЦП. Доступ к информации агента осуществляется через интерфейсы Ethernet NET 1 и NET 2 ЦП серии Nexto на TCP-порту 161. Таким образом, когда служба активна, информация об агенте может быть доступна через любой из двух

5. КОНФИГУРАЦИЯ

интерфейсов Ethernet, если они доступны. . Невозможно предоставить информацию агента через интерфейсы Ethernet модулей NX5000. На рисунке ниже показан пример SNMP-менеджера, в котором считываются некоторые значения.

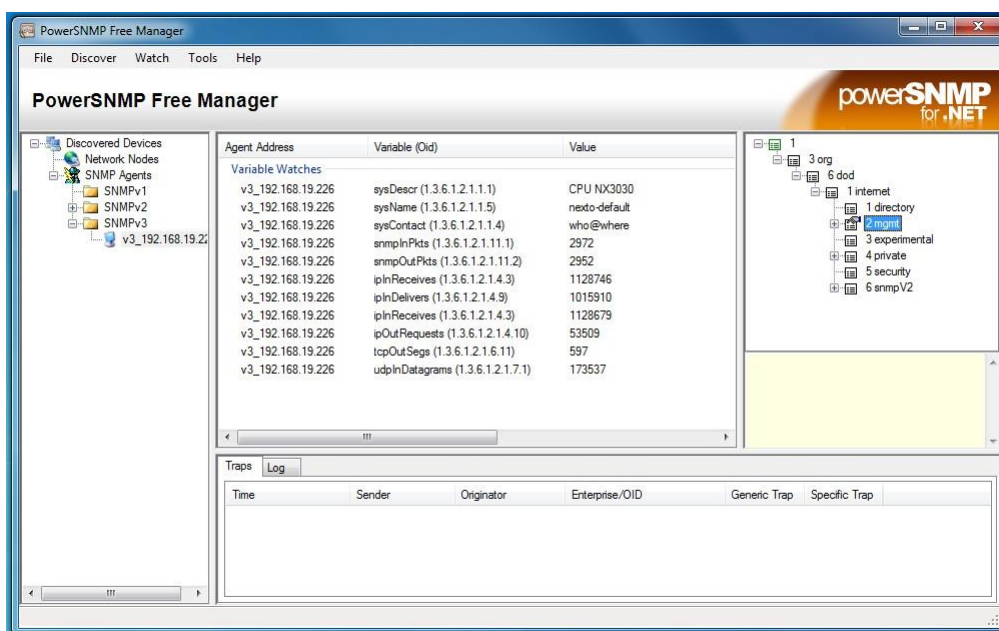


Рисунок 161: Пример SNMP-менеджера

Для SNMPv3, в котором предусмотрена аутентификация пользователя и пароль на запросы по протоколу SNMP, предусмотрен стандартный пользователь, описанный в разделе Пользователи и SNMP-сообщества.

Если вы хотите отключить службу, изменить пользователя SNMPv3 или сообщества для predeterminedенных SNMPv1 / v2c, вы должны получить доступ к веб-странице ЦП. Дополнительные сведения см. в разделе «Конфигурация SNMP».

5.13.3. Частный MIB

Частный MIB был прекращен с июня 2019 года.

5.13.4. SNMP-конфигурация

Настройки SNMP можно изменить через веб-страницу, на вкладке Управление ЦП в меню SNMP. Для доступа к настройкам необходимо сначала авторизоваться, как показано на рисунке ниже.

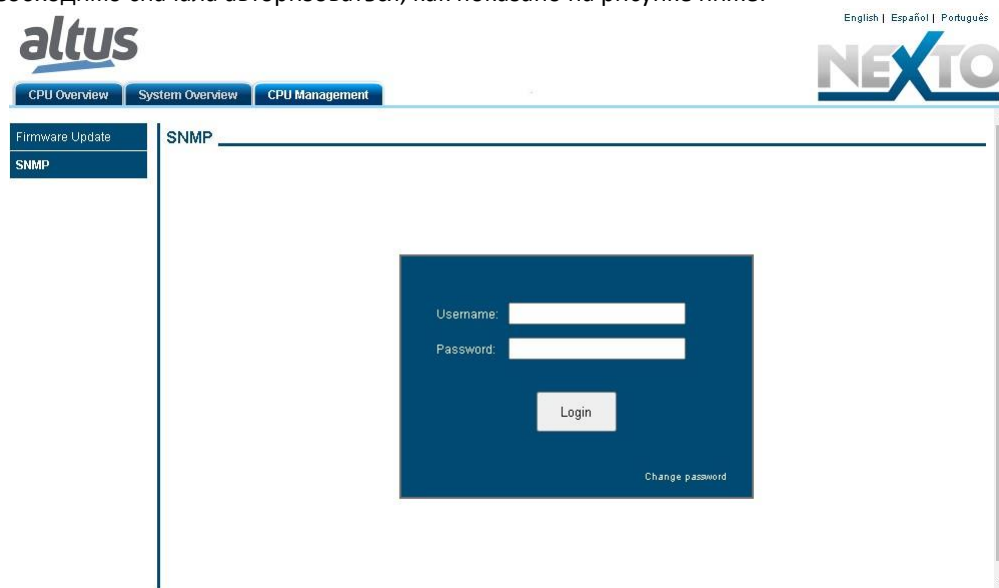


Рисунок 162: Экран входа в систему SNMP

ВНИМАНИЕ

Для входа на веб-страницу настроек SNMP в первый раз необходимо использовать имя пользователя и пароль по умолчанию.

После успешного входа в систему можно просмотреть текущее состояние службы (включено или отключено), а также информацию о пользователях SNMPv3 и сообществах для SNMPv1/v2c.

Пользователь может включить или отключить услугу с помощью флажка в верхней части экрана.

Также можно изменить информацию SNMPv3, нажав кнопку «Изменить» сразу под информацией о пользователе. Откроется форма, в которой необходимо ввести старый логин и пароль, а также новый логин и пароль. Другая информация о пользователе SNMPv3 не может быть изменена.

Чтобы изменить данные сообществ SNMPv1/v2c, процесс аналогичен, просто нажмите кнопку «Изменить» под информационным сообществом. Откроется новый экран, где будут вставлены новые данные для полей rcommunity и rwcommunity. Если вы не заполните какое-либо из полей, их сообщество будет отключено. Таким образом, если пользователь оставит два поля пустыми, доступ к агенту SNMP будет возможен только через SNMPv3.

Если пользователь хочет вернуться к настройкам по умолчанию, его необходимо вручную изменить в соответствии с разделом «Пользователи и SNMP-сообщества». Поэтому все текущие конфигурации SNMP будут сохранены в процессе обновления микропрограммы. Эти варианты показаны на рисунке ниже.

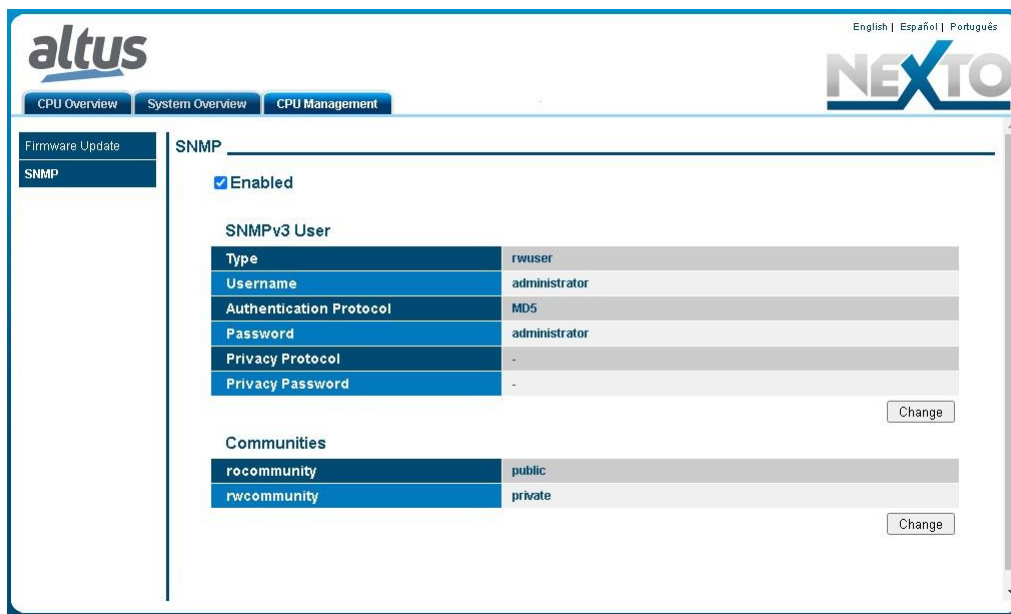


Рисунок 163: Экран конфигурации состояния SNMP

ВНИМАНИЕ

ИМЯ ПОЛЬЗОВАТЕЛЯ И ПАРОЛЬ для доступа к агенту по протоколу SNMP такие же, как и для входа на веб-страницу настроек SNMP.

5.13.5. Сообщества пользователей и SNMP

Для доступа к SNMPv1/v2c процессоров серии Nexto существует два сообщества, согласно таблице ниже.

Сообщества	Строка по умолчанию	Тип
rcommunity	государственный	Только чтение

rwcommunity	частный	Чтение и запись
-------------	---------	-----------------

Таблица 201: Информация о сообществах SNMPv1/v2c по умолчанию

Можно получить доступ к SNMPv3, используя пользователя по умолчанию, см. таблицу ниже:

Имя пользователя	Тип	Протокол аутентификации	Пароль	Протокол приватности	Пароль приватности
administrator	rwuser	MD5	administrator	-	-

Таблица 202: Информация о пользователе по умолчанию SNMPv3

Для всех настроек сообществ, пользователя и пароля должны соблюдаться некоторые ограничения, как описано в следующей таблице:

Настраиваемый элемент	Мин размер	Макс 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 203: Ограничения настроек SNMP

5.14. Управление пользователями и права доступа

Он предоставляет функции для определения учетных записей пользователей и настройки прав доступа к проекту и ЦП. С помощью программы MasterTool IEC XE можно создавать и управлять пользователями и группами, настройками, различными уровнями прав доступа к проекту.

В то же время процессоры Nexto имеют систему управления правами пользователей, которая блокирует или разрешает определенные действия для каждой группы пользователей в процессоре. Для получения дополнительной информации см. руководство пользователя MasterTool IEC XE MT85000 — MU299609 в разделе «Управление пользователями и права доступа».

7. Резервирование с ЦП NX3030

7.1. Введение

В этой главе описывается резервирование ЦП серии Nexto, которое можно использовать только с ЦП NX3030.

Резервирование Nexto — по типу горячего резерва, поэтому контроллеры дублируются. Один контроллер обычно находится в активном состоянии и управляет процессом, а другой обычно находится в режиме ожидания, сохраняя синхронность с активным контроллером. В случае сбоя в активном контроллере, наносящего ущерб его управлению процессом, резервный контроллер автоматически переключается в активный режим в течение очень короткого времени, чтобы не нарушать процесс и не вызывать перебоев в его выходах.

Горячее резервирование — это метод, используемый для повышения отказоустойчивости и, следовательно, повышения доступности систем автоматизации. Основная идея состоит в том, чтобы гарантировать, что простой отказ в дублированных компонентах не приведет к прерыванию управления процессом.

Горячее резервирование применяется к:

- Платформы для разведки нефти
- Установки по производству и распределению энергии
- Блокировка безопасности (Инструментальные системы безопасности)
- Непрерывные процессы, такие как химические заводы, нефтеперерабатывающие заводы, производство бумаги и т. д.

В процессорах серии Nexto с горячим резервированием, как уже было сказано, контроллеры сдвоены. Кроме того, можно дополнительно дублировать полевые шины (PROFIBUS-DP), а также сети управления Ethernet и сети управления Ethernet HSDN (высокоскоростной детерминированной сети). При выборе дублирования сетей доступность становится еще выше.

Горячее резервирование ЦП серии Nexto не применяется к модулям ввода-вывода. Если требуется резервирование модуля ввода/вывода, пользователь может настроить его на уровне приложения. Например, пользователь может продублировать или даже утроить модуль аналогового ввода и создать схему голосования, чтобы определить, какой ввод будет рассматриваться в определенное время приложения.

На рисунке ниже показан типичный пример резервированной архитектуры с использованием ЦП NX3030.

Центральная часть ЦП с резервированием состоит из двух одинаковых стоек, называемых PLCA и PLCB, и панели управления резервированием (PX2612). В контексте резервирования каждая стойка (PLCA или PLCB) называется полукластером, а группа, образованная этими двумя стойками, называется кластером.

В этом примере также можно наблюдать дублирование сети PROFIBUS, сети контроля Ethernet и сети управления Ethernet HSDN.

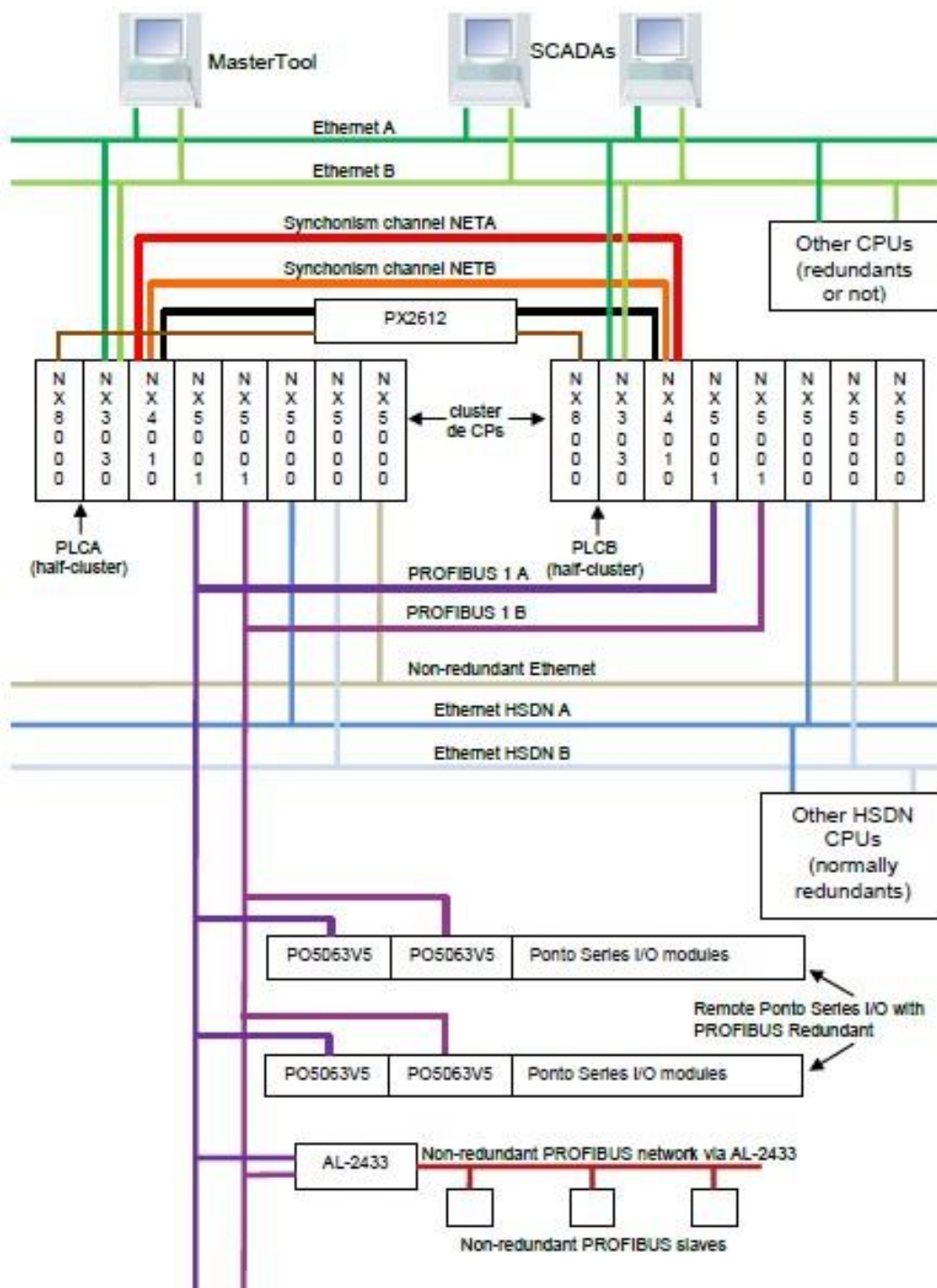


Рисунок 164: Пример резервированной архитектуры с ЦП NX3030

6.2. Техническое описание и конфигурация

6.2.1. Минимальная конфигурация резервного ЦП (без использования панели PX2612)

Резервный ЦП состоит, как минимум, из:

Два одинаковых полукластера

Каждый полукластер состоит как минимум из следующих модулей:

Сама стойка, куда вставляются модули, которая может быть одной из следующих:

- NX9000, 8 позиций

7. ОБСЛУЖИВАНИЕ

- NX9001, 12 позиций
- NX9002, 16 позиций
- NX9003, 24 позиции

Блок питания NX8000, позиции стойки 0 и 1

ЦП NX3030 в стойках 2 и 3

Модуль NX4010, на стойках 4 и 5

На рисунке ниже показан пример минимальной конфигурации резервного ЦП с использованием наименьшей стойки (NX9001, с 12 позициями). В этом случае можно заметить, что три модуля, вставленные в стойку, имеют двойную ширину (занимают две позиции в стойке).

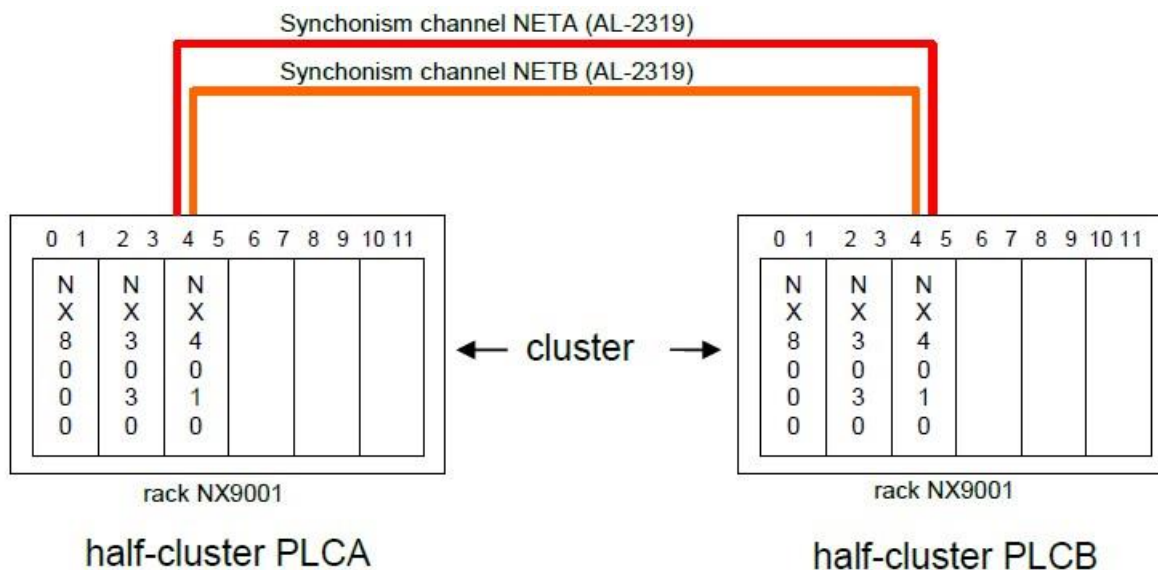


Рисунок 165: Минимальная конфигурация резервного ЦП в стойке NX9001

6.2.2. Типичные конфигурации резервного ЦП

Минимальная конфигурация, показанная на рис. 165, уже может работать как «избыточный процессор». Можно было бы использовать последовательные каналы связи и каналы Ethernet от ЦП NX3030, например, для связи MODBUS TCP с системой SCADA и связи MODBUS RTU и/или MODBUS TCP с интеллектуальными полевыми устройствами или системами удаленного ввода-вывода MODBUS.

Однако в типичных конфигурациях в полукластеры PLCA и PLCB вставляются дополнительные модули, например, для обеспечения удаленного ввода-вывода PROFIBUS и дополнительных каналов Ethernet.

Среди дополнительных модулей, которые опционально могут быть вставлены в полукластеры, можно выделить следующие:

- Мастер PROFIBUS NX5001
- Ethernet-интерфейсы NX5000

При необходимости можно использовать стойки большего размера, например, NX9002 (16 позиций) и NX9003 (24 позиции). Следует отметить, что все перечисленные модули в этой главе имеют двойную ширину (занимают две позиции). Кроме того, также можно использовать панель PX2612, которая позволяет выполнять некоторые избыточные переходы конечного автомата, которые в противном случае были бы невозможны, в дополнение к автоматическому отключению полукластеров в аварийных ситуациях.

7. ОБСЛУЖИВАНИЕ

6.2.2.1. Добавление модулей NX5001 для сетей PROFIBUS

Резервный ПЛК поддерживает до четырех модулей NX5001 для использования сетей PROFIBUS. Каждая сеть может быть одиночной или резервной. Если PROFIBUS «n» (т. е. «n» означает число от 1 до 4) является резервным, то две принадлежащие ему сети называются PROFIBUS «n» А и PROFIBUS «n» В. В случае, если PROFIBUS «n» одинарным, принадлежащая ему сеть будет называться PROFIBUS «n» А.

Для создания резервной сети PROFIBUS необходимо установить по два модуля NX5001 в каждый полукластер. Чтобы создать простую сеть PROFIBUS, просто вставьте модуль NX5001 в каждый полукластер. Таким образом, можно настроить до четырех простых сетей, или две резервные сети, или одну резервную и две простые. В других случаях в каждом полукластере потребуется менее четырех модулей NX5001. Более подробная информация о сетях PROFIBUS представлена в разделе Конфигурация сети PROFIBUS.

На рисунке 164 показана только одна сеть PROFIBUS (PROFIBUS 1), и столько же резервных (PROFIBUS 1 А и PROFIBUS 1 В). Таким образом, в этом примере в каждый полукластер было вставлено два модуля NX5001.

6.2.2.2. Добавление модулей NX5000 для сетей Ethernet

В каждый полукластер можно вставить до шести модулей NX5000, предоставляя шесть дополнительных каналов Ethernet помимо двух каналов Ethernet, уже существующих в ЦП NX3030.

Каналы Ethernet можно использовать по отдельности или организовать в пары объединения сетевых карт, которые используются для предоставления резервных каналов Ethernet и более подробно описаны в разделе «Резервные сети Ethernet с объединением сетевых карт».

Типичным применением модуля NX5000 может быть построение резервированной HSDN (высокоскоростной детерминированной сети) для связи между несколькими резервными ЦП. Через эту сеть многие резервированные ЦП могут обмениваться сообщениями в полностью изолированной сети, чтобы гарантировать детерминизм и быструю связь. Кроме того, настроив эту сеть как избыточную с парами NIC Teaming, можно добиться отличной доступности. Для построения такой сети (с резервированием HSDN) в каждый полукластер необходимо установить два модуля NX5000. На рис. 164 показан пример HSDN с резервированием, использующий два модуля NX5000 в каждом полукластере.

Приложения, в которых модули ввода и вывода подключены к сетям Ethernet, могут потребовать дополнительных интерфейсных модулей NX5000 для подключения к этим сетям. В этих случаях сеть, соединяющая модули входов и выходов, может быть простой или резервированной сетью. Кроме того, интерфейсы могут быть настроены с возможностью генерирования отказов в работе. В этом случае сбой сети вызовет переключение.

На рис. 164 также показан пример с модулем NX5000, используемым в изолированном виде (без резервирования объединения сетевых карт), вставленным справа от других модулей в каждую стойку.

6.2.3. Модуль NX4010

Модель NX4010, как показано на рисунке ниже, была задумана для обеспечения взаимосвязи между двумя полукластерами PLCA и PLCB, а также для подключения этих полукластеров к панели управления резервированием PX2612. Дополнительные сведения о соединениях этого модуля см. в разделе Взаимосвязи между полукластерами и панелью управления резервированием PX2612.



Рисунок 166: NX4010

6.2.3.1. Особенности NX4010

Его основные особенности:

- Синхронизация данных и приложений между двумя полукластерами
- Резервный интерфейс связи между двумя полукластерами
- Автоматическое переключение (смена активного полукластера) в случае тайм-аута связи NX4010 и ЦП. Возможность выключения другого полукластера.
- Диагностика одним касанием
- Электронная метка на дисплее
- Дисплей и светодиоды для индикации диагностики
- Другие характеристики (общие, электрические, механические и экологические) представлены в технических характеристиках модуля резервирования NX4010 — CE114900.

6.2.4. Панель управления резервированием PX2612

Панель управления PX2612 является дополнительным элементом в системе с резервированием. Его необходимо использовать, когда во время создания проекта с помощью мастера выбрана опция «Резервирование с панелью». На рис. 167 показана панель управления резервированием, а на рис. 168 — передняя панель с деталями.

Через разъем DB9, называемый CONTROL PLC A, осуществляется соединение с разъемом CONTROL от PLCA NX4010 с помощью кабеля AL-2317/A.

Через разъем DB9 под названием CONTROL PLC B осуществляется соединение с разъемом CONTROL от PLCB NX4010 с помощью кабеля AL-2317/B.

7. ОБСЛУЖИВАНИЕ

Кроме того, есть разъем с 5 штыревыми клеммами:

GND: клемма для заземления.

RL A: 2 клеммы, подключенные к контактам NO (нормально разомкнутые) реле, которым можно дать команду PLCB на отключение PLCA. Это реле должно быть замкнуто PLCB, чтобы отключить PLCA.

RL B: 2 клеммы, подключенные к контактам NO (нормально разомкнутые) реле, которым PLCA может дать команду на отключение PLCB. Это реле должно быть замкнуто PLCA, чтобы отключить PLCB.

ЦП (PLCA или PLCB) может отключить другой ЦП (PLCB или PLCA) в некоторых исключительных ситуациях, используя замыкающие реле в разъемах RLA и RLB. Такие ситуации описаны в разделе «Переход между состояниями резервирования».

PX2612 также имеет 6 кнопок для команды резервирования и 6 светодиодов, используемых для индикации состояния резервирования. Каждый ЦП считывает 3 из этих 6 кнопок и управляет 3 светодиодами.

Дополнительные сведения об этих кнопках и функциях светодиодов см. в разделе «Функции панели управления резервированием PX2612».



Рисунок 167: Панель управления резервированием PX2612

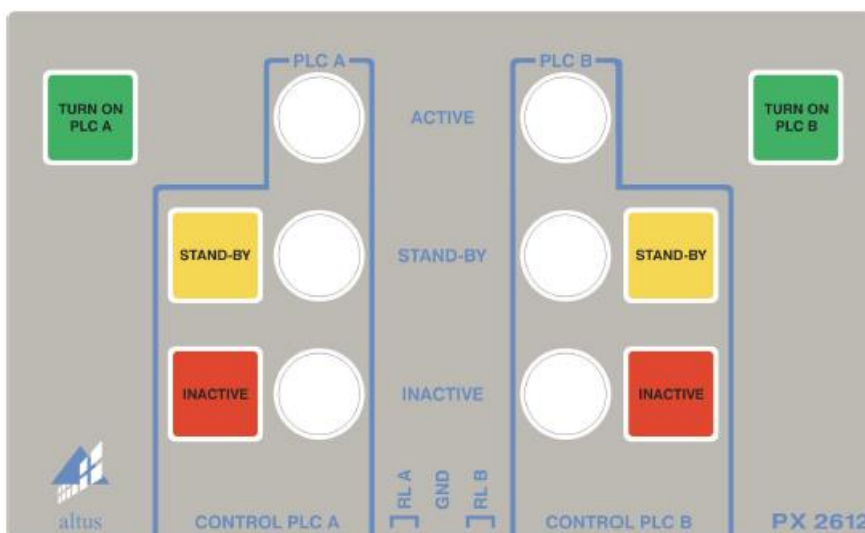


Рисунок 168: Панель управления резервированием PX2612, вид спереди

6.2.4.1. Характеристики PX2612

Панель управления резервированием PX2612 имеет следующие особенности:

7. ОБСЛУЖИВАНИЕ

CONTROL PLC A: подключение к модулю NX4010 от PLCA

CONTROL PLC B: подключение к модулю NX4010 от PLCB

RL A: релейные клеммы NO, используемые для отключения PLCA

RL B: клеммы NO реле, используемые для отключения PLCB GND: заземление

Другие функции (общие, электрические, механические и экологические) представлены в технических характеристиках панели управления резервированием PX2612 — СТ112500.

6.2.5. Взаимосвязи между полукластерами и панелью управления резервированием PX2612

На приведенном ниже рисунке показано, как должны быть выполнены соединения между PLCA, PLCB и PX2612, включая возможность разрешить одному ЦП выключать другой, что необходимо в исключительных ситуациях.

Для каналов синхронизации и резервирования NETA и NETB необходимо использовать два кабеля AL-2319. Один из этих двух кабелей соединяет разъем NX4010 NET 1 от каждого ЦП (канал NETA). Другой кабель соединяет разъем NET 2 NX4010 от каждого ЦП (канал NETB).

Кабель AL-2317/A соединяет разъем NX4010 CONTROL между PLCA и PX2612 CONTROL PLC A.

Кабель AL-2317/B соединяет разъем NX4010 CONTROL от PLCB с PX2612 CONTROL PLC B.

Кроме того, необходимо построить специальную схему питания, чтобы процессор мог в крайних случаях выключать другой.

Для большей надежности необходимо использовать два отдельных источника питания 24 В, один для питания PLCA, а другой — для питания PLCB.

Можно заметить, что для питания NX8000 необходимо использовать два внешних реле нормально замкнутого типа (NC) с допустимым током. Эти реле должны быть рассчитаны на номинальный ток около 2 А, однако необходимо учитывать бросок тока около 10 А. Шунтирующие диоды, подключенные к соленоидам реле NC, должны использоваться для защиты контактов реле PX2612 NO.

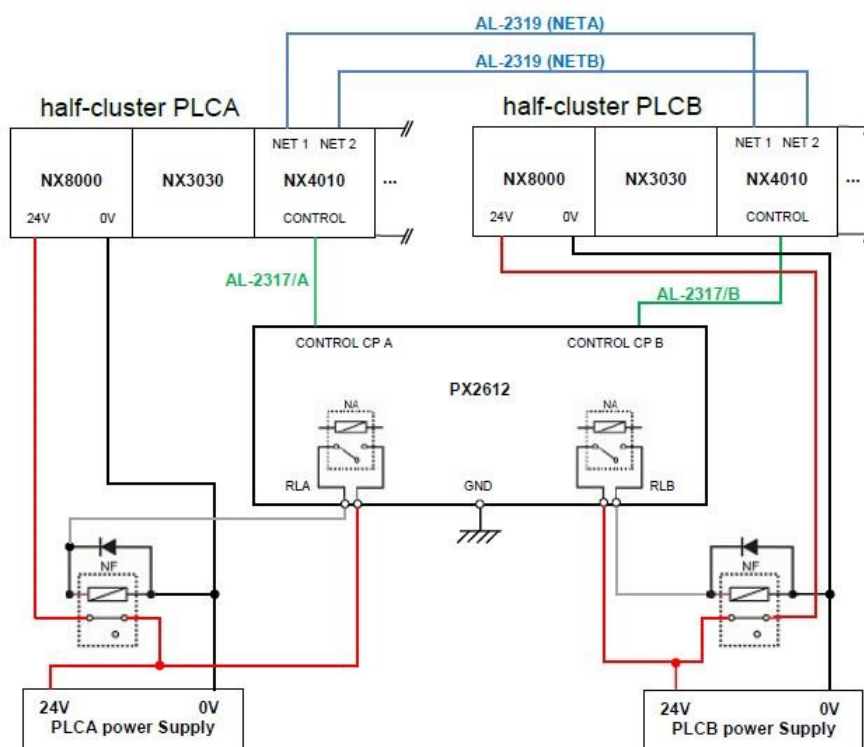


Рисунок 169: Взаимосвязи между PLCA, PLCB и PX2612.

6.2.6. Общие характеристики резервного CP

	Общие характеристики резервного ЦП
Разрешенные процессоры	NX3030
Тип резервирования	Горячее резервирование
Отказоустойчивые допуски	Терпит, по крайней мере, простые отказы в вдвоенном оборудовании в полукластерах. В определенных случаях он может допускать множественные сбои
Полукластер 5 состояний резервирования	<ul style="list-style-type: none"> - - Не настроено: начальное состояние, также считается, когда ЦП выключен или не выполняет основную задачу. - - Запуск: временное состояние, предполагаемое после Не настроено, где некоторые тесты определяют следующее состояние (Неактивно, Активно, Ожидание или обратно в Не настроено. - - Неактивно: состояние, достигнутое после некоторых типов сбоев или для обслуживания программирования. - - Активный: управление пользовательским процессом. - - В режиме ожидания: готов переключиться в активный режим и управлять пользовательским процессом в случае возникновения такой потребности (например, сбой активного процессора).
Основные отказы, вызывающие переключение между активным ЦП и резервным ЦП. Резервные переключатели CPU для Active и Active могут перейти в состояние Inactive или Notconfigured..	<ul style="list-style-type: none"> - Сбой подачи. - Источник питания. - CPU (остановка выполнения MainTask). - NX4010. - Сбой в обоих каналах синхронизации (NETA и NETB) и причина не в резервном ЦП. В этом случае резервный ЦП, помимо перехода в активное состояние, выключает другой ЦП. - Отказ какого-либо канала синхронизации (NETA и NETB) и причина в активном ЦП. - Сбой в какой-либо жизненно важной сети PROFIBUS. - Сбой в какой-то жизненно важной сети Ethernet.
Команды, вызывающие переключение между активным ЦП и резервным ЦП	<ul style="list-style-type: none"> - - Команды через панель управления резервированием (PX2612). - - Команды, полученные от MasterTool или из системы SCADA, через этот ЦП (локальный) или другой ЦП (удаленный). - - Команды, генерируемые пользовательским приложением (например, в случае другой диагностики, такой как сбой связи Ethernet) через этот ЦП (локальный) или другой ЦП (удаленный).

7. ОБСЛУЖИВАНИЕ

<p>Основные сбои, которые не позволяют процессору перейти в резервное состояние или остаться в нем. Такие сбои переводят ЦП в состояние Not-Configured или Inactive.</p>	<ul style="list-style-type: none"> - Сбой подачи. - Источник питания. - CPU (остановка выполнения MainTask). - NX4010. - Сбой в обоих каналах синхронизации (NETA и NETB) и причина в резервном ЦП. - Сбой в службе синхронизации для избыточных данных. - Сбой в службе синхронизации для избыточного принудительного списка. - Полный сбой в какой-то жизненно важной сети PROFIBUS. - Полный сбой в какой-то жизненно важной сети Ethernet. - Отдельный проект от Active CPU, с автоматической синхронизацией проекта Включено. - Версия микропрограммы несовместима с активным ЦП.
--	---

Общие характеристики резервного ЦП	
<p>Команды, выводящие ЦП из резервного состояния</p>	<ul style="list-style-type: none"> - Команды через панель управления резервированием (PX2612). - Команды, полученные от MasterTool или из системы SCADA, через этот ЦП (локальный) или другой ЦП (удаленный). - Команды, генерируемые пользовательским приложением (например, в случае другой диагностики, такой как сбой связи Ethernet) через этот ЦП (локальный) или другой ЦП (удаленный).
<p>Время переключения</p>	<ul style="list-style-type: none"> - До 3 циклов от MainTask, в зависимости от стимула для изменения состояния (команда или отказ). - В случае отказа сети PROFIBUS 2 цикла MainTask + 500 мс.
<p>Переключение без разрывов (без скачков)</p>	<ul style="list-style-type: none"> - Переключение не вызывает разрывов ни на выходах контроллера, ни во внутренних переменных
<p>Накладные расходы на избыточность (загрузка ЦП цикла MainTask увеличивается за счет избыточности)</p>	<ul style="list-style-type: none"> - Максимальное значение, автоматически рассчитанное MasterTool и сообщенное пользователю, с учетом пустого списка избыточных форсировок. - Типичное среднее значение 60 мс для 224 кбайт избыточных данных в системе с резервированной сетью PROFIBUS и двумя резервными сетями Ethernet HSDN.
<p>Дисплей процессора</p>	<p>Среди других диагностических данных показывает состояние резервирования (активен, резервный, неактивен, не настроен и запущен) вместе с идентификацией ЦП PLCA или PLCB.</p>

7. ОБСЛУЖИВАНИЕ

Панель управления резервированием PX2612	<ul style="list-style-type: none"> - С помощью кнопок позволяет выполнять команды переключения или перехода в режим резервирования для технического обслуживания. - Светодиоды сигнализируют о состоянии резервирования в каждом полукластере. - Реле НО позволяет полукластеру отключать другой в экстремальных ситуациях. Кнопка позволяет повторно активировать другой полукластер. - Встроенные ресурсы для проверки кнопок, светодиодов и реле. <p>Проект, созданный с панелью, нельзя преобразовать в проект без панели, и наоборот.</p>
Диагностика резервирования	<ul style="list-style-type: none"> - Индикация отказов в PLCA и в PLCB, независимо от их состояний (Активный или Неактивный). - Предотвращать «непонятные сбои». <p>Разрешить быстрое обслуживание, необходимое для высокой доступности.</p>
Команды резервирования	<ul style="list-style-type: none"> - Разрешить выполнение тех же действий панели управления PX2612, среди прочих команд (например, команду переключения). - Может выполняться в локальном ЦП или передаваться на другой ЦП (удаленный) по каналам синхронизации NETA/NETB. - Можно получить через MasterTool или систему SCADA. - Может выполняться через пользовательское приложение.
События резервирования	<p>Зарегистрируйте изменения команд диагностики и резервирования с отметкой времени, что позволит исследовать причины переключения.</p>
SNTP (простой протокол сетевого времени)	<p>Разрешить событиям иметь точную временную метку, настроенную на мировой час. Он также синхронизирует часы реального времени процессора с другими приложениями.</p>
Синхронизация команд и диагностики	<p>В каждом цикле MainTask PLCA и PLCB обмениваются диагностическими данными и командами через каналы синхронизации NETA или NETB. Таким образом, ЦП знает другие диагностические данные и команды.</p>

	Общие характеристики резервного ЦП
Резервная синхронизация данных	<p>В каждом цикле MainTask активный ЦП копирует избыточные данные в неактивный ЦП через каналы синхронизации NETA и NETB. Неизбыточные данные не синхронизируются.</p>
Избыточная принудительная синхронизация списка	<p>В каждом цикле MainTask активный ЦП копирует избыточный список форсирования в неактивный ЦП через каналы синхронизации NETA и NETB. Этот список включает только принудительно избыточные переменные, таким образом, PLCA и PLCB могут иметь разные избыточные группы данных, так как эти переменные не синхронизированы.</p>

7. ОБСЛУЖИВАНИЕ

Единый проект для PLCA и PLCB	Для PLCA и PLCB существует единый проект, созданный MasterTool. Проект состоит из проекта приложения (исполняемый код) и проекта архива (исходный код).
Идентификация процессора	Через MasterTool ЦП NX3030 идентифицирует себя как PLCA, PLCB или ЦП без резервирования. Эта идентификация не является частью проекта приложения, созданного MasterTool, несмотря на то, что она написана в ЦП с использованием MasterTool. Идентификация ЦП позволяет реализовать функцию единого проекта для PLCA и PLCB.
Автоматическая синхронизация проектов	Если проект Active CPU отличается от проекта Inactive CPU, он копируется из первого во второй. Эта синхронизация может занять несколько циклов MainTask. Следует помнить, что проект состоит из проекта приложения (исполняемый код) и проекта архива (исходный код), и оба они синхронизированы. Эта синхронизация может быть выключена в особых случаях, чтобы разрешить визуализацию изменений проекта, которые могут быть загружены только в автономном режиме в нерезервированные ЦП.
Онлайн-расширение модулей и пультов PROFIBUS	Существуют модификации проекта, которые невозможно выполнить в режиме онлайн в нерезервированном ЦП, например, включение новых модулей или удаленных устройств PROFIBUS. Однако, используя резервирование ЦП и сети PROFIBUS, была определена процедура для достижения этой цели, что очень важно для систем, которым требуется высокая доступность.
Частные IP-адреса	Можно подключиться к определенному ЦП (PLCA или PLCB) с использованием частного IP-адреса, например, для получения диагностики, специфичной для полукластера. IP-адрес PLCA всегда будет связан с интерфейсом PLCA NET(i), а IP-адрес PLCB всегда будет связан с интерфейсом PLCB NET(i).
ПЛКА и ПЛКБ	Имя стратегии, позволяющей Ethernet-клиенту подключаться к серверу с резервного ЦП, используя всегда один и тот же IP-адрес. Это предотвращает необходимость сложных сценариев для изменения IP-адреса при переключении из-за избыточности. Активный IP-адрес всегда будет связан с интерфейсом NET(i) активного ЦП.
Активный IP	Название стратегии, которая позволяет двум интерфейсам Ethernet из полукластера образовывать избыточную пару, использующую один и тот же IP-адрес. Таким образом, можно легко построить резервную сеть Ethernet без необходимости для клиентов, подключенных к NIC Teaming, реализовывать сложные сценарии для переключения IP-адресов.
	Общие характеристики резервного ЦП
Конфигурация сети PROFIBUS и существенных отказов	CPU поддерживает до 4 простых сетей PROFIBUS или до 2 резервных сетей PROFIBUS. Также можно настроить, считается ли каждый сбой сети PROFIBUS жизненно важным (вызывает переключение) или нет.

7. ОБСЛУЖИВАНИЕ

Конфигурация сети Ethernet и основных отказов	ЦП поддерживает до 8 простых сетей Ethernet или до 4 сетей Ethernet с резервированием (с учетом передних интерфейсов NX3030). Также можно настроить, считается ли каждый сбой сети Ethernet жизненно важным (вызывает переключение) или нет.
Одиночная и циклическая пользовательская задача	Разрешена только одна пользовательская задача, называемая MainTask. Эта задача циклична.
Основные программы POU: Non-SkippedPrg и ActivePrg	При создании избыточного проекта MasterTool автоматически генерирует две пустые программы POU, которые должны быть заполнены пользователем: - NonSkippedPrg: этот POU выполняется в обоих ЦП (PLCA и PLCB), независимо от состояния резервирования (активен или неактивен). Используется для диагностики и управления специальными командами. - ActivePrg: этот POU выполняется только в активном ЦП и используется для управления процессами конечного пользователя.

Таблица 204: Общие характеристики резервного ЦП

6.2.7. Данные о покупке

Минимальная конфигурация для резервного ЦП подразумевает покупку следующих модулей:

Две стойки, которые необходимо выбрать из четырех доступных моделей в соответствии с устанавливаемыми модулями:

- NX9000: 8 позиций (4 двойных модуля)
- NX9001: 12 позиций (6 двойных модулей)
- NX9002: 16 позиций (8 двойных модулей)
- NX9003: 24 позиции (12 двойных модулей)
 - Два NX8000
 - Два NX3030
 - Два NX4010
 - Два АЛ-2319

Кроме того, может потребоваться приобретение следующих дополнительных модулей:

- Один PX2612
- Один АЛ-2317/А
- Один АЛ-2317/Б
- Два модуля NX5001 для каждой простой сети PROFIBUS
- Четыре модуля NX5001 для каждой резервированной сети PROFIBUS
- Два модуля NX5000 для каждой дополнительной простой сети Ethernet

Четыре модуля NX5000 для каждой дополнительной сети Ethernet с резервированием (объединение сетевых карт)

ВНИМАНИЕ

В каждом полукластере можно установить до 4 модулей PROFIBUS. Это означает, что мы можем настроить до 4 простых сетей PROFIBUS или до 2 резервных сетей PROFIBUS.

6.3. Принципы работы

В этом разделе описаны функции резервного ЦП, использующие ЦП NX3030, а также его поведение и состояния. Также представлены концепции и ограничения по программированию и настройке, которые будут использоваться в следующих разделах.

6.3.1. Идентификация ЦП NX3030

ЦП NX3030 имеет энергонезависимый идентификационный регистр, в котором он может быть идентифицирован как:

Без резервирования: нельзя использовать в резервированном ЦП (конфигурация по умолчанию)

PLCA: используется в резервном ЦП PLCA

PLCB: используется в PLCB с резервированием ЦП.

Регистр идентификации можно настроить с помощью программатора MasterTool. Первое, что необходимо сделать в ЦП NX3030 перед загрузкой в него резервного проекта, — это идентифицировать его как PLCA или PLCB. В случае, если идентификация не выполнена, некоторые функции резервирования не будут работать корректно, например, синхронизация приложения между ПЛК.

ВНИМАНИЕ

Регистр идентификации ЦП не является частью проекта резервного ЦП, поэтому он не сохраняется как часть этого проекта на компьютере, на котором выполняется MasterTool. Регистр сохраняется только в энергонезависимой памяти ЦП.

6.3.2. Один резервный проект

Из-за ранее описанного идентификационного регистра для резервного ЦП существует единый проект, идентичный как для PLCA, так и для PLCB.

Параметры конфигурации, которые должны быть разными для PLCA и PLCB (например, IP-адрес интерфейса Ethernet), отображаются в проекте резервного ЦП дважды (один для PLCA, а другой для PLCB). Каждый ЦП будет рассматривать соответствующий, после анализа его идентификационного регистра.

6.3.3. Избыточная структура проекта

6.3.3.1. Шаблон резервирования

Проект резервного ЦП создается автоматически из модели, называемой шаблоном резервирования.

Шаблон начинается с минимальной конфигурации резервного ЦП, как определено в разделе «Минимальная конфигурация резервного ЦП (без использования панели PX2612)». Кроме того, некоторые диалоги с пользователем выполняются для вставки дополнительных модулей в полукластеры, таких как мастера PROFIBUS (NX5001) и модули Ethernet (NX5000).

Удаленные устройства PROFIBUS должны быть вставлены пользователем ниже уже вставленных ведущих устройств NX5001 PROFIBUS.

Кроме того, создаются задачи и базовые POU из типа программы, как описано в следующих разделах.

6.3.3.2. Одиночная и циклическая задача

В проекте резервного ЦП есть одна задача, называемая MainTask, которая является циклической. Пользователь может настроить время цикла задачи.

6.3.3.3. Программа MainPrg

MainTask подключается к одному POU из типа программы, который называется MainPrg. Программа MainPrg создается автоматически.

Код MainPrg следующий, на языке ST:

```
SpecialVariablesPrg();
IF isFirstCycle THEN StartPrg();
    isFirstCycle := FALSE;
ELSE fbRedundancyManagement();
    IF fbRedundancyManagement.m_fbDiagnosticsLocal.eRedState = REDUNDANCY_STATE.
        ACTIVE THEN
            SpecialVariablesRedundantPrg();
        END_IF;
    NonSkippedPrg();

    IF fbRedundancyManagement.m_fbDiagnosticsLocal.eRedState = REDUNDANCY_STATE.
        ACTIVE THEN
            ActivePrg();
        END_IF;
    END_IF;
```

MainPrg вызывает два POU из типа программы, которые называются NonSkippedPrg и ActivePrg. NonSkippedPrg вызывается всегда, так как выполняется в обоих процессорах. С другой стороны, ActivePrg вызывается только тогда, когда условие «RedDgnLoc.sGeneral.Diag.eRedState = Active» истинно, другими словами, когда ЦП находится в активном состоянии.

Однако программа NonSkippedPrg выполняется в обоих CPU (PLCA и PLCB) независимо от состояния резервирования этого CPU. С другой стороны, ActivePrg выполняется только в активном CPU.

В отличие от MainPrg, которую нельзя изменять, пользователь может изменять программы NonSkippedPrg и ActivePrg. Изначально при создании резервного проекта из шаблона резервирования эти две программы создаются «пустыми», но после этого пользователь может вставить свой код.

ВНИМАНИЕ

Когда при создании проекта включена опция OPC, программа NonSkippedPrg не создается пустой. Дополнительные сведения см. в разделе Использование связи OPC DA с резервными проектами.

6.3.3.4. Программа ActivePrg

Основная цель этой программы, которая выполняется только в активном ЦП, — управление конечным пользовательским процессом.

Эта программа обычно работает с избыточными переменными, среди которых находятся переменные прямого представления %I и %Q, связанные с системой удаленного ввода/вывода. Для получения дополнительной информации см. раздел Программирование резервного ЦП - Конфигурация основной задачи - Программа ActivePrg.

ВНИМАНИЕ

Успешна компиляция или нет, MasterTool информирует о расчетной неплотности и прогнозируемых издержках избыточности в окне сообщения.

6.3.3.5. Программа NonSkippedPrg

Эта программа выполняется в обоих CPU (PLCA и PLCB) независимо от состояния резервирования. Обычно он используется для таких функций, как:

Организовать нерезервированную диагностику с отчетом в SCADA-систему.

Для получения и выполнения нерезервированных команд из SCADA-системы.

7. ОБСЛУЖИВАНИЕ

Для управления условиями переключения, которые обычно не учитываются резервным ЦП автоматически, они могут варьироваться от пользователя к пользователю. Например, пользователь сможет выполнить переключение на резервный ЦП, если активный ЦП не обменивается данными с системой SCADA, в то время как другой пользователь может не захотеть переключения в этой ситуации.

Чтобы включить или отключить драйверы ввода-вывода в соответствии с состоянием резервирования, например, отключить ведущий Modbus RS-485 в неактивном ЦП.

Для обнаружения сбоев в драйверах ввода-вывода в неактивном ЦП во избежание скрытых сбоев. Некоторые драйверы ввода-вывода не включают автоматическое обнаружение таких сбоев, в то время как другие, такие как PROFIBUS, делают это автоматически.

Другие действия, которые по какой-либо причине должны выполняться либо в Активном ЦП, либо в Резервном ЦП.

Для получения дополнительной информации см. раздел Программирование резервного ЦП — Конфигурация основной задачи — Программа NonSkippedPrg.

6.3.3.6. Избыточные и неизбыточные переменные

Избыточные переменные ЦП можно разделить на избыточные и неизбыточные. Резервные переменные копируются из активного ЦП в неактивный ЦП в начале каждого цикла MainTask через каналы синхронизации NETA и NETB. С другой стороны, неизбыточные переменные не копируются между полукластерами, поэтому могут иметь разные значения в PLCA и PLCB.

Неизбыточные переменные используются для хранения личной информации каждого полукластера (PLCA или PLCB), такой как диагностика модуля внутри полукластера, включая диагностику резервирования (состояние диагностики полукластера и т. д.).

Избыточные переменные относятся к общей информации, связанной с управлением процессом. Переменные, связанные с модулями ввода-вывода, являются типичными примерами избыточных переменных.

6.3.3.7. Резервные и неизбыточные переменные %I

ЦП NX3030 выделяет 96 кбайт переменных %I (%IB0 ... %IB98303).

Первые 80 кбайт могут быть избыточными (%IB0 ... %IB81919). Последние 16 кбайт всегда не избыточны (%IB81920 ... %IB98303).

Область 80 кбайт, которая может быть избыточной, предназначена для входов, которые могут считываться с удаленного модуля ввода/вывода (PROFIBUS, MODBUS и т. д.).

Нерезервируемая область 16 кбайт выделена для полукластера «быстрой приватной диагностики», а также для кнопок командной панели резервирования PX2612. Быстрая диагностика должна обновляться в каждом цикле MainTask.

Пользователь может настроить количество избыточных переменных %I от 0 до 81920 кбайт, кратное 1 кбайту (значение по умолчанию 16384 байта - %IB0 ... %IB16383). Правильная конфигурация избыточного %I из %IB0 важна для сокращения времени, необходимого для синхронизации избыточных переменных (уменьшения накладных расходов на избыточность). Например, если реальное приложение выделяет только %IB0 ... %IB1499 для избыточных входов, размер избыточной области %I может быть определен как 1500 байт.

На рисунке ниже показано распределение избыточных и неизбыточных переменных прямого представления %I, где RI — это количество %I, действительно сконфигурированное как избыточное.

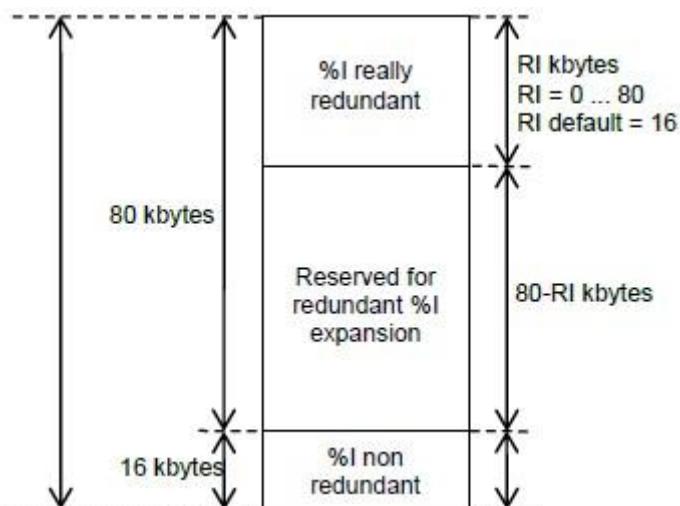


Рисунок 170: Резервированное и избыточное распределение %I

6.3.3.8. Резервные и избыточные переменные %Q

ЦП NX3030 выделяет 96 кбайт переменных %Q (%QB0 ... %QB98303). Первые 80 кбайт могут быть избыточными (%QB0 ... %QB81919). Последние 16 кбайт всегда избыточны (%QB81920 ... %QB98303).

Область размером 80 кбайт, которая может быть избыточной, разделена на две части:

Первые килобайты зарезервированы для выходов, которые могут быть избыточными, и обычно выделяются для удаленной системы ввода-вывода (PROFIBUS, MODBUS и т. д.). Значение размера настраивается, и его значение по умолчанию равно 16384. По умолчанию этот раздел включает %QB0 ... %QB16383 и может быть настроен до 64 кбайт (%QB0 ... %QB65535).

Следующие байты зарезервированы для диагностики, которая может быть избыточной, например, от системы ввода/вывода (диагностика модулей ввода/вывода, диагностика коммуникационных интерфейсов, диагностика ведомых устройств PROFIBUS и т. д.). В отличие от быстрой диагностики (размещенной в %I), для обновления такой диагностики, размещенной в %Q, может потребоваться более одного цикла MainTask. По умолчанию этот раздел включает 16 кбайт (%QB65536 ... %QB81919).

Нерезервированная область (%QB81920 ... %QB98303) обычно отводится для диагностики и частных команд полукластера, а также для светодиодов и реле панели управления резервированием PX2612.

Пользователь может уменьшить количество избыточных переменных %Q в каждой из секций, которые могут быть избыточными:

В первом разделе действительно избыточный размер области можно настроить в диапазоне от 0 байт до 65535 байт, кратным 1 байту (значение по умолчанию — 16384 байта). Правильная конфигурация избыточного %Q важна для сокращения времени, необходимого для синхронизации избыточных переменных (уменьшения накладных расходов на избыточность). Например, если реальное приложение выделяет только %Q0 ... %Q1499 для резервных выходов, размер резервной области %Q может быть определен как 1500 байт.

Во втором разделе реально избыточный размер области можно настроить в диапазоне от 0 байт до 81919 байт, кратным 1 байту (значение по умолчанию — 16384 байта). Правильная конфигурация избыточного %Q важна для сокращения времени, необходимого для синхронизации избыточных переменных (уменьшения накладных расходов на избыточность). Например, если реальное приложение выделяет только %QB65536 ... %QB66999 для избыточной диагностики, размер избыточной области %Q можно определить как 1464 байта.

На рисунке ниже показано распределение избыточных и избыточных переменных прямого представления %Q, где RQS — это выходная величина %Q, сконфигурированная как избыточная в первом разделе, а RQD — это диагностическая величина %Q, сконфигурированная как избыточная во второй секции.

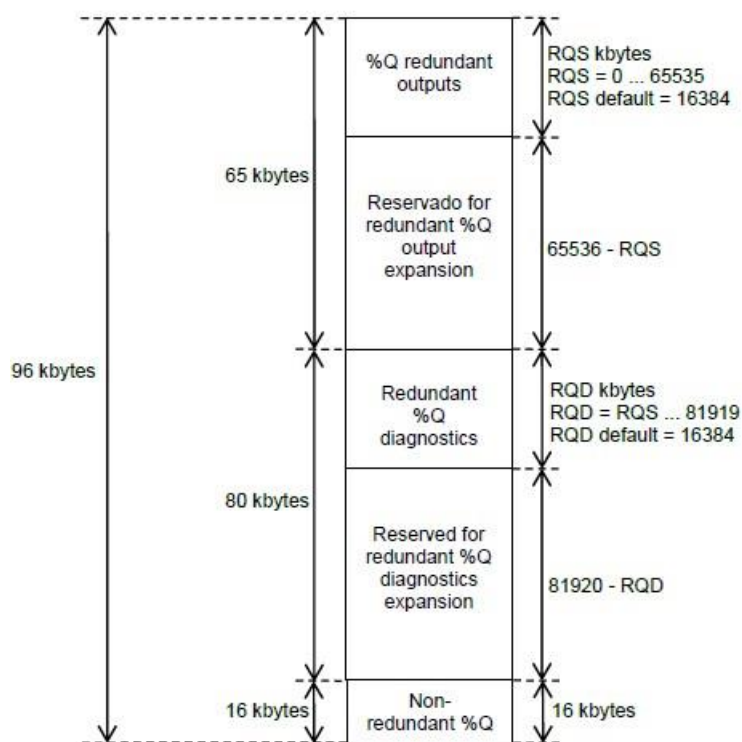


Рисунок 171: Резервированное и нерезервированное распределение %Q

6.3.3.9. Избыточные и неизбыточные переменные %M

ЦП NX3030 выделяет 64 Кбайт переменных %M (%MB0 ... %MB65535).

Все 65535 байт могут быть избыточными (%MB0000 ... %MB65535). По умолчанию количество избыточных переменных %M равно 0.

Следует избегать использования переменной %M и предпочитать использование символьных переменных (см. раздел «Избыточные и неизбыточные символьные переменные»).

6.3.3.10. Избыточные и неизбыточные символьные переменные

Помимо переменных прямого представления (%I, %Q и %M), которые распределяются автоматически, пользователь может явно объявить символические переменные внутри POU или GVL. Максимальный размер, допустимый для размещения избыточных символьных переменных, составляет 512 кбайт.

ВНИМАНИЕ

Символьные переменные не следует путать с переменными AT. Переменные AT — это просто символические имена, присвоенные переменным прямого представления (%I, %Q и %M) с использованием объявления «AT». Таким образом, переменные AT не выделяют память для символьных переменных.

Символьные переменные избыточны в следующих случаях:

При объявлении в POU из типа «программа», созданной в пользовательском приложении, в исключительных случаях программа NonSkippedPrg

При объявлении в GVL, созданных в пользовательском приложении, и эти GVL помечены как избыточные
Символьные переменные не являются избыточными в следующих случаях:

При объявлении в программе NonSkippedPrg. Эта программа была описана ранее в NonSkippedPrg.

Программный раздел

При объявлении в POU из типа «функция». Можно заметить, что эти POU обычно должны размещать переменные только в стеке (не статическом), что, следовательно, не должно быть избыточным. Если пользователь объявляет

7. ОБСЛУЖИВАНИЕ

статические переменные (VAR STATIC) внутри POU из типа «функция», это будет считаться плохим программированием. Такие статические переменные, в случае их создания, будут считаться неизбыточными.

При объявлении в POU типа «функциональный блок». Можно заметить, что простое объявление «функционального блока» не выделяет память (то, что выделяет память, — это превращение функционального блока в экземпляр)

Следует отметить, что экземпляры функциональных блоков, объявленные внутри POU из типа программы или внутри GVL, ведут себя как символические переменные, другими словами, выделяют избыточную память. Точно так же символические переменные, когда экземпляры функциональных блоков объявлены в программе NonSkippedPrg или когда GVL не помечен как избыточный, такие экземпляры не являются избыточными.

6.3.4. Множественное сопоставление

Если пользователь хочет сопоставить избыточные командные переменные более чем с одним коммуникационным портом (COMx или NETx), необходимо реализовать элемент управления пользователем в своем собственном приложении.

Реализуемая логика управления должна записывать в избыточные командные переменные на основе значений переменных (команд) из каждого коммуникационного порта (COMx или NETx). Кроме того, логика управления должна перезапускать командные переменные коммуникационных портов, так как управление резервированием просто перезапускает свои собственные командные переменные.

Ниже приведен пример этой реализации:

```
VAR var_StandBy_command_Ethernet_relation : BOOL;
var_StandBy_command_Serial_relation : BOOL;
var_Inactive_command_Ethernet_relation : BOOL;
var_Inactive_command_Serial_relation : BOOL;
var_TurnOn_command_Ethernet_relation : BOOL;
var_Turn_command_Serial_relation : BOOL; END_VAR

// Логика перевода локального ПЛК в ждущий режим
IF var_StandBy_command_Ethernet_relation = TRUE THEN
  DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal:=TRUE;
  var_StandBy_command_Ethernet_relation:=FALSE; END_IF
IF var_StandBy_command_Serial_relation = TRUE THEN DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal:=TRUE;
var_StandBy_command_Serial_relation:=FALSE;
END_IF
// Логика перевода локального ПЛК в неактивное состояние
IF var_Inactive_command_Ethernet_relation = TRUE THEN
  DG_NX4010.tRedundancy.RedCmdLoc.bInactiveLocal:=TRUE;
  var_Inactive_command_Ethernet_relation:=FALSE; END_IF
IF var_Inactive_command_Serial_relation = TRUE THEN
  DG_NX4010.tRedundancy.RedCmdLoc.bInactiveLocal:=TRUE;
  var_Inactive_command_Serial_relation:=FALSE; END_IF
// Логика включения локального ПЛК, выключенного PX2612
IF var_TurnOn_command_Ethernet_relation = TRUE THEN
  DG_NX4010.tRedundancy.RedCmdLoc.bTurnOnLocal:=TRUE;
  var_TurnOn_command_Ethernet_relation:=FALSE; END_IF
IF var_Turn_command_Serial_relation = TRUE THEN DG_NX4010.tRedundancy.RedCmdLoc.bTurnOnLocal:=TRUE;
var_Turn_command_Serial_relation:=FALSE;
END_IF
```

7. ОБСЛУЖИВАНИЕ

Выше приведен пример на языке ST, где команда резервирования может быть выполнена через две переменные с разных портов связи. В этом же примере были выполнены три разные команды (StandBy, Inactive и TurnOn). Где:

var_StandBy_command_Ethernet_relation: переменная логического типа, относящаяся к коммуникационной катушке Ethernet, которая

выполните команду, чтобы перевести локальный полукластер в ждущий режим.

var_StandBy_command_Serial_relation: переменная логического типа, относящаяся к катушке последовательной связи, которая будет выполняться.

cute команду перевести локальный Half-Cluster в ждущий режим.

DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal: эта команда выполняет действие, аналогичное действию кнопки STANDBY на PX2612 в локальном ПЛК.

var_Inactive_command_Ethernet_relation: переменная логического типа, относящаяся к коммуникационной катушке Ethernet, которая выполнит команду, чтобы перевести локальный полукластер в неактивное состояние.

var_Inactive_command_Serial_relation: переменная логического типа, относящаяся к катушке последовательной связи, которая выполнит команду перевода локального полукластера в неактивное состояние.

DG_NX4010.tRedundancy.RedCmdLoc.bInactiveLocal: эта команда выполняет действие, аналогичное кнопке INACTIVE на PX2612 в локальном ПЛК.

var_TurnOn_command_Ethernet_relation: переменная логического типа, относящаяся к коммуникационной катушке Ethernet, которая выполнит команду для повторной активации локального полукластера после его отключения реле PX2612.

var_Turn_command_Serial_relation: переменная логического типа, относящаяся к катушке последовательной связи, которая выполнит команду повторной активации локального полукластера после его отключения реле PX2612.

DG_NX4010.tRedundancy.RedCmdLoc.bTurnOnLocal: эта команда выполняет действие, аналогичное действию кнопки ВКЛЮЧИТЬ ПЛК на PX2612 в локальном ПЛК.

6.3.5. Диагностика, команды и структура пользовательских данных

Каждый ЦП имеет несколько структур данных, связанных с резервированием. Следующая структура представляет собой переменные AT, сопоставленные с переменными %Q:

RedDgnLoc: имеет диагностику ЦП (локальную), связанную с избыточностью, например, состояние избыточности ЦП.

RedDgnRem: это копия RedDgnLoc другого процессора, полученная по каналам синхронизации NETA/NETB. Таким образом, этот ЦП (локальный) имеет доступ к диагностике другого ЦП (удаленного).

RedCmdLoc: имеет команды, которые должны быть применены к этому ЦП (при вызове Local) или к другому ЦП (при вызове Remote). Например, поле StandbyLocal из этой структуры данных соответствует команде, которая должна быть выполнена в этом ЦП (локальном), а поле StandbyRemote соответствует команде, которая должна быть выполнена в другом ЦП (удаленный)

RedCmdRem: это копия RedCmdLoc другого процессора, полученная по каналам синхронизации NETA/NETB. Таким образом, этот ЦП (локальный) может выполнять команды, полученные от другого ЦП (удаленного).

RedUsrLoc: содержит 128 байт данных, свободно заполняемых пользователем (например, диагностика связи с системой SCADA).

Эти 128 байт данных могут быть обменены с другим ЦП (удаленным)

RedUsrRem: это копия RedUsrLoc другого процессора, полученная через NETA/NETB.

В разделе «Обслуживание резервирования» следующие подразделы предлагают более подробную информацию об этих структурах данных:

Структура диагностики резервирования

7. ОБСЛУЖИВАНИЕ

Команды резервирования

Обмен пользовательской информацией между PLCA и PLCB

6.3.6. Службы циклической синхронизации через NETA и NETB

В этом разделе описываются три службы синхронизации, которые циклически выполняются в резервированном ЦП между PLCA и PLCB через каналы синхронизации NETA и NETB.

Эти сервисы выполняются в начале каждого цикла MainTask и в последовательности, указанной ниже:

- Сначала выполняется служба диагностики и обмена командами.
- Во-вторых, выполняется служба резервной синхронизации данных.
- В-третьих, выполняется служба синхронизации резервного списка форсирования.

7. ОБСЛУЖИВАНИЕ

6.3.6.1. Диагностика и обмен командами

Эта служба отвечает за обмен следующими структурами данных в каждом цикле MainTask:

- Чтобы скопировать RedDgnLoc из PLCA в PLCB RedDgnRem
- Чтобы скопировать RedCmdLoc из PLCA в PLCB RedCmdRem
- Чтобы скопировать RedUsrLoc из PLCA в PLCB RedUsrRem
- Чтобы скопировать RedDgnLoc из PLCB в PLCA RedDgnRem
- Чтобы скопировать RedCmdLoc из PLCB в PLCA RedCmdRem Чтобы скопировать RedUsrLoc из PLCB в PLCA RedUsrRem

Служба будет выполняться с использованием только одного канала синхронизации (NETA или NETB). Таким образом, услуга может быть завершена, даже если один канал имеет проблемы.

6.3.6.2. Резервная синхронизация данных

Эта служба отвечает за перенос избыточных переменных из активного ЦП в неактивный ЦП. Как описано ранее, существуют символические избыточные переменные, а также избыточные переменные прямого представления (%I, %M и %Q).

Для выполнения этой услуги необходимо выполнение нескольких условий:

- Предыдущая служба синхронизации в этом цикле MainTask (обмен диагностикой и командами) должна быть выполнена успешно.
- Если этот ЦП находится в активном состоянии, другой должен быть в неактивном состоянии. С другой стороны, если этот ЦП находится в неактивном состоянии, другой должен быть в активном состоянии.
- Оба проекта (2 ЦП) должны быть идентичными, за исключением случаев, когда автоматическая синхронизация проектов отключена (см. раздел Отключение синхронизации проектов).

По крайней мере, один канал синхронизации (NETA и/или NETB) должен работать. Если оба канала синхронизации (NETA и NETB) работают, связь распределяется между обоими (балансировка нагрузки), чтобы сократить время синхронизации. В случае, если работает только один канал, синхронизация будет продолжать выполняться только по этому каналу, сохраняя избыточную синхронизацию данных.

6.3.6.3. Избыточная принудительная синхронизация списка

Эта служба отвечает за передачу избыточного списка форсирования с активного ЦП на неактивный ЦП.

Для выполнения этой услуги необходимо выполнение нескольких условий:

- Обе службы синхронизации, предшествующие этому циклу (диагностика и обмен командами), должны быть успешно завершены.
- Если этот ЦП находится в активном состоянии, другой должен быть в неактивном состоянии. С другой стороны, если этот ЦП находится в
- Неактивное состояние, другой должен быть в активном состоянии
- Оба проекта (2 ЦП) должны быть идентичными, за исключением случаев, когда автоматическая синхронизация проекта отключена (см. раздел Отключение синхронизации)

По крайней мере, один канал синхронизации (NETA и/или NETB) должен работать. Если оба канала синхронизации (NETA и NETB) работают, связь распределяется между обоими (балансировка нагрузки), чтобы сократить время синхронизации. В случае, если работает только один канал, синхронизация будет продолжать выполняться только по этому каналу, сохраняя избыточную синхронизацию данных.

ВНИМАНИЕ

Список избыточного форсирования имеет только форсирование избыточных переменных. На каждом CPU (PLCA и PLCB) может быть свой список форсирования, относящийся к неизбыточным переменным.

6.3.7. Службы спорадической синхронизации через NETA и NETB

Следующие службы синхронизации выполняются спорадически, другими словами, они не выполняются в каждом цикле MainTask. Другая системная задача выполняет эти спорадические службы в фоновом режиме.

6.3.7.1. Синхронизация проекта

Эта служба отвечает за синхронизацию проектов Active CPU и Non-Active CPU. Это происходит, когда проекты разные в обоих процессорах и на обоих процессорах включена автоматическая синхронизация проектов.

Синхронизация всегда происходит от активного ЦП к неактивному ЦП, и для выполнения этой службы достаточно, чтобы один из двух каналов синхронизации (NETA или NETB) работал.

Когда синхронизация включена, будут синхронизированы следующие файлы и сервисы:

- Приложение проекта (исполняемый код)
- Архив проекта (исходный код)
- Пользователи и группы
- Права доступа

Служба синхронизации запустится в течение тридцати секунд после перехода одного из ЦП в активное состояние, а после ее запуска CRC проекта будет проверяться каждые пять секунд.

Когда синхронизация запущена, неактивный ЦП переходит в режим остановки в состоянии «Не сконфигурирован». После передачи всех необходимых файлов неактивный ЦП переходит в состояние «Работа» в начальном состоянии. В случае сбоя передачи ЦП возвращается в состояние NotConfigured.

Время, необходимое для полной синхронизации, зависит от размера проекта. В среднем скорость передачи между каналами синхронизации составляет примерно 500 кбайт/с.

В случае прерывания синхронизации (потеря связи между каналами синхронизации) во время передачи файлов из активного ЦП в неактивный ЦП, процедура прерывается и перезапускается при восстановлении связи. Только после завершения всей процедуры Неактивный ЦП переходит в режим Run.

Помимо сохранения синхронизации проектов, синхронизация проектов также позволяет неактивному ЦП принимать более высокие состояния по отношению к запуску в случае, если CRC отличается или какое-либо онлайн-изменение должно быть выполнено в активном ЦП.

ВНИМАНИЕ

Синхронизация проекта будет иметь тот же эффект, что и загрузка в неактивный ЦП. Эта служба не выполняется, если автоматическая синхронизация проектов отключена, как описано в разделе «Отключение синхронизации проектов». Служба синхронизации между PLCA и PLCB не работает, если кабели каналов синхронизации перепутаны. Например, для подключения канала NETA в канал NETB от PLCB и канала NETB от PLCA в NETA в PLCB

ВНИМАНИЕ

При обновлении с версии 1.20 до более поздних версий MasterTool IEC XE была произведена модификация протокола связи между каналами синхронизации. Поэтому невозможно синхронизировать данные между двумя ПЛК, если одно из приложений было создано в версии до 1.21, а другое приложение было создано в равной или более поздней версии. Чтобы иметь возможность выполнить синхронизацию, вы должны выполнить действия, описанные в разделе Не загрузка приложения при запуске ПЛК с самым старым проектом. При этом приложение не будет загружено, но когда этот ПЛК перейдет в состояние «Не сконфигурировано» во время инициализации системы, приложения будут автоматически синхронизированы.

До версии 2.01 MasterTool IEC XE при отправке исходного кода на активный ЦП резервный ЦП переходил в состояние «Не настроен» для его синхронизации. Однако для завершения операции синхронизации ЦП оставался в состоянии «Не сконфигурирован», поэтому необходимо было перевести ЦП в состояние «Ожидание» с помощью кнопки «STAND-BY» на RX2612 или аналогичной команды. Начиная с версии 2.01 ЦП, находящийся в режиме ожидания, изменит свое состояние на Не настроено во время процесса синхронизации, но вернется автоматически, когда источники будут одинаковыми между двумя полукластерами.

6.3.8. Отключение синхронизации проекта

В разделе Службы спорадической синхронизации через NETA и NETB были описаны службы синхронизации проектов приложений и проектов архивов. Эти сервисы обычно должны быть активированы, и они полезны, когда изменения проекта могут быть загружены онлайн в активный ЦП, а затем в резервный ЦП автоматически через каналы синхронизации NETA/NETB.

Однако есть модификации проекта, которые нельзя загрузить в режиме онлайн ни в один ЦП, например, включение модулей в удаленный PROFIBUS или включение нового удаленного PROFIBUS. В этих случаях, используя резервирование ЦП и сети PROFIBUS, такие модификации могут быть выполнены без прерывания управления процессом. Процедура для достижения этой цели описана в разделе «Изучение избыточности для автономной загрузки модификаций без прерывания процесса управления».

В этой процедуре необходимо временно отключить синхронизацию проекта, позволяя на некоторое время одному процессору работать с новой версией проекта, в то время как другой процессор все еще работает со старой версией проекта.

ЦП NX3030 имеет энергонезависимый регистр для отключения синхронизации проекта, который позволяет отключать приложения проекта и службы синхронизации архива проекта. Этот регистр можно настроить с помощью MasterTool. Достаточно отключить синхронизацию проекта на одном из двух процессоров, чтобы он больше не работал.

Чтобы отключить синхронизацию проекта, пользователь должен, во-первых, подключиться к нужному ПЛК с помощью программного обеспечения MasterTool (см. раздел Соединение MasterTool с ЦП NX3030 с резервируемого ПЛК).

Затем в меню Online / Redundancy Configuration необходимо открыть поле со списком Project Synchronization, позволяющее выбрать два следующих параметра:

- Давать возможность
- Запрещать

Должна быть выбрана опция «Отключить» и нажата кнопка «Запись» в поле со списком. Сообщение информирует об успешной операции или нет.

Отключающая конфигурация синхронизации проекта не является частью избыточного проекта, разработанного в MasterTool. Такая конфигурация находится только в области энергонезависимой памяти ЦП, которую можно прочитать или записать с помощью MasterTool. MasterTool не сохраняет эту конфигурацию ни в одном файле.

Эта конфигурация копируется в каждом цикле MainTask из энергонезависимой памяти в DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable. Пользователь может проверить эту диагностику в ПЛК, чтобы убедиться, что команда выполнена успешно, поскольку ПЛК находится в режиме выполнения (DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable должно быть равно 1). Если ПЛК не находится в рабочем режиме, можно проверить конфигурацию непосредственно на дисплее ЦП NX3030 в ПЛК (см. раздел «Диагностика резервирования на графическом дисплее ЦП NX3030»).

Диагностику DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable также можно наблюдать в удаленном ПЛК через DG_NX4010.tRedundancy.RedDgnRem.sGeneral_Diag.bProjectSyncDisable (поскольку неактивный ПЛК находится в режиме выполнения). ПЛК (активный или неактивный) останавливает службу синхронизации проекта каждый раз, когда принимается любой из следующих битов:

DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable

- Это ПЛК, локальный бит. В этом ПЛК отключена синхронизация проекта DG_NX4010.tRedundancy.RedDgnRem.sGeneral_Diag.bProjectSyncDisable

- Другой ПЛК, удаленный бит. Удаленный ПЛК с отключенной синхронизацией проекта

ВНИМАНИЕ

Регистр отключения синхронизации проекта не является частью проекта резервного ЦП, поэтому он не сохраняется как его часть на компьютере, где выполняется MasterTool. Регистр сохраняется только в энергонезависимой памяти ЦП.

7. ОБСЛУЖИВАНИЕ

6.3.9. Конфигурация сети PROFIBUS

В каждом полукластере можно установить до четырех модулей PROFIBUS Master NX5001. Таким образом, мы можем определить до двух резервных сетей PROFIBUS, называемых PROFIBUS 1 и PROFIBUS 2, или до четырех простых сетей PROFIBUS, называемых PROFIBUS 1, PROFIBUS 2, PROFIBUS 3 и PROFIBUS 4, или даже одну резервную сеть и две простые, именуемые PROFIBUS 1, PROFIBUS 2 и PROFIBUS 3.

6.3.9.1. Резервирование PROFIBUS

Каждая из сетей PROFIBUS может быть резервированной или нерезервированной. Например, если сеть PROFIBUS 1 избыточна, она будет разделена на PROFIBUS 1 A и PROFIBUS 1 B. Если она не избыточна, будет существовать только PROFIBUS 1 A. То же самое относится и к PROFIBUS 2.

На рис. 164 показан пример с одной сетью PROFIBUS (PROFIBUS 1), которая является резервированной (PROFIBUS 1 A и PROFIBUS 1 B).

Только несколько удаленных типов могут быть подключены напрямую к этой резервированной сети PROFIBUS:

PO5063V5: PROFIBUS slave DP-V0 для пультов дистанционного управления серии Ponto

PO5065: PROFIBUS slave DP-V1 с Hart, для пультов дистанционного управления серии Ponto

AL-3416: PROFIBUS slave DP-V0 для ЦП AL-2004

NX5210: ведомый PROFIBUS DP-V0 для удаленных устройств серии Nexto

На рис. 164 также показана возможность подключения удаленных устройств без резервирования к этому типу резервированной сети PROFIBUS через модуль AL-2433 (ProfiSwitch). Такие нерезервированные пульта PROFIBUS могут быть любой марки и модели.

6.3.9.2. Режимы отказов PROFIBUS жизненно важные и некритические

Каждая из сетей PROFIBUS может быть сконфигурирована в двух разных режимах:

Критический сбой: в случае полного сбоя этой сети этот сбой может привести к переходу состояния резервирования в резервный ЦП (переключение). В случае резервной сети PROFIBUS полный отказ означает отказ обеих составляющих сетей.

Нежизненно важный отказ: даже если эта сеть полностью выйдет из строя, этот отказ не приведет к переходу состояния резервирования в резервный ЦП (переключение).

6.3.10. Резервные сети Ethernet с объединением сетевых карт

На рис. 164 показаны два примера сетей Ethernet с резервированием и объединением сетевых карт.

В первом случае ЦП NX3030 подключается к диспетчерской сети (SCADA), которая также используется для настройки через MasterTool. Оба порта Ethernet ЦП NX3030 (NET 1 и NET 2) образуют резервную пару NIC Teaming, подключенную к двум разным коммутаторам (Ethernet A и Ethernet B). В какой-то момент эти два коммутатора должны быть соединены между собой для соединения двух портов NIC Teaming и для еще более высокой доступности (против двойных сбоев).

Во втором случае два модуля NX5000 также образуют резервную пару NIC Teaming, соединенную между собой в двух разных коммутаторах (Ethernet HSDN A и Ethernet HSDN B). В какой-то момент эти два коммутатора должны быть соединены между собой для соединения двух портов NIC Teaming и для еще более высокой доступности (против двойных сбоев). Такие архитектуры Ethernet обеспечивают отличную доступность при сбоях портов Ethernet в кабелях и коммутаторах.

Если два модуля или интерфейса Ethernet образуют резервную пару NIC Teaming, конфигурация и включение устройств будут возможны только в первом интерфейсе. Во втором интерфейсе его параметры конфигурации будут заблокированы для редактирования.

7. ОБСЛУЖИВАНИЕ

Кластер из двух портов Ethernet, образующих пару NIC Teaming, имеет один IP-адрес, связанный с парой портов. Таким образом, клиенту, как SCADA или MasterTool, подключенному к серверу ЦП, не нужно беспокоиться об изменении IP-адреса в случае сбоя в любом парном порту NIC Teaming.

Каждый из интерфейсов Ethernet, образующих пару NIC Teaming, имеет уникальную диагностическую структуру, указывающую на сбой, которые в конечном итоге могут появиться в любом порту пары NIC Teaming.

Дополнительные сведения о настройке и диагностике объединения сетевых карт см. в следующих разделах:

Конфигурация портов Ethernet в ЦП NX3030 (NET 1 и NET 2) Конфигурация модулей NX5000

6.3.11. Способы смены IP

Кластер с резервированием серии Nexto имеет четыре метода изменения IP-адресов в портах Ethernet модулей NX5000 в каждом полукластере и один метод изменения IP-адресов в портах NET 1 и NET 2 ЦП NX3030. Эти методы определяют поведение портов в отношении их IP-адресов в соответствии с текущим состоянием полукластера (активный или неактивный) и с полукластером (PLCA или PLCB).

Существуют следующие методы: фиксированный IP-адрес, IP-адрес Exchange, активный IP-адрес и множественный IP-адрес.

Всего в списке может быть до четырех IP-адресов в зависимости от метода изменения IP-адреса.

6.3.11.1. Фиксированный IP

Это самый простой метод IP-адресации, который можно настроить в интерфейсах Ethernet в модулях Ethernet NX5000. В этом методе перечислены только IP-адреса из PLCA и из PLCB. Помимо состояния резервирования, активного или неактивного ПЛК, PLCA всегда будет отвечать настроенным IP-адресом, как и PLCB.

Cluster IP Addressing Method	
Cluster IP Addressing Method	Fixed IP
Cluster IP Addressing	
IP Address PLC A	192 . 168 . 15 . 68
IP Address PLC B	192 . 168 . 15 . 69
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253
Advanced...	

Рисунок 172: Фиксированный IP-метод

Параметры, которые необходимо настроить в методе «Фиксированный IP»:

- IP-адрес PLC A: адрес связи PLCA
- IP-адрес PLC B: адрес связи PLCB
- Маска подсети
- Адрес шлюза

6.3.11.2. Обмен IP

IP-адрес Exchange можно настроить в интерфейсах Ethernet в модуле Ethernet NX5000. В этом методе IP-адрес полукластера зависит от состояния ПЛК (активен или неактивен). При каждом переключении происходит изменение IP-адреса между полукластерами, что позволяет им принять IP-адрес из нового состояния резервирования.

PS: для этого метода адресации порты Ethernet от обоих ПЛК (PLCA и PLCB) предполагают одинаковый IP-адрес, когда они оба находятся в неактивном состоянии, создавая конфликт сетевых адресов. Учитывая необычность этой ситуации, когда ПЛК не управляет системой, это оказывается большой проблемой, и ее необходимо учитывать.

Cluster IP Addressing Method	
Exchange IP	
Cluster IP Addressing	
IP Address Active	192 . 168 . 15 . 68
IP Address Non Active	192 . 168 . 15 . 69
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253
Advanced...	

Рисунок 173: Автоматическая смена IP

Параметры, которые необходимо настроить в методе Exchange IP:

IP-адрес активен: адрес связи PLCA

IP-адрес не активен: адрес связи PLCB

Маска подсети

Адрес шлюза

6.3.11.3. Активный IP

Этот метод используется в резервированных сетях ЦП NX3030, и его также можно настроить в модулях NX5000. В этом методе есть IP-адрес для активного полукластера и еще два IP-адреса, один для PLCA, а другой для PLCB. В резервированных сетях ЦП NX3030 активный IP-адрес добавляется к интерфейсу активного ПЛК, и он может использовать активный IP-адрес или IP-адрес ПЛК для установления связи с ПЛК. С другой стороны, в Ethernet-модулях NX5000 активный IP-адрес заменяет неактивный IP-адрес ПЛК, когда ПЛК находится в активном режиме.

Cluster IP Addressing	
IP Address Active	192 . 168 . 15 . 1
IP Address PLC A	192 . 168 . 15 . 69
IP Address PLC B	192 . 168 . 15 . 70
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253
Advanced...	

Рисунок 174: Активный метод IP — резервный NX3030

Параметры, которые необходимо настроить в методе Active IP для NET резервного ЦП NX3030:

- IP-адрес активен: IP-адрес добавляется к интерфейсу, когда ПЛК находится в активном состоянии.
- IP-адрес PLC A: адрес связи PLCA, кроме его текущего состояния.
- IP-адрес PLC B: Коммуникационный адрес PLCB, кроме его текущего состояния Маска подсети Адрес шлюза

The screenshot shows a configuration window titled "Cluster IP Addressing Method". A dropdown menu is set to "Active IP". Below it, a section titled "Cluster IP Addressing" contains several input fields:

IP Address Active	192 . 168 . 15 . 68
IP Address PLC A Non Active	192 . 168 . 15 . 69
IP Address PLC B Non Active	192 . 168 . 15 . 70
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253

An "Advanced..." button is located at the bottom right of the configuration area.

Рисунок 175: Активный метод IP — NX5000

Параметры, которые необходимо настроить в методе Active IP для Ethernet-модулей NX5000:

IP Address Active: Активный коммуникационный адрес ПЛК. Заменяет IP-адрес неактивного PLCX.

IP Address PLC A Non Active: адрес связи PLCA, когда он находится в неактивном состоянии.

IP Address PLC B Non Active: Коммуникационный адрес PLCB, когда он находится в неактивном состоянии.

Маска подсети

Адрес шлюза

6.3.11.4. Несколько IP-адресов

Метод множественных IP-адресов можно настроить в интерфейсах Ethernet модулей Ethernet NX5000. В этом методе есть IP для каждого полукластера и для каждого состояния ПЛК. PLCA предполагает IP-адрес, когда он активен, и другой, когда он неактивен. То же самое происходит с PLCB в отношении его состояния (активен или неактивен).

The screenshot shows a configuration window titled "Cluster IP Addressing Method". A dropdown menu is set to "Multiple IP". Below it, a section titled "Cluster IP Addressing" contains several input fields:

IP Address PLC A Active	192 . 168 . 15 . 68
IP Address PLC A Non Active	192 . 168 . 15 . 69
IP Address PLC B Active	192 . 168 . 15 . 70
IP Address PLC B Non Active	192 . 168 . 15 . 71
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253

An "Advanced..." button is located at the bottom right of the configuration area.

Рисунок 176: Метод нескольких IP-адресов

Параметры, которые необходимо настроить в методе Multiple IP:

IP-адрес PLC A Active: Коммуникационный адрес PLCA, когда он находится в активном состоянии.

IP Address PLC A Non Active: адрес связи PLCA, когда он находится в неактивном состоянии.

IP Address PLC B Active: Коммуникационный адрес PLCB, когда он находится в активном состоянии.

7. ОБСЛУЖИВАНИЕ

IP Address PLC B Non Active: Коммуникационный адрес PLCB, когда он находится в неактивном состоянии.

Маска подсети

Адрес шлюза

6.3.12. Объединение сетевых карт и совместное использование активного IP-адреса

В случае, если определенная пара портов образует объединение сетевых карт в резервном ЦП, эти порты могут одновременно реализовывать стратегии объединения сетевых карт и активного IP.

Например, если порты NET 1 и NET 2 ЦП NX3030 образуют пару NIC Teaming, то:

IP Address PLC A: IP-адрес портов NET 1 + NET 2 в ЦП PLCA NX3030.

IP Address PLC B: IP-адрес портов NET 1 + NET 2 в ЦП PLCB NX3030.

IP Address Active: IP-адрес портов NET 1 + NET 2 в ЦП NX3030 в Активном ЦП.

Таким образом, отличная доступность стратегии NIC Teaming связана с практичностью стратегии Active IP, которая не требует сценариев в системах SCADA или других клиентах, подключенных к серверу Active CPU.

6.3.13. Использование интерфейсов Ethernet с индикацией существенных отказов

Порты Ethernet модулей NX3030 и NX5000 можно настроить на генерацию жизненно важных сбоев. Эта опция важна для приложений, в которых модули входов и выходов распределены по сети Ethernet. В этом случае, если на порту Ethernet произойдет сбой, произойдет переключение. Это поведение применимо только к портам Ethernet, где есть хотя бы коммуникационный драйвер, управляющий ошибкой.

Коммуникационные драйверы, вызывающие существенный сбой, — это клиент MODBUS и символьный клиент MODBUS (все ссылки на клиента MODBUS в следующих разделах относятся к обоим случаям). Драйверы сервера MODBUS, сервер символов MODBUS и ведущий EtherCAT не приводят к критическим отказам. Таким образом, если для порта Ethernet настроен клиентский драйвер MODBUS и в порту Ethernet произошел сбой, будет сгенерировано переключение, если включена опция существенной неисправности. Если драйвер, настроенный на порт Ethernet, является сервером MODBUS, даже если в двери произойдет сбой, он не вызовет критический сбой, который приведет к переключению.

Чтобы неисправность считалась существенной неисправностью порта Ethernet на клиенте MODBUS, все серверы, сконфигурированные в драйвере, должны быть неисправными. То есть, если на один и тот же Ethernet-порт настроено больше драйверов клиента MODBUS, считается критическим отказом, когда все серверы обоих клиентов неисправны.

Когда порт Ethernet настроен для работы с объединением сетевых карт, критический отказ будет рассматриваться только в случае отказа двух пар дверей.

6.3.13.1. Сбой в интерфейсе Ethernet

Переключение может быть вызвано сбоем в интерфейсе Ethernet, например потерей связи. Потеря связи может быть вызвана, например, обрывом кабеля или выходом из строя коммутатора в сети Ethernet. Соответственно, необходимо, чтобы в дополнение к настройке на генерацию жизненно важного сбоя был настроен экземпляр клиента MODBUS на интерфейсе Ethernet.

Когда интервал MainTask больше или равен 100 мс после обнаружения неисправности, переключение будет происходить в течение двух циклов MainTask. Когда интервал MainTask меньше 100 мс, переключение произойдет в течение 100 мс плюс время MainTask после обнаружения сбоя.

6.3.13.2. Сбой в подключенном сервере MODBUS

Время обнаружения неисправности на удаленном сервере MODBUS зависит от настроек тайм-аута, настроенных на каждом клиенте MODBUS. Когда неисправность обнаружена на всех Серверах, диагностика bAllDevicesCommFailure (см. раздел «Диагностика Modbus, используемая при резервировании») изменяет свое состояние на TRUE. Когда это произойдет, переключение произойдет через 3 секунды после этого перехода.

6.3.14. Использование связи OPC DA с резервными проектами

7. ОБСЛУЖИВАНИЕ

Протокол OPC DA можно настроить для связи с резервными кластерами через системы SCADA. Когда этот параметр выбран при создании резервного проекта, в проект добавляется объект конфигурации символов. В этом объекте задаются системные переменные, которые будут отправлены в систему SCADA. Эта опция связи включена в ЦП портов Ethernet NX3030. Дополнительную информацию о настройке связи OPC с резервными проектами см. в разделе «Конфигурация с ПЛК на сервере OPC DA с резервированием соединения» данного руководства.

6.3.15. Резервные состояния ЦП

В резервированной системе ЦП (PLCA или PLCB) может находиться в следующих состояниях:

- Активный
- Поддерживать
- Неактивный
- Не настроенный запуск

ВНИМАНИЕ

Часто в данном руководстве для каждого состояния, отличного от «Активное», используется обозначение «Неактивное», другими словами, для проектирования любого из остальных 4 состояний (Ожидание, Неактивное, Ненастроенное и Запуск). Активный ЦП — это тот, который находится в активном состоянии, а неактивный ЦП — это тот, который не находится в активном состоянии.

В следующих разделах кратко описаны эти пять состояний. Дополнительные сведения о состояниях избыточного ЦП описаны в разделе «Переход между состояниями избыточности», где также описываются конечный автомат и причины перехода.

6.3.15.1. Не настроенное состояние

Это начальное состояние резервирования. ЦП находится в этом резервированном состоянии:

- По соглашению, когда процессор выключен
- Перед запуском MainTask
- Перед переключением начального состояния
- В случае перезапуска с помощью команды Reset Warm, Reset Cold или Reset Origin
- В случае если MainTask выполняется в состоянии Not-Configured, выполняются следующие задачи:
- Мастера PROFIBUS отключены

Службы циклической синхронизации выполняются (см. раздел Службы циклической синхронизации через NETA и NETB), если выполняются условия для ее выполнения.

Службы спорадической синхронизации также могут выполняться (см. раздел Службы спорадической синхронизации через NETA и NETB).

CPU будет заблокирован в состоянии Not-Configured, если другой CPU находится в активном состоянии, и этот проект CPU отличается от проекта Active CPU (кроме случаев, когда автоматическая синхронизация проекта отключена — см. Отключение синхронизации проекта). В случае, если такой ситуации не возникает, переход из состояния Not-Configured в состояние Starting происходит сразу после поступления запроса на настройку.

Иногда CPU переходит в состояние Not-Configured, когда уже получен запрос на автоматическую настройку, когда новый запрос на изменение состояния Starting не требуется. Это происходит, например, при включении ЦП.

7. ОБСЛУЖИВАНИЕ

В других ситуациях пользователь должен вручную запросить эту конфигурацию, например, нажав кнопку на панели управления резервированием PX2612. Запросы на настройку вручную, как правило, не нужны, когда требуется техническое обслуживание пользователя перед выходом из состояния «Не настроено», например, если ЦП не достиг состояния Not-Configured из-за какого-либо сбоя.

После выхода из состояния Not-Configured ПЛК может вернуться в это состояние из-за таких событий, как:

- Перезапуск (сброс теплый, холодный или исходный)
- ПЛК выключен
- Различные проекты между этим ПЛК и активным ПЛК

6.3.15.2. Начальное состояние

В отличие от всех остальных 4 состояний, которые могут длиться бесконечно, начальное состояние является временным и занимает всего несколько секунд. Это состояние всегда достигается из состояния Not-Configured через запрос конфигурации.

В начале начального состояния выполняется несколько действий, тестов и проверок, чтобы решить, какое состояние будет следующим:

- Мастера PROFIBUS включены в пассивном состоянии. Пассивный режим используется для проверки передачи и приема.
- Цепи PROFIBUS и физический уровень, чтобы избежать скрытого сбоя
- Проверьте правильность идентификации ЦП (должно быть PLCA или PLCB)
- Проверьте, есть ли проблемы в параметрах конфигурации, извлеченных из проекта MasterTool.
- Проверьте целостность модуля NX4010.
- Службы циклической синхронизации выполняются (см. раздел Службы циклической синхронизации через NETA и NETB), если выполняются условия для ее выполнения.
- Проверьте версию совместимости прошивки между обоими процессорами.
- Убедитесь, что проекты обоих процессоров одинаковы, включена ли автоматическая синхронизация проектов (см. раздел Отключение синхронизации проектов)

Если другой ЦП находится в активном состоянии, проверьте возможность установления с ним пассивной связи PROFIBUS. Пассивный режим используется для тестирования каналов передачи и приема PROFIBUS и физического уровня, чтобы избежать скрытого отказа.

Если другой ЦП находится в неизвестном состоянии из-за сбоев в NETA и NETB, проверьте возможность установления с ним пассивной связи PROFIBUS. Если в проекте нет сети PROFIBUS и панели PX2612, проверьте, получают ли NET1/NET2 CPU пакеты Keep Alive из другого полукластера.

В зависимости от результатов этих проверок и тестов ЦП может перейти из состояния Starting в любое из остальных четырех состояний.

6.3.15.3. Активное состояние

В этом состоянии ЦП управляет автоматизированным процессом, используя программу ActivePrg, выполняемую только в этом состоянии. Активный ЦП также обновляет систему удаленного ввода-вывода PROFIBUS, переводя ее ведущие устройства PROFIBUS в активное состояние. Активное состояние используется для установления связи с удаленными устройствами PROFIBUS (ведомыми).

Активный ЦП также проверяет свою внутреннюю диагностику и пользовательские запросы на переключение, чтобы определить, необходимо ли переключение. ЦП выходит из активного состояния только в том случае, если он знает, что другой ЦП находится в режиме ожидания и может считать его активным.

7. ОБСЛУЖИВАНИЕ

Однако бывают ситуации, когда активный ЦП может выйти из активного состояния даже при отсутствии уверенности в том, что другой ЦП находится в состоянии ожидания (например, если ЦП выключен).

6.3.15.4. Резервное состояние

В этом состоянии ЦП готов к переключению в активное состояние, если есть запрос на это, как сбой в активном ЦП.

Резервный ЦП также проверяет свою собственную диагностику и может быть переключен в состояние «Не сконфигурировано» или «Неактивно» в случае возникновения каких-либо сбоев.

Мастера PROFIBUS включены в пассивном состоянии. Пассивный режим используется для тестирования каналов передачи и приема PROFIBUS и физического уровня, чтобы избежать скрытого сбоя. Полный сбой в сетях PROFIBUS, сконфигурированных как жизненно важные, приводит к переходу в неактивное состояние. Полный отказ в сети PROFIBUS повреждает как составляющие сети (резервированная сеть PROFIBUS), так и одну составляющую сеть (нерезервированная сеть PROFIBUS).

Если интерфейсы Ethernet включены с параметром жизненно важного сбоя, клиенты включаются в пассивном состоянии. Полные сбои в сетях Ethernet, настроенных как жизненно важные, приводят к переходу в состояние Неактивно. Полный сбой в сети Ethernet достигает двух составляющих сетей (параметр «Резервирование связи» включен) или единственной сети, входящей в состав (параметр «Резервирование связи» отключен).

6.3.15.5. Неактивное состояние

Это состояние обычно достигается после некоторых типов отказов или из-за ручного запроса перед запрограммированным обслуживанием.

Мастера PROFIBUS включены в пассивном состоянии. Пассивный режим используется для тестирования каналов передачи и приема PROFIBUS и физического уровня, чтобы избежать скрытого сбоя.

Перед переключением в другое состояние сначала должны быть исправлены диагностированные сбои или выполнено запрограммированное техническое обслуживание, если они привели ЦП в неактивное состояние.

После этого необходимо выполнить переход в состояние «Не настроено» с запросом конфигурации. Затем должен быть выполнен переход в состояние Starting. После состояния Start ЦП может:

Возврат в неактивное состояние, если определение типов ошибок остается

Возврат в состояние «Не настроено» в случае других типов сбоев.

Перейти в состояние ожидания, если другой ЦП находится в активном состоянии.

Перейти в активное состояние, если другой ЦП не находится в активном состоянии

6.3.16. Функции панели управления резервированием PX2612

Панель управления резервированием PX2612 показана на рис. 167, а на рис. 168 показан ее вид спереди с более подробной информацией. Кроме того, на рис. 169 показано, как эта панель должна быть подключена к полукластерам PLCA и PLCB.

PX2612 разделен на две секции: одна управляется PLCA, а другая — PLCB. Эти контроллеры возможны через кабели AL-2317/A для PLCA и AL-2317/B для PLCB и позволяют каждому ЦП считывать данные с трех кнопок, записывать на три светодиода и замыкающий контакт реле.

Соблюдая вид спереди на рис. 168:

PLCA выполняет чтение кнопок STAND-BY и INACTIVE в секторе PLC A

PLCA выполняет чтение кнопки TURN ON PLC B

PLCA выполняет запись трех светодиодов (ACTIVE, STAND-BY и INACTIVE) из сектора PLC A.

PLCA выполняет запись на реле RL B, используемом для отключения PLCB.

PLCB выполняет чтение кнопок STAND-BY и INACTIVE в секторе B PLC.

PLCB выполняет чтение кнопки TURN ON PLC A

PLCB выполняет запись трех светодиодов (ACTIVE, STAND-BY и INACTIVE) из сектора B PLC.

PLCB выполняет запись на реле RL A, используемом для отключения PLCA.

6.3.16.1. Кнопки PX2612

В этом разделе описываются функции кнопок PX2612.

Кнопка STAND-BY выполняет следующие функции:

Чтобы запросить переключение из активного состояния в резервное состояние, полезно, когда требуется техническое обслуживание активного ЦП. После переключения активного ЦП в режим ожидания (и, следовательно, резервного ЦП в активном состоянии) можно переключиться из режима ожидания в состояние неактивно с помощью кнопки НЕАКТИВНО, а затем выполнить запрограммированное техническое обслуживание в неактивном состоянии.

7. ОБСЛУЖИВАНИЕ

Для запроса конфигурации, которая вызывает переключение из состояния «Не настроено» в состояние «Начало», обычно после устранения сбоев, вызвавших переход в состояние «Не настроено». После начального состояния обычно предполагается, что ЦП переходит в состояние ожидания (или активен, если другой ЦП не находится в активном состоянии).

Чтобы запросить переключение из состояния Неактивно в состояние Ненастроено, уже запросив конфигурацию. Обычно это происходит после устранения сбоев, вызвавших переход в неактивное состояние. После состояния «Не настроено» конфигурация должна перейти в состояние «Начало». После начального состояния обычно предполагается, что ЦП переходит в состояние ожидания (или активен, если другой ЦП не находится в активном состоянии).

Кнопка INACTIVE запрашивает переключение из состояния Stand-by в состояние Inactive, что может быть полезно для выполнения запрограммированного обслуживания в Stand-by CPU. После этого обслуживания можно использовать кнопку STAND-BY, чтобы вернуть его в состояние Stand-by, минуя состояния Not-Configured и Starting (см. предыдущее описание кнопки STAND-BY).

Кнопка TURN ON PLCx (x = B для PLCA или x = A для PLCB) используется для повторной активации другого ЦП в случае, если локальный ЦП выключился. Как описано в разделе «Переход между состояниями резервирования».

Существуют исключительные ситуации, когда один ЦП выключает другой при переходе в активное состояние, чтобы исключить возможность одновременного перехода обоих ЦП в активное состояние.

ВНИМАНИЕ

Чтобы кнопка считалась, она должна быть нажата не менее 1 секунды. Кроме того, в течение этой секунды должна быть нажата только эта кнопка (остальные 2 кнопки должны быть отпущены).

ВНИМАНИЕ

Существуют альтернативные способы создания таких же эффектов кнопок STAND-BY, INACTIVE и TURN ON PLCx. Команды, сгенерированные локальным ЦП или удаленным ЦП, можно использовать, как описано предварительно в разделе «Диагностика, команды и структура пользовательских данных». Более подробное описание этих команд можно найти в разделе «Команды резервирования».

6.3.16.2. Светодиоды PX2612

Светодиоды PX2612 используются для информирования о состоянии резервирования, как показано в следующей таблице ниже:

Состояние резервирования	LED ACTIVE	LED STAND-BY	LED INACTIVE
Не настроено	выкл	выкл	выкл
Начиная	вкл	вкл	вкл
Активный	вкл	выкл	выкл
Активный (недавний)	мигает	выкл	выкл
Активен (отключение другого ЦП)	вкл	мигает	выкл
Активный (недавний и выключение другого процессора)	мигает	мигает	выкл
Поддерживать	выкл	вкл	выкл
Неактивный	выкл	выкл	вкл

Таблица 205: Светодиоды PX2612

Каждый светодиод может быть выключен, включен или мигать. Если он мигает, он остается включенным в течение 0,5 секунды и выключается в течение того же времени.

7. ОБСЛУЖИВАНИЕ

Обратите внимание, что существует четыре различных анимации для активного состояния из-за следующих особенностей:

В первые 2 секунды в активном состоянии светодиод ACTIVE мигает, а затем горит постоянно. Эта анимация была создана потому, что в первые моменты активного состояния ЦП не будет принимать команды для выхода из этого состояния. Дополнительные сведения об этом поведении активного ЦП см. в разделах «Переход между состояниями избыточности и первыми моментами в активном состоянии».

Если этот ЦП выключает другой ЦП через свое реле PX2612, светодиод STAND-BY мигает. В противном случае он остается выключенным

6.3.16.3. Реле PX2612

PX2612 имеет два реле NO. PLCA может управлять RL B, чтобы выключить PLCB. PLCB может управлять RL A, чтобы выключить PLCA.

Такие ситуации отключения случаются в исключительных ситуациях, описанных в разделе «Переход между состояниями резервирования».

6.3.17. Переход между состояниями резервирования

На следующем рисунке показан конечный автомат избыточности, иллюстрирующий все возможные переходы между состояниями избыточности.

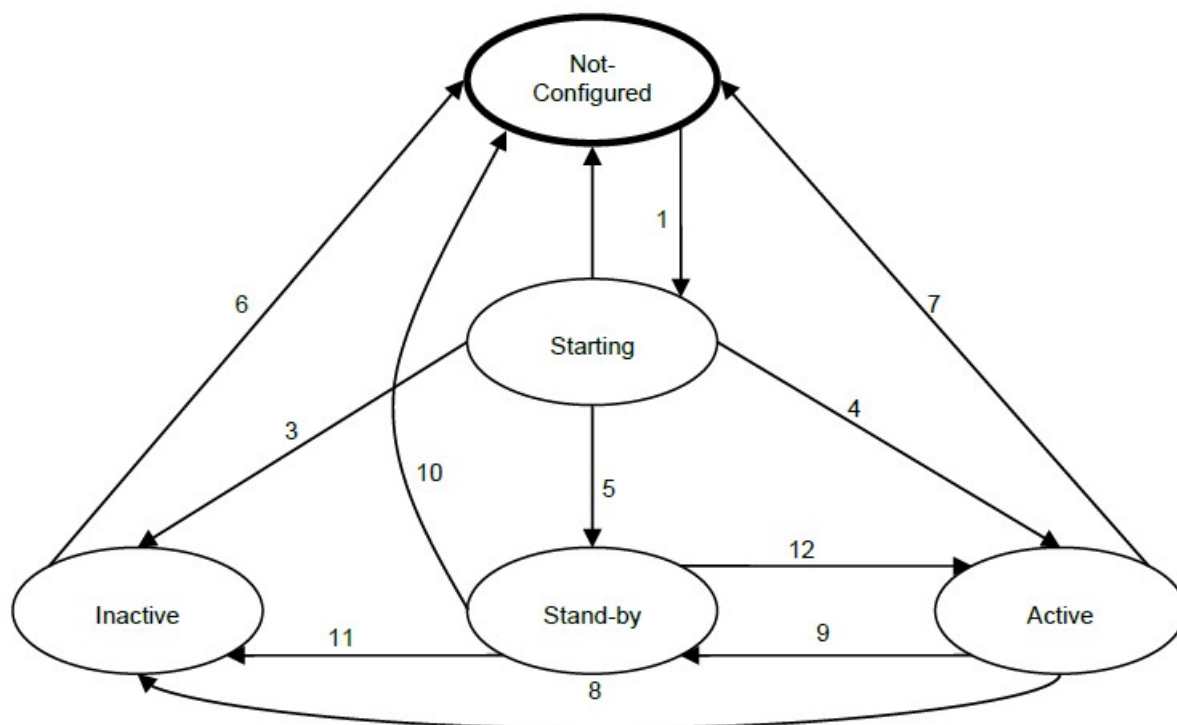


Рисунок 177: Конечный автомат избыточности

7. ОБСЛУЖИВАНИЕ

В следующих подразделах описаны все эти переходы и причины, которые могут их вызвать. Чтобы правильно интерпретировать работу этого конечного автомата, необходимо установить некоторые правила и последовательности:

Переходы, происходящие из одного и того же состояния, должны анализироваться в последовательности, установленной их номером. Например, переходы 2, 3, 4 и 5 происходят из начального состояния. В этом примере сначала анализируется переход 2, затем 3, 4 и, наконец, 5. В случае срабатывания перехода 2 переходы 3, 4 и 5 анализироваться не будут.

Внутри определенного подраздела, описывающего переход, его могут вызвать несколько условий. Эти условия должны быть проанализированы в той последовательности, в которой они указаны в подразделе. Любое условие, которое выполняется, может вызвать переход. Если условие вызывает переход, следующие условия анализировать не нужно.

Переходы могут запускаться только в том случае, если ЦП включен и выполняется MainTask. В противном случае предполагается, что CPU находится в состоянии Not-Configured.

В нескольких случаях упоминаются переходы, вызванные кнопками панели PX2612. Следует напомнить, что для этих кнопок существуют альтернативы, которые являются внутренними командами от одного ЦП или другого (через NETA/NETB). Такие команды были предварительно упомянуты в разделе «Диагностика, команды и структура пользовательских данных» и более подробно описаны в разделе «Команды резервирования». В следующих подразделах для упрощения эти команды не упоминаются, но нужно помнить, что они могут вызывать те же переходы, что и кнопка PX2612.

6.3.17.1. Переход 1 – Не настроено к запуску

ВНИМАНИЕ

Условия этого подраздела не должны анализироваться, если другой ЦП находится в активном состоянии и проекты разные. Этот CPU должен оставаться в состоянии Not-Configured, пока его проект отличается от проекта другого CPU, если другой находится в активном состоянии. Это замечание недействительно, если автоматическая синхронизация проекта отключена (см. раздел «Отключение синхронизации проекта»), так как в этом случае допускаются различия между проектами ЦП.

Запрос конфигурации уже существует в начале состояния Not-Configured. Это происходит в тот момент, когда

ЦП включен, а также другие ситуации, описанные в следующих подразделах, кнопка STAND-BY была нажата в состоянии Not-Configured. Это вызывает запрос ручной настройки. Пользователь обычно нажимает STAND-BY после устранения сбоя, который привел этот ЦП в состояние «Не настроен».

6.3.17.2. Переход 2 — от начала до ненастроенного

Этот процессор был выключен или перезапущен (Reset Warm, Cold или Origin), или его процессор перешел в режим остановки.

Идентификационный регистр этого ЦП недействителен (отличается от PLCA или PLCB)

В проекте, полученном от MasterTool IEC XE, имеются ошибки конфигурации логики.

Другой ЦП находится в активном состоянии, и версия микропрограммы в этом ЦП несовместима с версией микропрограммы в нем.

Другой ЦП находится в активном состоянии, и проект в этом ЦП отличается от проекта в нем. Помимо перехода в состояние Not-Configured выполняется запрос конфигурации. Таким образом, после синхронизации проектов ЦП автоматически переходит из состояния «Не сконфигурировано» в состояние «Начало». Это условие не анализируется, если автоматическая синхронизация проекта отключена (раздел «Отключение синхронизации проекта»)

6.3.17.3. Переход 3 — от начала до неактивности

7. ОБСЛУЖИВАНИЕ

Модуль NX4010 не обнаружен в шине или сбой его микропроцессора

Канал синхронизации (NETA или NETB) находится в состоянии сбоя, и этот ЦП знает, что этот сбой был вызван аппаратными компонентами или внутренним программным обеспечением (внутренние сбои NETA или NETB).

Другой процессор находится в активном состоянии. Однако невозможно синхронизировать избыточные данные или избыточный принудительный список.

Другое состояние ЦП невозможно обнаружить через NETA / NETB, но этот ЦП может отслеживать трафик в некоторых сконфигурированных сетях PROFIBUS в режиме серьезного сбоя. Таким образом, похоже, что другой ЦП контролирует процесс, хотя NETA/NETB не работает, чтобы подтвердить это.

При резервировании без панели PX2612 и без сети PROFIBUS, в случае, если состояние другого ПЛК не может быть известно через NETA/NETB, но этот ПЛК получает информацию о том, что состояние другого кластера АКТИВНО, посредством пакетов Keep Alive, полученных через NET1 или NET2 NX3030.

Произошла потеря связи с интерфейсом Ethernet, настроенным как критический сбой.

6.3.17.4. Переход 4 — от начала до активности

Другой ЦП находится в неактивном состоянии. Прежде чем переход станет возможным, это условие должно оставаться верным в течение некоторого времени, выше PLCB, чем PLCA. Таким образом, в данный момент PLCA и PLCB включены одновременно; PLCA имеет приоритет в активном состоянии

Другое состояние ЦП не может быть обнаружено через NETA/NETB, и, кроме того, этот ЦП не может отслеживать трафик в любой сети PROFIBUS, настроенной как режим критического отказа, или эти сети не были созданы. Следовательно, он действительно смотрит на другой ЦП, если он выключен или не работает. Из соображений безопасности, помимо переключения в активное состояние, этот ЦП выключает другой с помощью своего реле PX2612. Это условие должно сохраняться некоторое время перед выполнением перехода

6.3.17.5. Переход 5 – Переход в режим ожидания

Другой ПЛК находится в активном состоянии. Службы избыточной синхронизации данных и избыточной принудительной синхронизации списка работают правильно.

6.3.17.6. Переход 6 — от неактивного к ненастроенному

Этот ПЛК был выключен или перезапущен (теплый сброс, холодный сброс или сброс исходного состояния), или его ЦП перешел в режим останова.

На PX2612 была нажата кнопка STAND-BY. Помимо перехода в состояние Not-Configured выполняется запрос конфигурации. Таким образом, CPU автоматически переходит из состояния Not-Configured в состояние Starting. Обычно пользователь нажимает эту кнопку после устранения сбоя, из-за которого ЦП перешел в неактивное состояние.

У этого ПЛК отключена синхронизация, и проект отличается от активного ПЛК, при нажатии кнопки STAND-BY ПЛК переходит из неактивного в ненастроенный

6.3.17.7. Переход 7 — от активного к ненастроенному

Этот ПЛК был выключен или перезапущен (теплый сброс, холодный сброс или сброс исходного состояния), или его ЦП перешел в режим останова.

6.3.17.8. Переход 8 — от активного к неактивному

Модуль NX4010 не обнаружен в шине или сбой его микропроцессора. Этот ЦП знает, что другой ЦП находился в состоянии ожидания до того, как произошел этот сбой. Это условие не анализируется в первые 2 секунды в активном состоянии

Этот ПЛК потерял связь с другим ПЛК через NETA и NETB из-за внутреннего сбоя, но знает, что другой ПЛК находился в режиме ожидания непосредственно перед тем, как произошел сбой. Это условие не анализируется в первые 2 секунды в

Активное состояние

7. ОБСЛУЖИВАНИЕ

Этот ЦП не может управлять всеми сетями PROFIBUS, сконфигурированными в режиме критического отказа, и знает, что другой ЦП находится в состоянии ожидания. Это условие не анализируется в первые 2 секунды в активном состоянии

Этот ЦП обнаружил полный сбой в сетях Ethernet, настроенных в режиме серьезного сбоя, и знает, что другой ЦП находится в состоянии ожидания.

6.3.17.9. Переход 9 — активный в резервный

Оба ПЛК по какой-то причине находятся в активном состоянии, и этот конфликт необходимо разрешить. PLCA переходит в состояние ожидания, если этот конфликт сохраняется. PLCB делает то же самое после задержки, меньшей, чем PLCA. Таким образом, в этом случае PLCA имеет приоритет оставаться в активном состоянии.

Была нажата кнопка STAND-BY, и этот ЦП знает, что другой ЦП находится в состоянии ожидания. Это условие не анализируется в первые 2 секунды в активном состоянии

6.3.17.10. Переход 10 — из режима ожидания в состояние «не настроено»

Этот ПЛК был выключен или перезапущен (теплый сброс, холодный сброс или сброс исходного положения).

Другой ПЛК находится в активном состоянии, и известно, что этот проект ПЛК отличается от активного ПЛК. Помимо перехода в состояние Not-Configured выполняется запрос конфигурации. Таким образом, после синхронизации проектов ПЛК автоматически переходит из состояния «Не настроен» в состояние «Запуск». Это условие не анализируется, если автоматическая синхронизация проекта отключена (раздел «Отключение синхронизации проекта»)

Другой ПЛК находится в активном состоянии, и версия прошивки этого ПЛК несовместима с версией прошивки активного ПЛК.

6.3.17.11. Переход 11 — из режима ожидания в состояние неактивности

Модуль NX4010 не обнаружен в шине или сбой его микропроцессора

На PX2612 была нажата кнопка INACTIVE. Обычно это делается для выполнения запрограммированного обслуживания в неактивном ЦП. Следует избегать любого запрограммированного обслуживания резервного ЦП, поэтому рекомендуется переключиться в неактивный режим.

Другой процессор находится в активном состоянии. Однако избыточная синхронизация данных или избыточная принудительная синхронизация списка не работали в последних четырех циклах MainTask или служба синхронизации диагностики не работала в последних двух циклах MainTask.

Другой ПЛК находится в активном состоянии. Однако этот ПЛК не может отслеживать трафик в каждой сети PROFIBUS, сконфигурированной в режиме жизненно важного отказа.

Другой процессор находится в активном состоянии. Однако этот ЦП обнаружил сбой в портах Ethernet, настроенных как критический режим сбоя.

6.3.17.12. Переход 12 — из режима ожидания в активный режим

Другое состояние ЦП неизвестно из-за сбоев NETA и NETB. В этом случае, помимо перехода в активное состояние, из соображений безопасности этот ЦП отключает другой ЦП с помощью реле PX2612. Когда резервирование не использует панель PX2612 и PROFIBUS DP, ЦП использует механизм Keep Alive через порты NET1/NET2 NX3030, чтобы обмениваться информацией о состоянии между ПЛК и обнаруживать, что АКТИВНЫЙ больше не управляет процессом. Другое состояние ЦП известно и отличается от активного.

6.3.18. Первые мгновения в активном состоянии

В первые 2 секунды в активном состоянии, как уже описано в разделе «Функции командной панели резервирования PX2612», светодиод ACTIVE мигает и продолжает гореть по истечении этого времени.

Пока мигает светодиод ACTIVE, несколько переходов, которые обычно могут вывести ЦП из активного состояния, не анализируются (см. предыдущие подразделы, определяющие переходы из активного состояния). Например, в это время невозможно нажать кнопку STAND-BY, чтобы попытаться перевести ЦП в состояние ожидания.

7. ОБСЛУЖИВАНИЕ

Только два условия позволяют процессору выйти из активного состояния, пока мигает светодиод ACTIVE. Они следующие:

Этот ПЛК был выключен или перезапущен (теплый сброс, холодный сброс или сброс исходного состояния), что привело к переходу в состояние «Не сконфигурировано».

Оба ПЛК по какой-то причине находятся в активном состоянии, и этот конфликт необходимо разрешить. PLCA переходит в состояние ожидания, если этот конфликт сохраняется. PLCB делает то же самое после задержки, меньшей, чем PLCA. Таким образом, в этом случае PLCA имеет приоритет оставаться в активном состоянии.

Кроме того, в самые первые моменты, когда ПЛК переходит в активное состояние, некоторые нерезервированные диагностические данные могут оказаться недействительными, например, диагностические данные модулей NX5000 и NX5001. Метод, используемый для игнорирования диагностики, которая может оказаться недействительной, описан в разделе Чтение избыточной диагностики.

6.3.19. Распространенные сбои, вызывающие автоматическое переключение между полукластерами

В этом разделе перечислены наиболее распространенные сбои, которые автоматически вызывают переключение с активного ЦП на неактивный и с резервного ЦП на активный ЦП. Эти отказы запускают подгруппу тех переходов, которые рассматриваются в разделе «Переход между состояниями резервирования».

Неисправность источника питания в активном ЦП. Важно, чтобы оба ЦП имели резервные источники питания, чтобы сбой питания не повлиял на резервный ЦП.

Неисправность блока питания NX8000 в активном ЦП

Сбой шины стойки (NX9000, NX9001, NX9002 или NX9003) в активном ЦП Сбои в ЦП NX3030 из-за активного ЦП, например:

- Перегрузка (сброс теплый, холодный или исходный)
- Останавливаться
- Отказ шинных интерфейсов в одном или обоих каналах синхронизации NETA и NETB Отказы в NX4010 от активного ПЛК, например:
 - Нераспознанный модуль на шине ЦП NX3030.
 - Сбой в микропроцессоре NX4010, препятствующий обновлению внутренней диагностики NETA/NETB и панели управления PX2612 (кнопки, светодиоды и реле).
 - Внутренние сбои, влияющие на один или оба канала синхронизации NETA и NETB.

Полный отказ сети PROFIBUS активного ПЛК, если эта сеть сконфигурирована в жизненно важном режиме. В случае резервирования сети PROFIBUS обе составляющие сети должны выйти из строя (двойной отказ).

Полный отказ сети Ethernet в активном ЦП, если эта сеть настроена на критический отказ. Если сеть Ethernet избыточна, обе составляющие ее сети должны быть неисправны (двойной отказ).

6.3.20. Сбои, связанные с переключениями между полукластерами, управляемыми пользователем

Среди переходов, описанных в разделе «Переход между состояниями избыточности», некоторые позволяют пользователю управлять переключениями между полукластерами из-за сбоев, которые не вызывают автоматических переключений.

Есть очень частные случаи, которые зависят от философии каждого клиента. Например: случай, когда система SCADA теряет связь с активным ЦП, но продолжает поддерживать связь с резервным ЦП.

7. ОБСЛУЖИВАНИЕ

Некоторые клиенты предпочитают ручное переключение, когда оператор нажимает кнопку STAND-BY PX2612, на активный ЦП. Переключение вызывает повторную попытку связи с новым активным ЦП.

Альтернативным решением было бы вызвать переключение, отправив команду из системы SCADA на резервный ЦП, которая будет передана на активный ЦП через NETA/NETB, используя RedCmdLocal (резервный ЦП) и RedCmdRem (активный ЦП).) структуры данных для передачи команды, эквивалентной кнопке STAND-BY PX2612.

Также возможно, что активный ЦП сам обнаружит потерю связи с системой SCADA и активирует команду в RedCmdLocal, эквивалентную кнопке STAND-BY PX2612. Это будет полностью автоматическое решение без вмешательства оператора, которое обычно выполняется в POU ActivePrg.

С помощью структур данных, описанных в разделе «Диагностика, команды и структура пользовательских данных», можно обмениваться диагностикой и командами между полукластерами через NETA и NETB. Таким образом, пользователь может выполнить специальное управление резервированием на случай сбоев, которые обычно не приводят к переключению. Более подробная информация об этих структурах данных представлена в следующих разделах:

Структура диагностики резервирования

Команды резервирования

Обмен пользовательской информацией между PLCA и PLCB

Ниже показано, как пользователь может управлять сбоями и выполнять переключение из-за ошибки в интерфейсах Ethernet из Active PLC (этот код следует использовать в программном модуле ActivePrg):

```
// Убедитесь, что объединение сетевых карт включено.  
IF ((DG_NX3030.tDetailed.Ethernet.NET1.szIP = '0.0.0.0') OR (DG_NX3030.tDetailed.Ethernet.NET2.szIP = '0.0.0.0'))  
    THEN  
// Объединение сетевых карт включено: ошибка в двух сетях для выполнения переключения.  
IF (DG_NX3030.tDetailed.Ethernet.NET1.bLinkDown AND DG_NX3030.tDetailed.Ethernet.NET2.bLinkDown) THEN //  
Переведите локальный ПЛК в ждущий режим.  
DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal := TRUE;  
END_IF  
ELSE  
// Объединение сетевых карт отключено: ошибка в одной из сетей для выполнения переключения. IF  
(DG_NX3030.tDetailed.Ethernet.NET1.bLinkDown OR DG_NX3030.tDetailed.Ethernet.NET2.bLinkDown) THEN //  
Переведите локальный ПЛК в ждущий режим..  
DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal := TRUE;  
END_IF  
END_IF
```

ВНИМАНИЕ

Когда два интерфейса Ethernet образуют пару NIC Teaming, неактивный интерфейс всегда будет иметь IP-адрес 0.0.0.0. Это недопустимый IP-адрес, и невозможно вручную настроить интерфейс с этим адресом.

6.3.21. Отказоустойчивость

Основной целью резервного ЦП является повышение доступности системы. Доступность — это отношение времени, в течение которого система работает исправно, к общему времени, прошедшему с момента внедрения системы. Например, если система была внедрена 10 лет назад и за это время год не работала из-за сбоев, то ее

7. ОБСЛУЖИВАНИЕ

готовность составляла всего 90%. Такая доступность обычно неприемлема для критических систем, где требуется доступность 99,99% или даже больше.

Для достижения этого уровня доступности необходимо несколько стратегий:

Использование более надежных компонентов (с высоким MTBF или средним временем наработки на отказ), способствующих увеличению MTBF системы в целом.

Использование избыточности, по крайней мере, для наиболее важных компонентов или компонентов с меньшей средней наработкой на отказ таким образом, чтобы можно было допустить отказ компонента без остановки системы. Если избыточность реализована за счет дублирования компонентов, необходимо, чтобы оба вышли из строя, чтобы система в целом стала недоступной.

Широкий охват диагностикой, особенно в резервных компонентах. Избыточность компонентов не очень полезна для повышения доступности, когда невозможно определить, какой компонент вышел из строя. В этом случае первый сбой в одном компоненте все равно не приводит к падению системы, а остается скрытым до тех пор, пока не произойдет второй сбой, приводящий к падению системы, так как первый сбой еще не был устранен. Неисправности можно разделить на диагностируемые и скрытые. Настоятельно рекомендуется, чтобы все отказы резервных компонентов поддавались диагностике.

Также важно, чтобы нерезервированные компоненты имели широкий охват диагностикой, так как часто система может продолжать работу даже при отказе нерезервированного компонента. Компонент может не запрашиваться, например, реле с нормально разомкнутым контактом, обмотка которого редко срабатывает, неисправность не обнаруживается до того момента, когда система требует его закрытия

Низкое время ремонта нерезервированных компонентов. Отказ нерезервированного компонента может привести к падению системы, и во время ремонта система будет недоступна.

Возможность ремонта или замены резервного компонента без остановки системы. Если такая возможность существует, она получила большое увеличение доступности. В противном случае необходимо запрограммировать остановку для замены компонента, а время ремонта рассчитывается как время недоступности.

Низкое время ремонта резервных компонентов. Отказ резервного компонента не приводит к падению системы, но во время ее ремонта может произойти отказ в его резервной паре. По этой причине важно, чтобы неисправность была устранена быстро после диагностики. Чем больше время ремонта, тем выше вероятность повторного отказа резервируемого компонента за это время, что приведет к падению системы. Следовательно, чем больше время ремонта, тем ниже доступность системы. Запрограммируйте периодические автономные тесты в компонентах, чтобы обнаружить не диагностируемые системой сбои автоматически. Цель состоит в том, чтобы обнаружить скрытые отказы, особенно в избыточных компонентах или простых компонентах, которые не запрашиваются (например, реле безопасности). Автономные тесты иногда подразумевают остановку системы, что снижает доступность. Обычно для этой цели используются особые ситуации, такие как техническое обслуживание, запрограммированное процессом. Чем больше период между автономными испытаниями, тем больше время, в течение которого сбой может оставаться скрытым, и тем выше вероятность того, что сбой приведет к повреждению системы, иными словами, тем меньше доступность.

Эти принципы учитывались в проекте резервного ЦП с использованием NX3030.

В следующих подразделах анализируются несколько типов сбоев и то, насколько они допустимы или нет, а также существуют ли переключения, связанные с допустимыми сбоями.

6.3.21.1. Простой сбой с недоступностью

Некоторые компоненты, поскольку они не дублируются, не терпят даже простого отказа, не вызывая какой-либо недоступности. В резервированном ЦП, использующем ЦП NX3030, это связано со следующими компонентами:

- Удаленные устройства PROFIBUS (ведомые устройства) в сети PROFIBUS без резервирования
- Удаленные устройства Ethernet (ведомые устройства) в сети без резервирования. Модули ввода-вывода.

Проблема отказоустойчивости нерезервированной сети PROFIBUS может быть решена путем использования резервированной сети PROFIBUS, что рекомендуется в системах, требующих высокой отказоустойчивости. На рис. 164 показан пример архитектуры сети PROFIBUS с резервированием. Аналогично, нетерпимость к сбоям в сети Ethernet без резервирования может быть решена путем использования конфигурации сети Ethernet с резервированием с объединением сетевых карт.

7. ОБСЛУЖИВАНИЕ

Что касается недоступности модуля ввода-вывода, следует отметить, что это не подразумевает полную недоступность системы. Это представляет собой частичную недоступность только в сетке управления, которая использует этот модуль ввода-вывода.

Несмотря на то, что резервирование модулей ввода-вывода не предусмотрено, пользовательское приложение может управлять им в особых случаях. Например, пользователь может вставить 3 модуля аналоговых входов в 3 разных пульта PROFIBUS и реализовать схему голосования между тройками аналоговых входов для критической системы. Однако, как уже упоминалось, такими решениями должен управлять пользователь. Для них нет автоматической поддержки. Такие решения, вообще говоря, также подразумевают резервирование преобразователей и приводов в полевых условиях.

6.3.21.2. Простой сбой без недоступности, вызывающей переключение

Некоторые резервные компоненты допускают простые сбои, не вызывая недоступности, но вызывая переключение:

- Стойки (NX9000, NX9001, NX9002 или NX9003)
- Блок питания (NX8000)
- ЦП (NX3030)
- Модули NX4010
- Модули NX5001 (ведущие устройства PROFIBUS) в конфигурации сети PROFIBUS без резервирования
- Модули NX5000 (Ethernet) в конфигурациях без объединения сетевых карт

Ведомый интерфейс PROFIBUS в резервном удаленном устройстве (PO5063V5, PO5065, NX5210 или AL-3416). В этом случае, в отличие от предыдущего, переключение происходит внутри удаленного устройства, между сетями PROFIBUS A и B.

ВНИМАНИЕ

В случае отказа модуля ЦП NX3030 или NX4010 в архитектурах, где не используется панель PX2612 или сеть PROFIBUS, ЦП останется в своем текущем состоянии. В этом случае, если сбой происходит в активном полукластере, происходит простой системы.

6.3.21.3. Двойной сбой без недоступности, вызывающей переключение

Некоторые компоненты дублируются в каждом полукластере, таким образом, прежде чем вызвать переключение, оба должны выйти из строя:

Модули NX5001 (ведущие устройства PROFIBUS) в конфигурации с резервированием, сконфигурированные в режиме критического отказа.

Модули NX5000 (Ethernet) в конфигурациях с объединением сетевых карт (резервированием управляет пользователь).

6.3.22. Накладные расходы на избыточность

Избыточное приложение подразумевает увеличение времени обработки приложения по сравнению с необходимым временем для неизбыточного эквивалентного приложения.

Это дополнительное время возникает из-за выполнения служб циклической синхронизации, описанных в разделе Службы циклической синхронизации через NETA и NETB, и меньшего времени на управление резервированием (конечные автоматы и т. д.). Общее дополнительное время из-за избыточности (накладные расходы на избыточность) оценивается MasterTool после компиляции проекта избыточного ЦП.

ВНИМАНИЕ

Рассчитанные MasterTool накладные расходы учитывают пустой список избыточных переменных.

7. ОБСЛУЖИВАНИЕ

Кроме того, пользователь должен определить диапазон для MainTask относительно:

Необходимое время для выполнения основных программных модулей (NonSkippedPrg и ActivePrg). Это время обычно измеряется после разработки проекта (с учетом резервирования дополнительного времени).

Время, необходимое для обнаружения и генерации событий внутренних точек (например, возникновение 1000 событий аналоговых точек с зоной нечувствительности в одном цикле может занять до 30 мс)

Некоторая незакрепленность цикла MainTask для выполнения других задач ЦП (операционная система, драйверы I/O PROFIBUS, MODBUS и т. д.). Этот процент ослабления может варьироваться в зависимости от требуемой производительности от этих других задач. Например. если для связи MODBUS с системой SCADA требуется выделение слишком большого объема обработки для достижения удовлетворительной производительности, эта неплотность должна быть увеличена.

ВНИМАНИЕ

В зависимости от выравнивания памяти количество байтов, используемых в исчислении избыточности, может быть больше, чем общее количество байтов, объявленных в переменных.

6.4. Программирование резервного процессора

6.4.1. Мастер создания нового избыточного проекта

Чтобы создать новый резервный проект, необходимо использовать команду «Файл/Новый проект» и выбрать стандартный проект MasterTool.

Первоначально пользователь должен указать желаемое имя для проекта и каталог, в котором он хочет его сохранить, как показано на рисунке ниже.

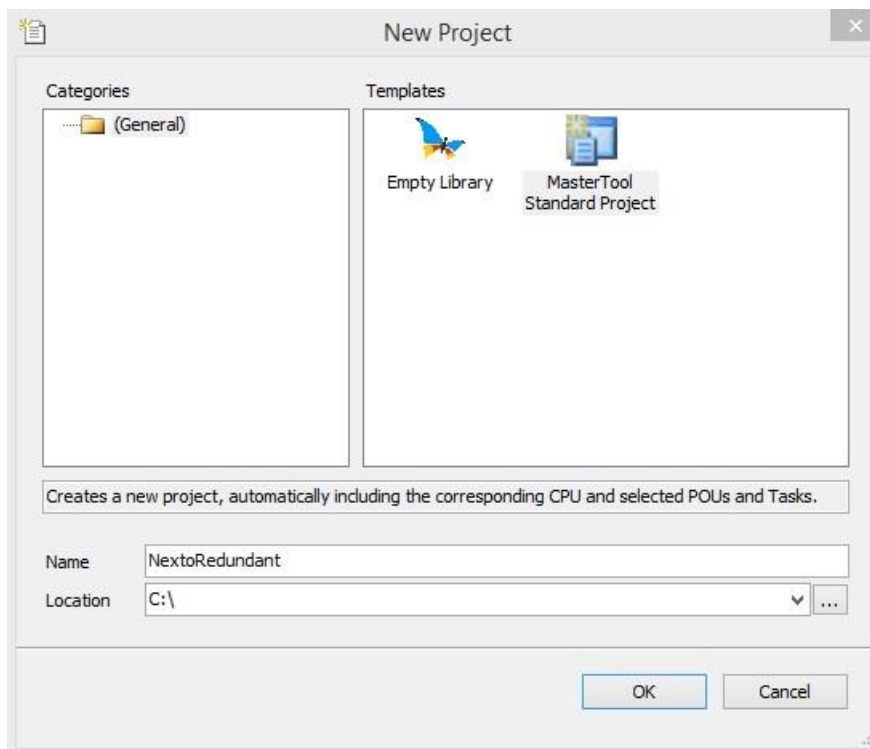


Рисунок 178: Новый проект

7. ОБСЛУЖИВАНИЕ

Затем мастер, создающий проект резервирования, задает пользователю несколько вопросов относительно желаемой конфигурации, на которые необходимо последовательно ответить.

Первый момент, который необходимо определить, — это начальная конфигурация аппаратного обеспечения полукластера:

- Выберите категорию устройства: NX3030 можно выбрать в двух категориях «Все устройства» или «Модульные контроллеры».
- Выберите модель ЦП: Поскольку резервирование реализовано только в NX3030, его должен выбрать пользователь.
- Выберите модель стойки: доступны четыре модели стойки, и выбор зависит от количества модулей, используемых в резервировании. Для MasterTool важен размер стойки в соответствии с настроенным количеством сетей (следующий пункт мастера) Выберите модель блока питания
- Выберите конфигурацию резервирования. Для избыточного проекта необходимо выбрать вариант With Redundancy.
- Выберите режим работы резервирования. В этом случае используются варианты с резервной панелью или без нее (PX2612).
- Выберите, будет ли включена опция связи OPC DA или нет.
- Выберите, будет ли использоваться резервирование расширения шины

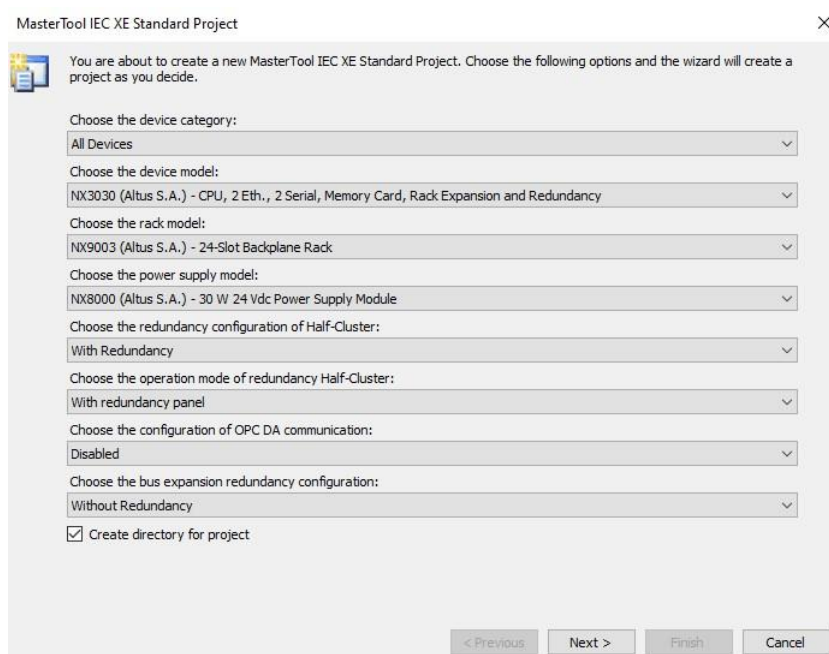


Рисунок 179: Начальная конфигурация оборудования

- После этого пользователь должен определить сети связи, используемые в резервированном приложении:
- Выберите категорию устройства: NX3030 можно выбрать из двух категорий: «Все устройства» или «Модульные контроллеры».
- Выберите количество сетей PROFIBUS: Мастером можно создать до четырех сетей PROFIBUS, и они могут быть одиночными или резервными. Важно подчеркнуть, что эта архитектура, предложенная Мастером, является типовой. После этого можно создать больше сетей PROFIBUS, соблюдая максимальное ограничение в четыре ведущих модуля PROFIBUS, NX5001, в каждом полукластере.

Выберите тип сети PROFIBUS:

- Нет (модуль NX5001 не выделен)
- Единая сеть (выделен один модуль NX5001)
- Резервированная сеть (располагает два модуля NX5001) Выберите тип сети Ethernet ЦП.
- Единая сеть с отключенным режимом отказа (не вызывает переключения в случае отказа)

7. ОБСЛУЖИВАНИЕ

- Единая сеть с включенным режимом отказа (генерирует переключение в случае отказа)
- Резервная сеть с отключенным режимом отказа (работает совместно с другим интерфейсом и не вызывает переключения в случае отказа)
- Резервная сеть с включенным режимом отказа (работает совместно с другим интерфейсом и генерирует переключение в случае отказа)

Выберите количество сетей Ethernet: В этом случае Мастер позволяет пользователю создать до четырех одиночных сетей, или до трех резервных сетей, или ни одной. Важно подчеркнуть, что это только архитектура, предложенная Мастером. После этого MasterTool позволяет создать всего до шести сетей (максимум три резервных), всегда соблюдая максимальное ограничение в шесть модулей Ethernet, NX5000, в каждом полукластере. Выберите тип сети Ethernet:

- Нет (модуль NX5000 не выделен)
- Единая сеть с отключенным режимом отказа (выделяет один NX5000 и не вызывает переключение в случае отказа)
- Единая сеть с включенным режимом отказа (выделяет один NX5000 и генерирует переключение в случае отказа)
- Резервная сеть с отключенным режимом отказа (выделяет два NX5000 и не производит переключение в случае отказа)
- Резервная сеть с включенным режимом отказа (выделяет два NX5000 и создает переключение в случае отказа).

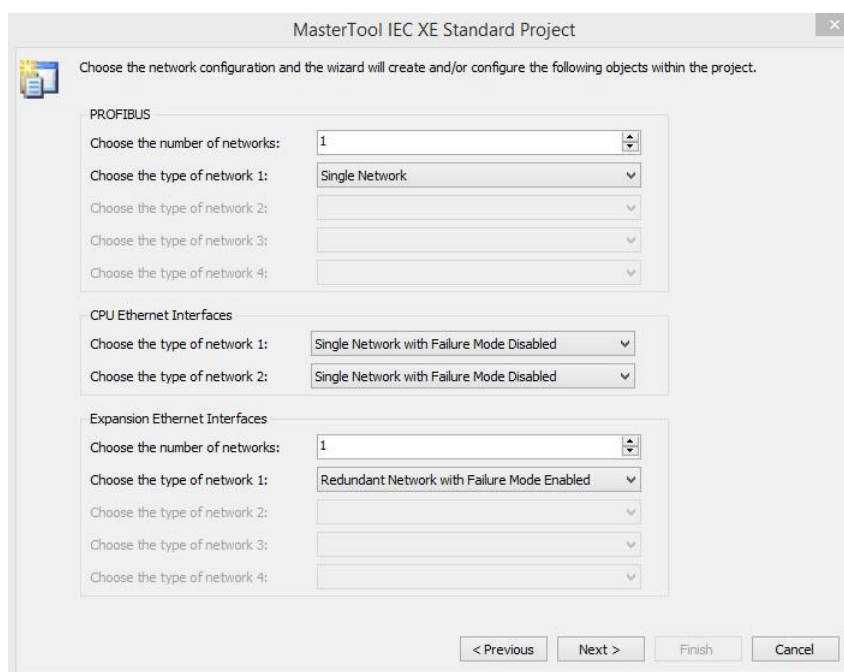


Рисунок 180: Конфигурация сетей связи

Затем необходимо выбрать профиль проекта и стандартный язык для создания программы:

Выберите профиль проекта: для резервирования можно использовать только профиль «Единый проект»; поэтому опция выбора отключена

Выберите язык по умолчанию для всех программ: язык, выбранный пользователем, является стандартным для всех программ, но любой другой может быть использован для конкретного ПМ.

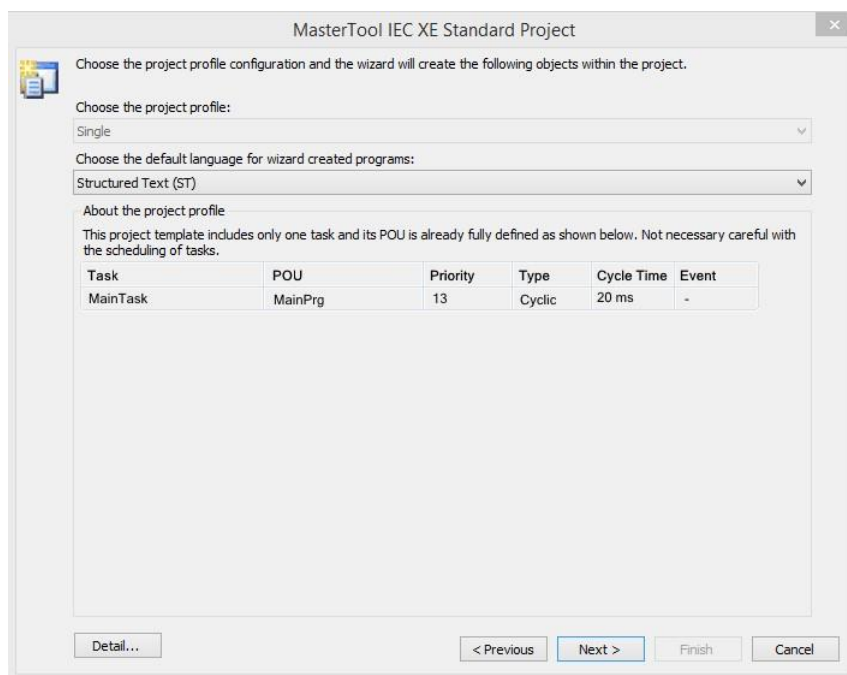


Рисунок 181: Профиль проекта и стандартный язык

Чтобы закончить, пользователь должен выбрать язык программы, общий и связанный с резервированием:

- Программа, связанная с MainTask (MainPrg): она должна быть обязательно на языке ST, так как MasterTool отключает другие параметры.
- Программы, связанные с резервированием Основные задачи

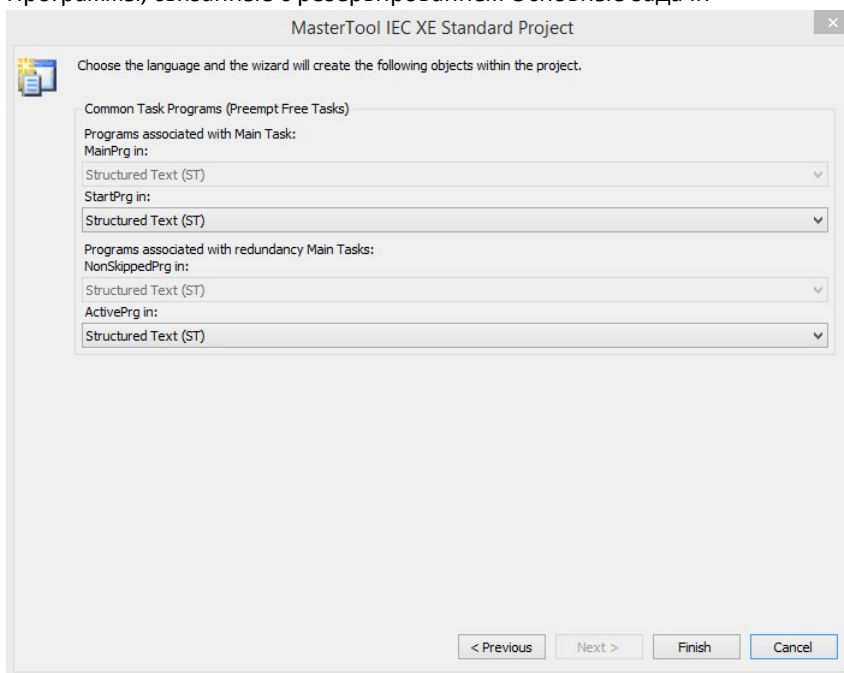


Рисунок 182: Конкретный язык программ

ВНИМАНИЕ

POU ActivePrg и NonSkippedPrg создаются автоматически, пустые, на языке, выбранном в предыдущих вопросах. Другие программные модули, созданные пользователем вручную, могут использоваться на любом доступном языке, за исключением избыточных программных модулей, которые не могут быть написаны на языке SFC, поскольку в качестве фона используется таймер IES. Для получения дополнительной информации см. Ограничения на программирование резервного ПЛК

ВНИМАНИЕ

Программный модуль MainPrg всегда автоматически генерируется на языке ST и не может быть изменен пользователем. Этот программный модуль вызывает программные модули ActivePrg (только в активном ПЛК) и NonSkippedPrg (в обоих ПЛК).

Получив ответы на предыдущие вопросы, Мастер формирует основной проект, определяя полукластер со следующей начальной аппаратной конфигурацией:

- Выбранная стойка
- Блок питания NX8000 (позиции 0 и 1)
- ЦП NX3030 (позиции 2 и 3)
- Модули NX4010 (позиции 4 и 5) и панель PX2612, если они выбраны.

После модуля NX4010 вставляются NX5001 для реализации сети PROFIBUS с функциями, ранее добавленными пользователем.

После модулей NX5001 вставляются модули NX5000 для реализации сети Ethernet с функциями, ранее добавленными пользователем.

6.4.2. Конфигурация полукластеров

Мастер всегда используется для создания первой версии избыточного проекта. Это гарантирует, что первоначальная версия будет сгенерирована быстро и правильно.

Однако возможно, что в полукластере потребуются некоторые модификации, например вставка новых модулей NX5001 и NX5000, которые можно выполнить, изменив экран конфигурации полукластера. В следующих разделах показано, как вставить и настроить модули NX5000, NX5001 и NX4010.

Для проекта с резервированием необходимо соблюдать некоторые правила и меры предосторожности, как описано в следующих разделах.

6.4.2.1. Фиксированная конфигурация в стойках от 0 до 5

В позиции от 0 до 5 выбранной стойки всегда должны быть установлены следующие модули:

- Блок питания NX8000 (позиции 0)
- ЦП NX3030 (позиции 2)
- Модуль NX4010 (позиции 4)

Эти модули нельзя удалять из исходного проекта, созданного Мастером.

Любая другая конфигурация в этих позициях приводит к ошибке, отображаемой MasterTool при компиляции проекта.

6.4.3. Конфигурация портов Ethernet в ЦП NX3030 (NET 1 и NET 2)

6.4.3.1. Конфигурация IP-адреса

На рисунке ниже представлена конфигурация порта NET 1 ЦП NX3030 (на экране конфигурации порта NET 2 есть подгруппа этих параметров). Чтобы открыть этот экран, необходимо выполнить двойной щелчок на NET 1 или NET 2 под CPU NX3030 в дереве устройств.

Cluster IP Addressing	
IP Address Active	192 . 168 . 15 . 1
IP Address PLC A	192 . 168 . 15 . 69
IP Address PLC B	192 . 168 . 15 . 70
Subnetwork Mask	255 . 255 . 255 . 0
Gateway Address	192 . 168 . 15 . 253

Advanced...

Рисунок 183: Параметры порта Ethernet NET 1

Далее необходимо отредактировать основные параметры интерфейсов NET 1 и NET 2. Адрес должен быть установлен в соответствии с методом Active IP Change, как описано в разделе Принципы работы — Методы изменения IP — Active IP.

ВНИМАНИЕ

The NET 1 and NET 2 interfaces IP addresses, as the Gateway Address, must belong to the same subnet.

ATTENTION

The NET 2 configuration screen has the same structure as the NET 1 configuration screen, but it doesn't have the checkbox Redundancy of Communication, neither the NIC Teaming configuration parameters.

6.4.3.2. Объединение сетевых карт между NET 1 и NET 2

Параметр Advanced на экране конфигурации NET 1 открывает новый экран конфигурации, который определяет, будет ли NET 1 резервной. Если установлен флажок «Избыточность связи», интерфейсы NET 1 и NET 2 образуют резервную пару с объединением сетевых карт, как описано в разделе «Принципы работы — избыточные сети Ethernet с объединением сетевых карт». Автоматически другие параметры включены и должны быть настроены:

Период проверки избыточности (мс): Период передачи кадра проверки связи между двумя сетями NET. Его можно настроить со значениями от 100 до 9900.

Повторы проверки избыточности: максимальное количество раз, которое сеть, отправившая кадр, будет ожидать ответа. Его можно настроить со значениями от 1 до 100.

Switching Period (s): Максимальное время, в течение которого Active NET будет ожидать любой пакет. Его можно настроить со значениями от 1 до 25.

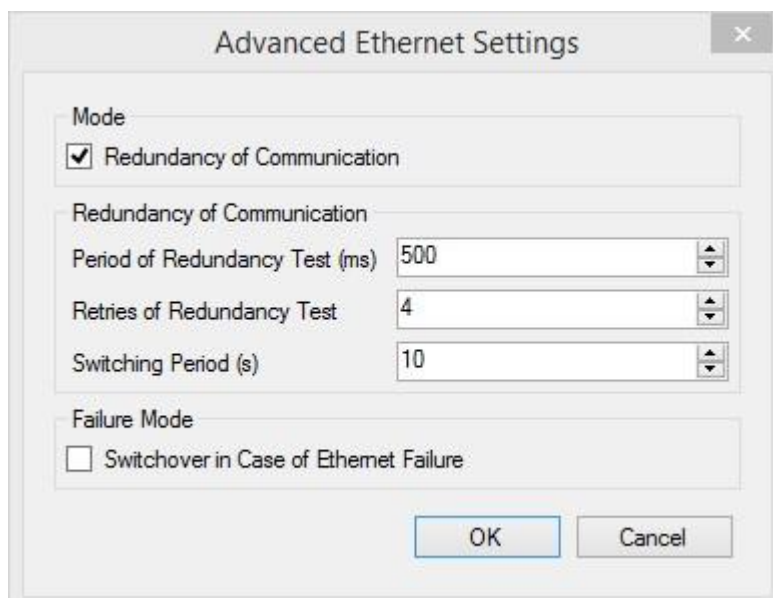


Рисунок 184: Расширенная конфигурация Ethernet

Если время ответа для теста избыточности достигает периода теста, умноженного на количество повторных попыток, и активный интерфейс остается на некоторое время дольше, чем период переключения, не получая никаких пакетов, произойдет переключение, в результате чего интерфейс, который был неактивен, становится активным. Важно подчеркнуть, что существует задержка между обнаружением сбоя и активацией неактивного интерфейса из-за времени, необходимого для настройки интерфейса. Эта задержка может достигать нескольких десятков миллисекунд.

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

ATTENTION

When a Reset Origin is performed in a CPU configured with NIC Teaming Включено for local Ethernet interfaces (NET 1 and NET 2), only the last active interface before the reset will be accessible. After the reset command, the accessible interface could be viewed in the [CPU's Informative and Configuration Menu](#).

6.4.3.3. Настройка критического отказа в NET 1 и NET 2

Параметр «Дополнительно» на экране настройки интерфейсов NET 1 и NET 2 открывает экран конфигурации, где помимо включения резервирования связи также можно настроить, будет ли интерфейс генерировать переключение в случае сбоя, как описано в Принципах работы. Использование интерфейсов Ethernet с индикацией существенных отказов.

При настройке в сочетании с избыточностью NIC Teaming сбой считается существенным сбоем, когда сбой происходит в интерфейсах NET 1 и NET 2.

6.4.4. Конфигурация модулей NX5001

6.4.4.1. Установка или удаление модулей NX5001

Модули NX5001 можно вставлять или извлекать из полукластерной стойки. Для правильного выполнения этой операции необходимо знать следующие правила:

Количество модулей NX5001 в каждом полукластере может варьироваться от нуля до четырех.

Можно определить до 4 простых сетей PROFIBUS или 2 резервных сетей PROFIBUS с соблюдением ограничения в 4 модуля PROFIBUS Master NX5001 в каждом полукластере.

7. ОБСЛУЖИВАНИЕ

Когда сеть PROFIBUS проста, ей требуется один модуль NX5001 в каждом полукластере. При резервировании требуется 2 модуля NX5001 в каждом полукластере.

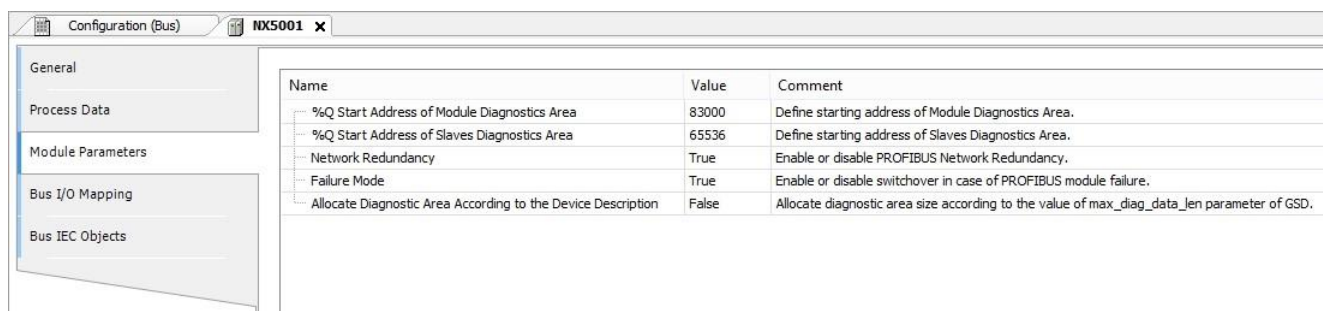
Два модуля NX5001, используемые для формирования резервированной сети PROFIBUS, должны располагаться рядом в стойке. Количество модулей NX5001 в стойке должно быть совместимо с количеством существующих сетей PROFIBUS и атрибутом резервирования каждой сети, другими словами:

- 0 x NX5001: сети PROFIBUS нет.
- 1 x NX5001: одна простая сеть PROFIBUS
- 2 x NX5001: В этом случае есть два варианта:
 - Две простые сети PROFIBUS
 - Одна резервная сеть PROFIBUS
- 3 x NX5001: В этом случае есть два варианта:
 - Три простые сети PROFIBUS
 - Одна резервная сеть PROFIBUS и одна простая сеть PROFIBUS.
- 4 x NX5001: В этом случае есть три варианта:
 - Четыре простые сети PROFIBUS
 - Одна резервная сеть PROFIBUS и две простые сети PROFIBUS.
 - Две резервные сети PROFIBUS

После установки или удаления модулей NX5001 необходимо проверить конфигурацию оставшихся в стойке модулей NX5001.

6.4.4.2. Настройка параметров модулей NX5001

Каждый модуль NX5001, используемый в простой сети PROFIBUS, или каждая резервная пара NX5001, используемая в резервированной сети PROFIBUS, имеет следующие параметры, которые необходимо настроить.



Name	Value	Comment
%Q Start Address of Module Diagnostics Area	83000	Define starting address of Module Diagnostics Area.
%Q Start Address of Slaves Diagnostics Area	65536	Define starting address of Slaves Diagnostics Area.
Network Redundancy	True	Enable or disable PROFIBUS Network Redundancy.
Failure Mode	True	Enable or disable switchover in case of PROFIBUS module failure.
Allocate Diagnostic Area According to the Device Description	False	Allocate diagnostic area size according to the value of max_diag_data_len parameter of GSD.

Рисунок 185: Параметры резервирования NX5001

Для группировки двух модулей NX5001 в резервированной сети PROFIBUS необходимо выполнить двойной щелчок на негруппированном модуле NX5001, справа от которого в стойке находится другой негруппированный модуль NX5001. Далее параметр Network Redundancy, доступный на вкладке Module Parameters, должен быть помечен как TRUE, как показано на рисунке 185. Чтобы его разгруппировать, необходимо выполнить ту же процедуру, но пометив параметр как FALSE. Если этот параметр помечен как TRUE, параметры DP и параметры NX5001 справа от него заблокированы для редактирования.

ATTENTION

In case of redundant networks, only the parameters of the NX5001 to the far left on the bus must be adjusted, while the NX5001 at the right remain blocked for edition. Some network parameters are identical to the other network while others are calculated automatically from network parameters of the left NX5001.

Рекомендуется, чтобы адрес ведущего устройства NX5001 в резервированном ПЛК был сконфигурирован равным 2, поскольку адрес ведущего устройства NX5001 в неактивном ПЛК уменьшается на одну единицу, поэтому адрес главного устройства NX5001 получается равным 1.

Помимо этого, важно помнить:

Адреса от 3 до 125 обычно используются для ведомых устройств PROFIBUS.

7. ОБСЛУЖИВАНИЕ

Адрес 0 часто используется для настройки и диагностики устройства.

Адрес 1 зарезервирован для динамического использования ведущим устройством PROFIBUS в неактивном ПЛК (ведущим устройством PROFIBUS в пассивном режиме).

Адрес 126 часто используется для ведомых устройств, если он исходит от производителя. Адрес 127 используется для широкопередаточных кадров.

При следующей компиляции проекта MasterTool проверяет возможные ошибки, которые мог допустить пользователь при установке или удалении модулей NX5001 вручную.

Важно отметить, что во время выполнения проекта, ранее сконфигурированного с резервными модулями NX5001, бит 0 команды (интерфейс включения канала %QXn.0 на вкладке Bus I/O Mapping) обрабатывается резервным приложением. Интерфейсы должны оставаться квалифицированными на протяжении всей программы. Таким образом, команда, запущенная пользователем для отключения интерфейса, не будет работать так, как ожидалось. Например, если в интерфейсе состояние этого бита изменилось с TRUE на FALSE на активном ЦП, это не будет интерпретироваться как сбой, который переводит ЦП в активное состояние для неактивного состояния. В этом случае ЦП останется в активном состоянии, а другой ЦП перейдет в неактивное состояние. По этим причинам пользователь не должен манипулировать этим командным битом в резервном приложении.

Дополнительную информацию о конфигурации сетей PROFIBUS см. в Руководстве по использованию PROFIBUS-DP NX5001.

6.4.4.3. Конфигурация удаленных устройств PROFIBUS

Чтобы настроить удаленные устройства PROFIBUS под управлением ведущего устройства NX5001, необходимо ознакомиться с Руководством по использованию ведущего устройства PROFIBUS-DP NX5001, а также со следующими руководствами:

Руководство по эксплуатации серии Ponto

Руководство по использованию головной части PROFIBUS PO5063V1 и резервной головки PROFIBUS PO5063V5

Головная часть PROFIBUS PO5064 и резервная головная часть PROFIBUS PO5065 Руководство по использованию HART через сеть PROFIBUS Руководство по использованию

Для резервированной системы мы должны обратить внимание на настройку параметра сторожевого таймера с пульта дистанционного управления PROFIBUS. Если на экране удаленной настройки установлен флажок Watchdog control, необходимо правильно настроить поле Time. Есть два варианта настройки времени, и мы должны использовать большее время между ними:

$WT \geq I \times 2 + 500$ мс; и $WT \geq I \times 3$;

Где WT — это время сторожевого таймера, а I — настроенный интервал MainTask.



Рисунок 186: Конфигурация сторожевого устройства PROFIBUS Remote

6.4.5. Конфигурация модулей NX5000

6.4.5.1. Установка или удаление модулей NX5000

Модули NX5000 можно вставлять или извлекать из полукластерной стойки. Чтобы правильно выполнить эту операцию, необходимо знать, что количество модулей NX5000 в каждом полукластере может варьироваться от нуля до шести. Следует обратить внимание на тот факт, что модули, образующие резервную пару NIC Teaming, должны быть вставлены в стойки рядом друг с другом.

При следующей компиляции проекта MasterTool проверяет возможные ошибки, которые мог допустить пользователь при установке или удалении модулей NX5000 вручную. Например, если пользователь вставил более 6 модулей NX5000, возникает ошибка.

7. ОБСЛУЖИВАНИЕ

Интерфейс каждого модуля будет идентифицирован как NET 1, так как они физически идентифицированы на изделии. В случае, если пользователь вручную добавляет модули NX5000 в шину, идентификация происходит так же, как и в мастере.

После установки или удаления модулей NX5000 необходимо проверить конфигурацию оставшихся в стойке модулей NX5000.

6.4.5.2. Конфигурация модулей NX5000

Для каждого модуля NX5000 в резервированном ПЛК необходимо отрегулировать параметры адреса, как описано в разделе «Принципы работы — методы изменения IP», доступ к которому можно получить, дважды щелкнув интерфейс NET 1 под каждым модулем NX5000, размещенным на устройствах.

ATTENTION

In case two consecutive modules form a redundant NIC Teaming pair, only the basic parameters of the left NX5000 should be edited, the right NX5000's parameters edition will be blocked.

6.4.5.3. Группировка модулей NX5000 с резервированием NIC Teaming

Модули NX5000, такие как ЦП NX3030 и интерфейс NX3020 NET 1, представляют экран расширенной конфигурации, который определяет, образует ли модуль резервную пару NIC Teaming с модулем справа от него. Конфигурация выполняется, как описано в разделе Объединение сетевых карт между NET 1 и NET 2.

Чтобы сгруппировать два модуля NX5000 с резервной парой, должны выполняться следующие условия:

Оба модуля NX5000 должны быть вставлены в стойку близко друг к другу.

При этом правый модуль блокирует редактирование своих параметров, а параметры левого модуля становятся одинаковыми для обоих модулей. Снятие флажка «Избыточность связи» в левом модуле приводит к разделению модулей, благодаря чему они снова ведут себя как отдельные модули без избыточности объединения сетевых карт.

6.4.5.3.1. Неисправность Жизненно важная настройка

Модули NX5000, а также интерфейсы NET 1 и NET 2 позволяют настроить, будет ли интерфейс генерировать переключение в случае сбоя, как описано в разделе «Принципы работы — использование интерфейсов Ethernet с индикацией существенных отказов».

При настройке в сочетании с избыточностью NIC Teaming существенный отказ будет считаться, если отказ произойдет в обоих модулях резервной пары.

6.4.6. Конфигурация резервирования NX4010

Доступ к конфигурации резервных переменных %I, %Q и %M можно получить, дважды щелкнув модуль NX4010 после выбора вкладки Redundancy Parameters.

Чтобы понять эти параметры, необходимо прочитать разделы «Резервные и нерезервированные переменные %I», «Резервные и избыточные переменные %Q» и «Резервные и избыточные переменные %M».

Следующие параметры должны быть настроены:

Configuration	Description	Default	Options
	Memory (%M)		
Redundancy %M Memory Offset	Redundant %M memory initial address	0	0 (Выключено)
Redundancy %M Memory Length	Redundant %M memory size	0	0 to 65536
	Memory (%I)		
Redundancy %I Memory Offset	Redundant %I memory initial address	0	0 (Выключено)

Redundancy %I Memory Length	Redundant %I memory size	16384	0 to 81920
	Memory (%Q)		
Redundancy %Q Memory Offset Reserved For I/O Drivers	%Q redundant memory offset reserved for I/O drivers initial address	0	0 (Выключено)
Redundancy %Q Memory Length Reserved For I/O Drivers	%Q redundant memory offset reserved for I/O drivers size	16384	0 to 81920
Redundancy %Q Memory Offset Reserved For Diagnostics	%Q redundant memory offset reserved for diagnostics initial address	65536	0 to 81919
Redundancy %Q Memory Length Reserved For Diagnostics	%Q redundant memory offset reserved for diagnostics size	16384	0 to 81920

Таблица 206: Параметры NX4010

6.4.7. Конфигурация драйверов ввода/вывода

Конфигурация драйверов ввода-вывода, во-первых, не отличается от нерезервированного ЦП.

Можно заметить, что некоторые драйверы ввода-вывода имеют команды, которые позволяют использовать его в резервном ЦП, но это не подразумевает различий в конфигурации. Эти команды обычно должны выполняться в программе NonSkippedPrg. Например, главный драйвер MODBUS RTU в последовательной сети RS-485 должен быть отключен в неактивном ЦП с помощью кода, введенного пользователем в NonSkippedPrg. Дополнительную информацию об администрировании драйвера MODBUS в резервированной системе можно найти в разделе «Управление экземплярами MODBUS в резервированной системе».

В случае сети PROFIBUS также существуют специальные команды для процессоров в активном и неактивном состояниях. Однако в этом случае управление резервированием выполняет такие команды автоматически, без какого-либо управления пользователями.

Для настройки удаленных устройств ввода/вывода PROFIBUS, включая удаленные устройства и модули ввода/вывода, см. раздел Конфигурация модулей NX5001 в этом руководстве.

6.4.8. Конфигурация основной задачи

Доступ к экрану конфигурации, связанному с единственной задачей резервного ЦП, называемой MainTask, которая является циклической, можно получить, щелкнув MainTask в дереве устройств.

На этом экране необходимо отрегулировать два параметра:

Интервал основной задачи
Время сторожевого таймера

Кроме того, на экране отображается оценка необходимого времени для управления резервированием, рассчитанная MasterTool. Такая оценка надежна только после завершения проекта, когда все программные модули разработаны и определены избыточные области памяти. Чтобы правильно настроить время цикла MainTask, необходимо принять во внимание несколько соображений:

Время интервала должно быть достаточно малым, чтобы обеспечить надлежащее управление процессом, принимая во внимание все времена обратной связи управления. Время интервала должно быть достаточно большим, чтобы обеспечить, по крайней мере, сумму следующих времен:

- Максимальное время выполнения POU NonSkippedPrg и ActivePrg вместе
- Время, необходимое для управления резервированием (накладные расходы на резервирование)

Кроме того, время интервала должно иметь дополнительную погрешность, необходимую для времени выполнения других процессов (связь PROFIBUS, связь Ethernet со SCADA-системами и т. д.).

MasterTool имеет условия расчета необходимого времени для управления избыточностью (резервирование накладных расходов) после завершения проекта (определены все разработанные POU и избыточные области памяти).

Что касается максимального времени выполнения POU NonSkippedPrg и ActivePrg, то их можно измерить после того, как эти POU уже разработаны. Первоначально MasterTool оценивает максимальное время этих двух POU в 10 мс вместе, но впоследствии пользователь должен пересмотреть это поле при измерении с использованием окончательного проекта.

7. ОБСЛУЖИВАНИЕ

После каждой компиляции MasterTool суммирует накладные расходы на избыточность, рассчитанные с параметром, который сообщает время POU (NonSkippedPrg и ActivePrg), и проверяет, соблюдается ли минимальное параметризованное ослабление.

Например.:

Параметры, настраиваемые на экране MainTask:

- Интервал основной задачи: 100 мс
- расчетное время POU NonSkippedPrg + ActivePrg: 10 мс
- Минимальный допуск: 30%

Расчетные накладные расходы на резервирование: 50 мс

В этом случае общее используемое время составляет 60 мс (10 мс + 50 мс), что составляет 60% цикла MainTask (100 мс). Таким образом, максимальное ослабление составляет 40%, а минимальное ослабление 30% соблюдается.

ATTENTION

A compilation error is produced in case the minimum looseness isn't respected, if it is configured in the CPU Project Parameters.

ATTENTION

The compilation being successful or not, MasterTool informs the calculated looseness and the redundancy overhead predicted on the message window.

6.4.8.1. Программа ActivePrg

В этом POU пользователь должен создать основное приложение, отвечающее за управление его процессом. Этот POU вызывается основным POU (MainPrg) и выполняется только в активном CPU.

Пользователь также может создавать дополнительные POU (программы, функции или функциональные блоки) и вызывать их или создавать экземпляры внутри POU ActivePrg, чтобы структурировать свою программу. Также можно вызывать функции и экземпляры функциональных блоков, определенные в библиотеках.

Следует помнить, что все символические переменные, определенные в POU ActivePrg, как экземпляры функциональных блоков, являются избыточными переменными. Символические переменные, определенные в дополнительных POU из типа программы, которые вызываются внутри ActivePrg, также являются избыточными переменными.

ВНИМАНИЕ

Переменные типа VAR_TEMP не должны использоваться в резервной программе.

6.4.8.2. Программа NonSkippedPrg

Этот программный модуль используется для управления, которое должно выполняться в обоих процессорах (PLCA и PLCB), независимо от состояния резервирования. Этот POU также вызывается основным POU (MainPrg).

Следует помнить, что все символические переменные, определенные в POU NonSkippedPrg, а также экземпляры функциональных блоков, являются неизбыточными переменными. Пользователь должен создать дополнительные POU (программу, функцию или функциональный блок) и вызвать или создать их экземпляр внутри POU NonSkippedPrg, чтобы структурировать свою программу. Также можно вызывать функции и экземпляры функциональных блоков, определенные в библиотеках.

ВНИМАНИЕ

Следует избегать вызова дополнительных POU из типа программы внутри NonSkipped- Prg, так как символические переменные, объявленные в этом типе POU, являются избыточными, а внутри NonSkippedPrg обычно желательны неизбыточные переменные. Обычно код NonSkippedPrg небольшой и не требует вызова дополнительных POU из типа программы для своей структуры. Если требуется структура NonSkippedPrg, необходимо использовать функциональные блоки или функции.

7. ОБСЛУЖИВАНИЕ

Типичными примерами элементов управления, выполняемых в NonSkippedPrg, являются следующие:

Создать компактную диагностическую структуру (%Q) для передачи в систему SCADA из полной диагностической структуры, в которой многие диагностические данные не представляют интереса для SCADA-системы. Эти диагностические данные могут быть извлечены из таких структур данных, как RedDgnLoc, RedDgnRem, RedUsrLoc, RedUsrRem и т. д.

Копировать команды, полученные от SCADA-системы, для соответствующих полей структуры данных RedCmdLoc и при необходимости соединять эти команды между собой

Для управления переключениями, контролируруемыми пользователем, в случае несущественных сбоев, таких как связь с системой SCADA или с устройством MODBUS.

Включайте и отключайте некоторые определенные драйверы ввода-вывода в зависимости от состояния резервирования (активен или неактивен). Например. ведущий драйвер MODBUS RTU на шине RS-485 должен быть отключен в неактивном ЦП. Для получения дополнительной информации см. MODBUS

Раздел «Управление экземплярами в резервной системе»

ВНИМАНИЕ

Не рекомендуется использовать функциональные блоки TOF_RET, TON_RET, TOF и TON в программе NonSkippedPrg. См. Ограничения на программирование резервного ПЛК.

6.4.9. Объект конфигурации резервирования

Этот объект, расположенный в дереве устройств, создается мастером автоматически. Он используется для определения того, какие POU и GVL являются резервными и поэтому синхронизируются между ПЛК. По умолчанию POU и GVL, созданные пользователем, помечаются как избыточные, оставляя пользователю возможность отменить маркировку при необходимости.

ВНИМАНИЕ

Объекты PV, PIDControl и PidRetainGVL не могут быть помечены по отдельности. В случае необходимости внесения изменений необходимо отметить опцию Выбрать все.

6.4.10. GVL Module_Diagnostics

Этот специальный GVL создается и заполняется мастером автоматически и не может быть изменен пользователем.

Системная диагностика и команды, включая структуру данных резервирования (RedDgnLoc, RedDgnRem, RedCmdLoc, RedCmdRem), помещаются в специальные переменные прямого представления %Q или %I.

В GVL Module_Diagnostics есть много предложений с ключевым словом AT для определения символических имен для этих диагностик и команд. Таким образом, когда пользователю нужно сослаться на эти переменные, он может использовать символическое имя вместо числовой ссылки.

6.4.11. GVL с избыточными символическими переменными

Пользователь может создать другие GVL, отличные от ранее перечисленных, чтобы объявить избыточные символические переменные. Для этого после создания GVL необходимо отметить его в объекте Конфигурация резервирования, в дереве устройств проекта. По умолчанию все GVL, созданные пользователем, изначально являются избыточными..

ВНИМАНИЕ

Для хорошей практики рекомендуется избегать использования директивы AT в GVL, которые имеют избыточное объявление символических переменных, чтобы предотвратить сопоставление переменных в избыточных областях.

6.4.12. POU из типа программы с избыточными символическими переменными

7. ОБСЛУЖИВАНИЕ

Пользователь может объявить избыточные символические переменные в POU из типа программы, за исключением POU NonSkippedPrg, где объявленные символические переменные считаются избыточными.

Чтобы определить новый POU как резервный, он должен быть отмечен в объекте Redundancy Configuration после его создания в дереве устройств проекта. По умолчанию все POU, созданные пользователем, изначально являются избыточными

ВНИМАНИЕ

Для хорошей практики рекомендуется избегать использования директивы AT в POU, которые имеют избыточное объявление символьных переменных, чтобы предотвратить отображение переменных в избыточных областях.

6.4.13. Использование точек останова в резервных системах

Для резервированных систем рекомендуется использовать точки останова только в активном полукластере, а другой полукластер деактивирован. В противном случае, когда выполнение приложения достигает точки останова, резервная точка останова принимает активное состояние, отключая активный ПЛК.

6.4.14. Управление экземплярами MODBUS в резервированной системе

В случае отключения существенной неисправности интерфейсов Ethernet или в случае сервера экземпляров MODBUS, экземпляры MODBUS не зависят от избыточности и, следовательно, должны управляться в приложении, на усмотрение пользователя, какие экземпляры следует включать/отключать, когда ПЛК переходит в неактивное состояние. Когда для портов Ethernet с клиентом MODBUS разрешена критическая ошибка, нет необходимости реализовывать дополнительный код для управления переключением.

Пример ниже, вставленный в программу NonSkippedPrg, выполняет проверку текущего состояния ПЛК и, если он находится в неактивном состоянии, отключает экземпляры ведущего и подчиненного устройства MODBUS RTU и экземпляр сервера MODBUS Ethernet:

```
VAR eRedStateLocal : REDUNDANCY_STATE;
eRedStateLocal_old : REDUNDANCY_STATE; END_VAR

// Чтение текущего состояния локального ПЛК eRedStateLocal :=
DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.eRedState;

// Изменилось ли состояние локального ПЛК?
IF eRedStateLocal <> eRedStateLocal_old THEN
IF eRedStateLocal = REDUNDANCY_STATE.ACTIVE THEN
// Локальный ПЛК перешел в активное состояние
Diagnostics.DG_MODBUS_RTU_Slave.tCommand.bRestart := TRUE;
Diagnostics.DG_MODBUS_RTU_Master.tCommand.bRestart := TRUE;
Diagnostics.DG_MODBUS_Server.tCommand.bRestart := TRUE;
ELSE
// Локальный ПЛК перешел в неактивное состояние
Diagnostics.DG_MODBUS_RTU_Slave.tCommand.bStop := TRUE;
Diagnostics.DG_MODBUS_RTU_Master.tCommand.bStop := TRUE;
Diagnostics.DG_MODBUS_Server.tCommand.bStop := TRUE;
END_IF
// Сохраняет последнее состояние локального ПЛК eRedStateLocal_old:=
eRedStateLocal;
END_IF
```

6.4.15. Ограничения на программирование резервного ПЛК

На резервированном ПЛК существуют некоторые ограничения в отношении его полукластерного программирования. Эти ограничения рассматриваются в подразделах ниже.

6.4.15.1. Ограничения в резервных GVL и POU

7. ОБСЛУЖИВАНИЕ

В резервированном GVL или POU программного типа для корректного функционирования полукластеров необходимо соблюдать следующие ограничения:

Не используйте переменные типа VAR_TEMP

Не смешивайте типы переменных (VAR, VAR RETAIN, VAR PERSISTENT и т. д.). В каждом GVL или POU должен использоваться только один тип.

Не смешивайте объявление символических переменных с AT в GVL. Должны быть созданы отдельные GVL, где в одном будут объявлены переменные AT, а в другом — символические переменные.

Не храните адрес переменной в избыточной переменной (используйте избыточную переменную в качестве указателя), так как адреса переменных могут отличаться в PLCA и PLCB.

Не используйте функциональные блоки для чтения и записи RTC в резервных POU. Более подробную информацию можно найти в разделе Часы RTC.

6.4.15.2. Ограничения программы без резервирования (NonSkippedPrg)

В POU из типа программы, которые не являются избыточными, в случае POU NonSkippedPrg, для корректного функционирования полукластеров необходимо соблюдать следующие ограничения:

Традиционные функциональные блоки TON и TOF нельзя использовать, поскольку они используют таймер IEC. Когда резервный ПЛК переходит в активное состояние (при этом другой полукластер выходит из активного состояния), таймер IEC синхронизируется, вызывая разрыв в значении таймера. Вместо этого должны использоваться функциональные блоки TON_NR и TOF_NR, доступные в библиотеке NextoStandard. См. Таймер без резервирования

POU из типа программы, написанной на языке SFC (последовательная функциональная схема), не должны использоваться, так как они используют таймер IEC для синхронизации перехода.

Не смешивайте объявление символических переменных с AT в GVL. Должны быть созданы отдельные GVL, где в одном будут объявлены переменные AT, а в другом — символические переменные.

6.4.16. Получение состояния избыточности полукластера

Состояние резервирования полукластера можно проверить в структуре диагностики резервирования:

VAR

```
eRedStateLocal : REDUNDANCY_STATE; END_VAR
```

```
eRedStateLocal := DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.eRedState;
```

Таким образом, пользователь может управлять логикой программы, которая зависит от состояния резервирования ПЛК.

6.4.17. Чтение избыточной диагностики

Проект с резервированием, помимо существующей резервной диагностики (Структура диагностики резервирования или диагностика с удаленного устройства PROFIBUS), представляет также нерезервированную диагностику (диагностика от модулей NX5000, NX5001, NX3030 и т. д.). Эти нерезервированные диагностические данные могут быть недействительными и не должны рассматриваться в первые моменты времени в активном состоянии, поскольку они не синхронизированы с другим ПЛК (состояние диагностики, когда удаленный ПЛК был активен, неизвестно). Следовательно, эти диагностические данные следует игнорировать в первые моменты активного состояния, пока они не будут иметь допустимых значений. Обычно время, в течение которого диагностика не должна рассматриваться, составляет 5 с.

В приведенном ниже примере показано, как не учитывать диагностические сообщения bSlaveNotPresent и bPbusCommFail от ведущего модуля NX5000 PROFIBUS.

Логика в NonSkippedPrg:

```
PROGRAM NonSkippedPrg
VAR
TON_DiagEnable : TON_NR; bDiagEnable :
BOOL; blsActiveState : BOOL; blsActiveState_old
: BOOL; END_VAR

blsActiveState := (DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.eRedState = REDUNDANCY_STATE.ACTIVE);
TON_DiagEnable(IN:= (blsActiveState = blsActiveState_old), PT:= T#5S, Q=> bDiagEnable);
blsActiveState_old := blsActiveState;

Logic in ActivePrg:
IF NonSkippedPrg.bDiagEnable THEN
IF DG_NX5001.tGeneral.bSlaveNotPresent OR DG_NX5001.tGeneral.bPbusCommFail THEN
// Действия, выполняемые при активной диагностике
END_IF
END_IF
```

6.5. Загрузка программы резервного ЦП

В разделе «Программирование резервного ЦП» описаны вопросы, связанные с разработкой проекта резервного ЦП с ЦП NX3030.

В этом разделе описаны многие методы и шаги для загрузки этого проекта в резервный ЦП с учетом таких ситуаций, как:

Загрузка проекта в новый ЦП NX3030 или в ЦП с неизвестным проектом

Онлайн скачивание модификаций

Автономная загрузка модификаций с прерыванием управления процессом, при программной остановке процесса

Автономная загрузка модификаций без прерывания управления технологическим процессом, с использованием функций резервирования

6.5.1. Первоначальная загрузка избыточного проекта

В этом разделе описываются необходимые шаги для запуска первой загрузки резервного проекта в ЦП NX3030. Это необходимо, например, для недавно выпущенного нового процессора или для процессора с неизвестным проектом.

ВНИМАНИЕ

Следующие шаги должны быть выполнены для обоих полукластеров (PLCA и PLCB), которые составляют резервный ЦП. Сначала все шаги должны быть выполнены для одного процессора, а затем для другого.

6.5.1.1. Первый шаг — определение IP-адреса для подключения к MasterTool

Первым шагом является обнаружение IP-адреса из канала NET 1 в этом ЦП для подключения MasterTool.

Это необходимо сделать с помощью дисплея и кнопки ЦП NX3030, как описано в разделе «Информационное и конфигурационное меню ЦП». Меню NETWORK сообщает IP-адрес, который можно использовать для подключения MasterTool.

6.5.1.2. Второй шаг — проверка конфликта IP-адресов

7. ОБСЛУЖИВАНИЕ

Перед выполнением третьего шага необходимо убедиться, что к сети не подключено другое оборудование с таким же IP-адресом, обнаруженное на первом шаге. Это можно обнаружить, например, отключив ЦП от сети и выполнив «пинг» по его IP-адресу. Поскольку ЦП отключен от сети, функция «ring» должна завершиться ошибкой. Если нет, то есть оборудование с таким же IP-адресом.

Если IP-адрес уже используется оборудованием в сети, необходимо выполнить третий шаг, а также некоторые из следующих шагов, используя перекрестный кабель для подключения MasterTool к ЦП, избегая конфликта IP-адресов. В одном из следующих случаев при загрузке проекта в ЦП в нем обновляются окончательные IP-адреса (см. раздел Конфигурация портов Ethernet в ЦП NX3030 (NET 1 и NET 2)).

6.5.1.3. Третий шаг — подготовка подключения к MasterTool (установка активного пути)

Третий шаг заключается в двойном щелчке устройства (ПЛК NX3030) в дереве устройств, переходе на вкладку «Настройки связи», щелчке шлюза и нажатии кнопки «Сканировать сеть», чтобы вывести список всех процессоров, обнаруженных MasterTool в сети.

В этот момент должен появиться ЦП, идентификация которого имеет IP-адрес, найденный на первом шаге. Если пользователь ранее изменил имя сетевого процессора, это имя будет отображаться. В разделе «Соединение MasterTool с ЦП NX3030 из резервированного ПЛК» более подробно описаны возможные идентификации, которые можно увидеть в этом списке. Так или иначе, у всех возможных идентификаций есть поле, показывающее IP-адрес или его часть. Например, байты в квадратных скобках образуют адрес процессора. Правый байт внутри скобок указывает на конец IP-адреса в шестнадцатеричном формате. Если байты образуют адрес [0010], это означает, что байт со значением «10» указывает, что конец IP-адреса ЦП равен xxx.xxx.xxx.16. Далее необходимо щелкнуть ЦП в списке и нажать кнопку «Установить активный путь». После этого выбранный ЦП должен появиться в списке под нагрузкой, указывая на то, что MasterTool готов подключиться к этому ЦП.

6.5.1.4. Четвертый шаг — идентификация ЦП NX3030 и проверка дисплея ЦП

Четвертый шаг состоит в идентификации полукластера как PLCA или PLCB. Это делается через меню Online/Redundancy Configuration:

Далее поле со списком «Идентификация ПЛК» позволяет выбрать один из трех следующих вариантов:

- ПЛК А
- ПЛК Б
- Без резервирования

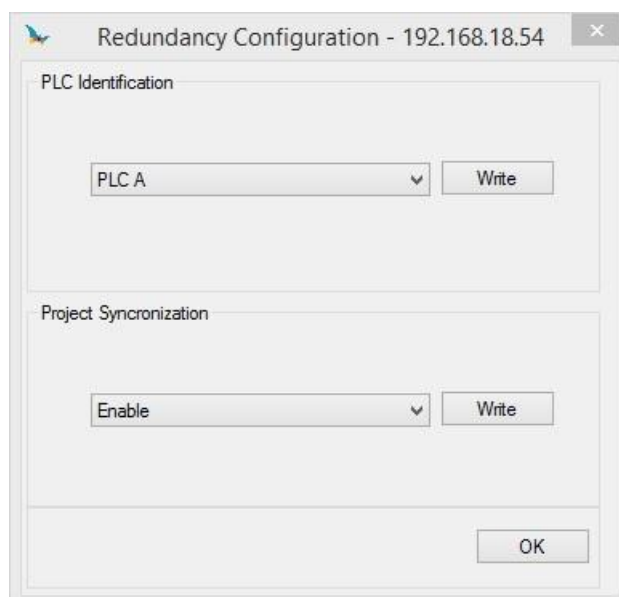


Рисунок 187: Идентификация ПЛК

В случае резервного ЦП пользователь должен выбрать PLCA или PLCB. После выбора нужной опции необходимо нажать кнопку «Запись», соответствующую этому выпадающему списку. MasterTool возвращает сообщение, предупреждающее о том, что ЦП будет перезапущен, и ожидает подтверждения действия пользователем. Затем появится сообщение об успешном или неудачном выполнении команды. В случае успеха процессор будет перезапущен.

ВНИМАНИЕ

ЦП NX3030 не может находиться в режиме Run во время выполнения этой команды. Перед выполнением этой команды пользователь должен перевести ЦП в режим остановки. Если ЦП находится в режиме Run, команда не выполняется, и MasterTool предупреждает, что команда не выполнена.

Сразу после успешного выполнения команды идентификации можно заметить, что выбранная идентификация появляется в диагностике резервирования на графическом дисплее ЦП NX3030.

Идентификация ЦП также доступна во внутренней диагностике (DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.ePLC_ID). Эта диагностика обновляется из энергонезависимой памяти в каждом цикле MainTask, поэтому ЦП необходимо вернуться в режим Run для ее обновления. Коды, возвращаемые диагностикой, и соответствующие ограничения перечислены ниже:

Без резервирования: 0

ПЛКА: 2

ПЛКБ: 3

Идентификация ЦП не является частью избыточного проекта, разработанного с помощью MasterTool. Такая идентификация есть только в области энергонезависимой памяти ЦП, которую можно модифицировать с помощью MasterTool.

ПРЕДУПРЕЖДЕНИЕ

Резервирование не работает должным образом, если один из ЦП не идентифицирован как PLCA, а другой PLCB, когда может произойти прерывание управления процессом. В случае необходимости замены одного ЦП NX3030 (например, после повреждения), новый ЦП должен быть предварительно идентифицирован с тем же идентификатором, что и поврежденный. Дисплей ЦП должен использоваться для проверки правильности идентификации обоих ЦП.

7. ОБСЛУЖИВАНИЕ

6.5.1.5. Пятый шаг — загрузка избыточного проекта

Этот шаг описывает резервную загрузку проекта в CPU. Этот проект должен быть подготовлен в соответствии с разделом «Программирование резервного ЦП».

Простой проект (базовый) можно подготовить, следуя как минимум следующим подразделам, представленным в этом разделе:

Мастер создания нового избыточного проекта

Конфигурация портов Ethernet в ЦП NX3030 (NET 1 и NET 2)

Очевидно, что также можно построить полный проект и только потом загрузить его в PLCA и PLCB, например, в случае, если аппаратное обеспечение этих процессоров недоступно во время разработки проекта в MasterTool.

Первая загрузка избыточного проекта в ЦП, ранее идентифицированного как PLCA или PLCB, по-прежнему должна быть выполнена с использованием IP-адреса, обнаруженного на первом шаге, и выбора третьего шага этой процедуры. Загрузка проекта осуществляется через меню Online/Login.

ВНИМАНИЕ

Внутри разработанного проекта с помощью MasterTool и загруженного в ЦП на этом этапе были определены новые IP-адреса для интерфейса NET 1 в PLCA и PLCB (IP-адрес PLC A и IP-адрес PLC B), а также IP-адрес для Интерфейс NET 1 в активном ЦП (IP-адрес активен) — см. Конфигурация портов Ethernet в разделе ЦП NX3030 (NET 1 и NET 2).

Следовательно, после этой первой загрузки IP-адрес, обнаруженный на первом этапе этой процедуры, обычно уже недействителен. Это изменение IP-адреса в NET 1 вызывает потерю соединения между MasterTool и ЦП, что отображается на экране. Дополнительные сведения о переподключении MasterTool см. в разделе «Подключение MasterTool к ЦП NX3030 из резервного ПЛК».

6.5.2. Соединение MasterTool с ЦП NX3030 от резервного ПЛК

После выполнения процедуры, описанной в разделе «Начальная загрузка резервного проекта» в обоих ПЛК (PLCA и PLCB), подключение MasterTool через интерфейс NET 1 от ЦП NX3030 может быть выполнено по одному из следующих адресов:

IP-адрес PLC A: адрес NET 1 исключительно для PLCA IP-адрес PLC B: адрес NET 1 исключительно для PLCB

Независимо от состояния ПЛК, MasterTool может подключаться к нему только с использованием эксклюзивного адреса ПЛК, настроенного в IP-адресе ПЛК X. Но если ПЛК находится в активном состоянии, все другие службы могут подключаться к нему либо по IP-адресу ПЛК X, либо по активному IP-адресу.

Чтобы подключиться к определенному ПЛК, сначала необходимо дважды щелкнуть устройство (NX3030) в дереве устройств, перейти на вкладку «Настройки связи», щелкнуть шлюз и нажать кнопку «Сканировать сеть», чтобы просмотреть список всех ПЛК. обнаружены MasterTool в сети.

В этом списке можно найти следующие стандартные обозначения, если имя ПЛК в сети не было ранее изменено пользователем:

NX3030_<IP-адрес>_PLCA: идентификация, связанная с PLCA. В этом случае поле <IP-адрес> должно совпадать с IP-адресом ПЛК A, настроенным в проекте.

NX3030_<IP-адрес>_PLCB: идентификация, относящаяся к PLCB. В этом случае поле <IP-адрес> должно совпадать с IP-адресом ПЛК B, настроенным в проекте.

Затем необходимо выбрать из списка ПЛК, к которому MasterTool должен подключиться, и нажать кнопку «Установить активный путь». Затем, при выполнении команды из меню Online/Login, MasterTool подключается к этому ПЛК.

ВНИМАНИЕ

MasterTool может одновременно подключаться только к одному ПЛК. Для подключения к нескольким ПЛК в MasterTool должно быть открыто несколько экземпляров, при этом необходимо позаботиться об открытии правильного проекта в каждом экземпляре.

6.5.3. Загрузка модификации в избыточном проекте

После того, как для обоих ПЛК (PLCA и PLCB) из резервированной пары уже загружена исходная программа, как описано в разделе «Первоначальная загрузка резервного проекта», можно загрузить последовательные изменения в проекте, когда такие изменения необходимы.

Соединение MasterTool с ПЛК, ответственными за загрузку модификаций, должно быть выполнено, как описано в разделе Соединение MasterTool с ЦП NX3030 из резервированного ПЛК. В этом разделе объясняется, как можно подключиться к конкретному ПЛК (PLCA или PLCB), к активному ПЛК или к неактивному ПЛК.

Обычно модификации должны быть загружены в активный ПЛК, а затем автоматически синхронизированы с неактивным ПЛК.

Активный ПЛК через каналы синхронизации NETA/NETB. Поэтому MasterTool обычно должен использовать эксклюзивный IP-адрес активного ПЛК (IP-адрес PLC X) для подключения к каналу NET 1 от ЦП NX3030 в активном ПЛК. Чтобы проверить, какой ПЛК находится в активном состоянии, можно выполнить тот же шаг, который описан в разделе «Первоначальная загрузка резервного проекта — четвертый шаг — идентификация ЦП NX3030 и проверка дисплея ЦП».

ВНИМАНИЕ

Загрузка проекта в неактивный ПЛК обычно бесполезна, так как автоматическая синхронизация проекта (активного в неактивный ПЛК) отменит эффект этой загрузки. Однако существуют особые ситуации, когда синхронизацию проекта необходимо временно отключить, поскольку это возможно и полезно для загрузки другого проекта в неактивный ПЛК. Эти особые ситуации обсуждаются в разделе «Изучение избыточности для автономной загрузки модификаций без прерывания процесса управления».

6.5.4. Загрузка офлайн и онлайн модификаций

Модификации проекта могут быть загружены офлайн или онлайн.

Автономные загрузки требуют остановки ПЛК, где предполагается выполнение загрузки. С другой стороны, онлайн-загрузки позволяют ПЛК продолжать выполнение своего приложения во время загрузки модификации.

Некоторые типы модификаций требуют автономной загрузки и не могут быть выполнены онлайн в ПЛК, к которому подключен MasterTool. В этом случае есть два варианта:

Чтобы прервать управление процессом, выполните процедуру, описанную в Оффлайн-загрузке модификаций с

Секция прерывания процесса управления

Используйте резервирование ПЛК и сети PROFIBUS, чтобы избежать прерывания управления процессом, даже при необходимости выполнения автономных загрузок в каждом полукластере (PLCA или PLCB). Процедура для достижения этой цели описана в разделе «Изучение избыточности для автономной загрузки модификаций без прерывания процесса управления».

6.5.4.1. Модификации, требующие автономной загрузки и прерывания управления процессом

7. ОБСЛУЖИВАНИЕ

Следующие изменения в проекте сделают невозможной загрузку в резервированную систему без прерывания управления процессом:

Изменения в резервных областях памяти (изменения в параметрах резервирования из модуля NX4010)

ВНИМАНИЕ

Невозможно будет изменить размер избыточных областей памяти без прерывания управления процессом. Таким образом, эти области должны быть тщательно спланированы и предварительно настроены.

6.5.4.2. Модификации, требующие автономной загрузки

Следующие модификации требуют автономных загрузок в ПЛК, к которому подключен MasterTool:

Чтобы добавить или удалить устройства из дерева устройств, например:

- Модули в основной стойке (ведущие устройства NX5001 PROFIBUS, интерфейсы NX5000 Ethernet и т. д.)
- Пульты в сетях PROFIBUS
- Модули ввода/вывода в удаленных устройствах в сетях PROFIBUS
- экземпляры MODBUS

Чтобы изменить параметры внутри устройств из дерева устройств, например:

- IP-адреса и другие параметры интерфейсов Ethernet
- Ведущие параметры PROFIBUS
- Параметры удаленных устройств PROFIBUS
- Параметры модулей ввода/вывода в удаленных устройствах PROFIBUS.

Чтобы изменить период задачи

Обновление проекта в связи с обновлением программатора MasterTool IEC XE

6.5.4.3. Модификации, позволяющие загружать онлайн

Априори, модификации, не упомянутые в разделах «Модификации, требующие автономной загрузки» и «Прерывание управления технологическим процессом» и «Модификации, требующие автономной загрузки», допускают онлайн-загрузку.

Даже в этом случае модификации, позволяющие загружать онлайн в ПЛК, к которому подключен MasterTool, перечислены ниже. Эти модификации действительны для переменных, программных модулей и GVL, избыточных или нет:

Чтобы добавить POU из типа программы, если эти POU не нужно связывать с какой-либо задачей

Чтобы удалить POU из типа программы, если эти POU не связаны ни с одной задачей

Чтобы добавить или удалить POU из функции или типа функционального блока

Для изменения кода любого типа POU (программы, функции или функционального блока)

Чтобы добавить или удалить символические переменные в любом типе POU (программа, функция или функциональный блок)

Чтобы добавить или удалить экземпляры функциональных блоков в POU из программы или типа функционального блока

Чтобы добавить или удалить GVL

Чтобы добавить или удалить символические переменные или экземпляры функциональных блоков в GVL

6.5.5. Онлайн-загрузка модификаций

7. ОБСЛУЖИВАНИЕ

В разделе «Автономная и онлайн-загрузка модификаций» были описаны модификации, требующие автономной загрузки, а также модификации, допускающие онлайн-загрузку.

Онлайн-изменение должно быть выполнено путем подключения MasterTool к каналу NET 1 активного ЦП с использованием его уникального IP-адреса. До версии 2.01 MasterTool IEC XE было необходимо, чтобы пользователь выбрал опцию «Создать загрузочное приложение» в онлайн-меню, после отправки приложения для отправки изменений в область энергонезависимой памяти ЦП и мог быть синхронизированы. Начиная с версии 2.01 эта операция больше не нужна. После отправки заявки операция отправки энергонезависимой памяти выполняется автоматически.

ВНИМАНИЕ

Важно помнить, что онлайн-модификации без ранее выбранной опции будут потеряны в случае перезагрузки или отключения процессора.

ВНИМАНИЕ

Онлайн-изменение в объявлении сохраняемых переменных приложения (добавление или удаление переменных) с последующим сбросом силового ПЛК перед «Созданием загрузочного приложения» приведет к повреждению сохраняемой памяти, поскольку значение сохраняемых переменных, которые были сохранены, не сопоставить извлеченные переменные приложения в восстановленной памяти.

Когда неактивный ПЛК понимает, что его проект отличается от проекта активного ПЛК, он выполняет следующие действия:

Согласует автоматическую синхронизацию проекта с активным ПЛК.

Если он находится в состоянии Stand-by или Starting, он переключается в состояние Not-Configured и остается в нем до тех пор, пока проекты не будут снова синхронизированы. После этого автоматически возвращается в состояние ожидания

Если он находится в состоянии Not-Configured или Inactive, необходимо нажать кнопку STAND-BY на панели PX2612 или выполнить эквивалентную команду. Таким образом, после синхронизации проекта он выходит из состояния Not-Configured и может перейти в состояние Stand-by или вернуться в состояние Inactive в случае сбоя.

6.5.6. Автономная загрузка модификаций с прерыванием управления технологическим процессом

В данном разделе определена процедура для выполнения автономной загрузки, которая прерывает управление процессом. Такая ситуация допустима в конкретных типах процессов и при программной остановке процесса.

Автономная загрузка из этого типа должна выполняться при подключении MasterTool к каналу NET 1 активного ПЛК с использованием эксклюзивного IP-адреса из активного ПЛК (IP-адрес ПЛК X). Перед началом автономной загрузки в Active PLC пользователь получает два предупреждения MasterTool:

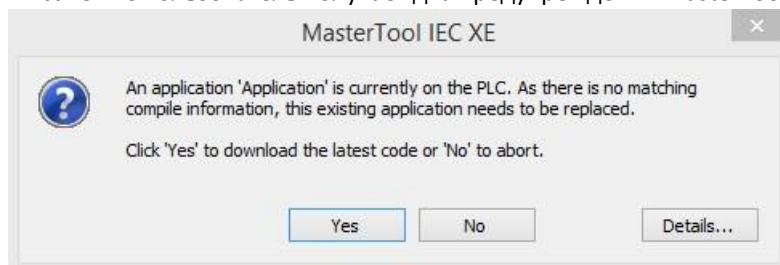


Рисунок 188: Предупреждение о загрузке в автономном режиме

7. ОБСЛУЖИВАНИЕ

При нажатии Да проект загружается. Когда выполняется автономная загрузка, управление процессом прерывается, поскольку проект отправляется в активный ПЛК, который выходит из состояния «Выполнение» и, следовательно, находится в состоянии «Не сконфигурирован».

Еще одним важным моментом является то, что если другой ПЛК находится в состоянии ожидания, его необходимо переключить в неактивное состояние, например, нажав кнопку PX2612 INACTIVE на этом ПЛК. Таким образом, предотвращается отключение этого ПЛК другим ПЛК и его передача в качестве активного.

ВНИМАНИЕ

Когда активный ПЛК выходит из режима «Выполнение» и переходит в состояние «Не настроен», если другой ПЛК был забыт в состоянии «Ожидание», он становится активным и выключает ПЛК, который только что перешел из режима «Активен» в состояние «Не настроен». Таким образом, в этом случае автономная загрузка не может быть завершена, поскольку ПЛК, подключенный к MasterTool, выключен.

Когда автономная загрузка завершится, можно перезапустить выполнение программы ПЛК, в которой было загружено приложение (снова ввести Run). Через несколько секунд этот ПЛК снова принимает активное состояние.

После того, как этот ПЛК снова перейдет в активное состояние, другой ПЛК может выйти из неактивного состояния, т.е. нажав PX2612

Кнопка STAND-BY на нем. Это приводит к переходу этого ПЛК в состояние «Не сконфигурировано». Этот ПЛК остается в состоянии NotConfigured до завершения автоматической синхронизации проекта. Затем он переходит в начальное состояние, а затем возвращается в состояние ожидания.

6.5.7. Предыдущее планирование автономных модификаций без прерывания процесса управления

В следующем подразделе – Предварительное планирование «горячих» модификаций в резервированных сетях PROFIBUS – описывается очень важная процедура, позволяющая загружать автономные модификации без прерывания управления процессом. Несмотря на то, что эта процедура не применяется к любым модификациям, требующим автономной загрузки, она применяется к наиболее часто используемым модификациям.

Однако, чтобы применить эту процедуру, проекты должны быть разработаны с предварительным планированием, особенно для модификаций, влияющих на сеть PROFIBUS. В следующих подразделах описывается такое предварительное планирование модификаций, влияющих на сеть PROFIBUS, а также другие модификации.

6.5.7.1. Предыдущее Планирование «горячих» модификаций в резервированных сетях PROFIBUS

Среди модификаций, влияющих на сеть PROFIBUS и требующих автономной загрузки, следующие поддерживаются процедурой, позволяющей выполнять автономные загрузки без прерывания управления процессом, если сеть PROFIBUS резервирована:

Вставьте новую сеть PROFIBUS

Вставьте новый пульт дистанционного управления Ponto Series

Вставьте новый модуль ввода-вывода в удаленное устройство серии Ponto.

Изменение параметров в пультах Ponto Series или в модулях ввода-вывода в пультах Ponto Series.

ВНИМАНИЕ

Можно вставить нерезервированные удаленные устройства в резервированную сеть PROFIBUS, используя модуль AL-2433 (ProfiSwitch), как показано в примере на рис. выполнен.

Далее описываются этапы планирования, которые должны начаться при создании новой резервированной сети PROFIBUS.

6.5.7.1.1. Шаг 1. Планирование будущего расширения удаленных устройств, вставленных в сеть PROFIBUS. Начальная версия.

7. ОБСЛУЖИВАНИЕ

Сначала необходимо составить список модулей ввода/вывода, которые составляют каждый резервный удаленный PROFIBUS из серии Ponto в начальной версии сети PROFIBUS. В списке должна быть указана позиция, в которой каждый модуль ввода-вывода вставлен в удаленную стойку.

Затем необходимо спланировать будущее расширение каждого пульта. Для этого необходимо составить дополнительный список, состоящий из модулей ввода/вывода, которые могут быть вставлены в будущем. В нем должно быть указано положение, в котором каждый модуль ввода-вывода может быть вставлен в удаленную стойку.

ВНИМАНИЕ

При физической конструкции этих пультов (электрических панелей) настоятельно рекомендуется вставлять совместимые основания с будущими модулями ввода-вывода в соответствующие позиции. Таким образом, когда необходимо вставить модуль ввода-вывода в этот пульт, нет необходимости выключать пульт, чтобы вставить базу. В случае, если эта деталь не соблюдена, необходимо будет выключить конкретный пульт, так как невозможна горячая вставка базы в пульт. Можно заметить, что дистанционная остановка в некоторых случаях может быть допустимой, но не всегда.

Исходные базы модулей ввода-вывода должны быть вставлены в первые позиции удаленных стоек, а будущие модули ввода-вывода — в последние позиции удаленных стоек.

ВНИМАНИЕ

При составлении этого списка следует учитывать ограничения резервных удаленных устройств серии Ponto, таких как Руководство по использованию головной части PROFIBUS PO5063V1 и Резервной головки PROFIBUS PO5063V5, а также Руководство по использованию головной части PROFIBUS PO5064 и Резервной головки PROFIBUS PO5065. Существуют ограничения, касающиеся количества модулей на пульт, количества байтов на пульт, потребляемого тока на источник питания и т. д. Эти ограничения автоматически проверяются ProPonto. Для получения дополнительной информации см. Руководство по эксплуатации MT6000 MasterTool ProPonto — MU299040.

6.5.7.1.2. Шаг 2. Вставьте исходную версию сети PROFIBUS с резервированием в проект

Чтобы вставить в проект резервную начальную версию сети PROFIBUS, сначала в стойку должны быть вставлены два резервных модуля NX5001, либо использовать уже вставленные мастером резервирования.

Далее каждый пульт должен быть вставлен в дерево устройств под этими двумя NX5001, а также модули ввода/вывода под каждым пультом.

Что касается вставленных модулей ввода-вывода, существует две категории, которые следует рассматривать по-разному:

Те, которые являются частью начальной версии сети PROFIBUS и будут установлены немедленно Те, которые будут использоваться для будущего расширения

В случае тех, которые являются частью начальной версии сети PROFIBUS, сам модуль должен быть вставлен в дерево устройств, в запланированную позицию удаленного корреспондента.

В случае тех, которые будут использоваться для будущего расширения, в запланированную соответствующую позицию необходимо вставить виртуальный модуль. Виртуальный модуль, соответствующий реальному модулю, должен выделять такое же количество байтов ввода-вывода, как и этот реальный модуль. Вставка виртуального модуля вместо реального модуля позволяет избежать диагностики отсутствия реального модуля.

В следующей таблице показаны реальные модули и соответствующие им виртуальные модули:

Реальный модуль	Корреспондентский виртуальный модуль
PO1000	PO9999 — 2 байта ввода
PO1001	PO9999 — 2 байта ввода

PO1002	PO9999 — 2 байта ввода
PO1003	PO9999 — 2 байта ввода
PO2020	PO9999 — 2 байта Выход
PO2022	PO9999 — 2 байта Выход

Таблица 207: Виртуальные модули, соответствующие реальным модулям

6.5.7.1.3. Шаг 3 – Выделите области переменных %I и %Q для сети PROFIBUS с учетом будущего удаленного расширения

Так как удаленные устройства и модули ввода-вывода NX5001 вставлялись в дерево устройств на предыдущем шаге, переменные %I и %Q размещались в трех разных областях:

Область переменных %I для входных данных

Область переменных %Q для выходов

Область переменных %Q для диагностики

MasterTool выполняет выделение каждой из этих трех переменных областей непрерывным образом, без промежутков между ними.

Начальный и конечный адреса каждой из этих трех областей должны быть запланированы с учетом изначально установленных удаленных устройств в сети (см. шаги 1 и 2), а также удаленных устройств, которые могут быть вставлены в будущем в эту же сеть PROFIBUS.

При определении начального адреса каждой области важно зарезервировать расширение для устройства, которое выделяет адреса непосредственно перед началом этой области. С другой стороны, при определении конечного адреса каждой области важно зарезервировать расширение для этой сети PROFIBUS.

Далее показан пример такого планирования для области переменных %I для входов:

Сеть PROFIBUS 1:

- %IB0 ... %IB499 (адреса, выделенные для уже установленных пультов)
- %IB500 ... %IB999 (адреса, назначенные будущим удаленным устройствам) Сеть PROFIBUS 2:
- %IB1000 ... %IB1499 (адреса, выделенные для уже установленных удаленных устройств)
- %IB1500 ... %IB1999 (адреса, выделенные будущим удаленным устройствам) Сервер Modbus TCP:
- %IB2000 ... %IB2999 (адреса, выделенные для текущего сопоставления)
- %IB3000 ... %IB3999 (адреса, выделенные для будущего сопоставления)

Для двух других областей (выходной %Q и диагностический %Q) могут быть выполнены аналогичные примеры.

Можно предсказать размер первоначально выделенных и будущих областей расширения, используя следующую таблицу, в которой указано количество байтов, выделенных для 3 областей для каждого модуля:

Модуль	Входы %I (байты)	Выходы %Q (байты)	Диагностика %Q (байты)
NX5001	4	2	86
PO5063V5	0	0	25
PO5065	0	0	25
PO9100 (one each remote)	2	2	10
PO1000	2	0	10
PO2020	0	2	10
PO9999 – 2 bytes Output	0	2	10
PO9999 – 2 bytes Input	2	0	10

Таблица 208: Назначение переменных %I и %Q для сетевых модулей PROFIBUS

7. ОБСЛУЖИВАНИЕ

Примечание:

Распределение переменных: Дополнительную информацию о размере и типе памяти, выделенной для каждого модуля, можно найти в Руководстве по использованию ведущего устройства PROFIBUS-DP NX5001.

После выполнения планирования для 3 областей (начальный и конечный адрес каждой области) начальные адреса должны быть вставлены в проект, начатый на шаге 2.

Сначала необходимо изменить параметр «%Q Начальный адрес области диагностики модуля» в первом модуле NX5001, как показано в таблице на следующем рисунке. Запланированный начальный адрес должен использоваться для диагностических переменных %Q.

Во-вторых, необходимо найти первый сетевой модуль ввода/вывода, начиная с NX5001, который выделяет %I переменных для входов. При его нахождении необходимо изменить соответствующий параметр «Адрес».

В-третьих, необходимо найти первый сетевой модуль ввода-вывода, начиная с NX5001, который выделяет переменные %Q для выходов. При его нахождении необходимо изменить соответствующий параметр «Адрес».

ВНИМАНИЕ

На данный момент рекомендуется проверить выделенный диапазон адресов для 3 переменных областей, проверяя, находятся ли окончательные адреса каждой области в запланированном диапазоне, и есть ли хорошая свободная область для расширения для вставки новых удаленных устройств в будущем.

6.5.7.2. Предыдущее планирование других горячих модификаций

Есть и другие горячие модификации, которые хоть и не затрагивают сеть PROFIBUS, но тоже требуют офлайн-загрузки. Далее представлены некоторые примеры модификаций этого типа, поддерживаемые процедурой, которые позволяют выполнять автономную загрузку модификаций без прерывания управления процессом:

Установка модулей NX5000 (Ethernet)

Вставка драйвера ввода-вывода Ethernet или последовательной связи

Драйвер ввода-вывода Ethernet или последовательной связи, новая вставка сопоставления

С другой стороны, предыдущие примеры модификаций подразумевают прямое представление распределения переменных %I и %Q для диагностики, входов и выходов, аналогично рассмотренному на шаге 3 из предыдущего планирования горячих модификаций, влияющих на сеть PROFIBUS (см. Шаг 3 – Выделить области переменных %I и %Q для сети PROFIBUS с учетом будущего удаленного расширения).

Таким образом, при вставке модуля NX5000, Ethernet- или последовательного драйвера ввода-вывода для вставляемого устройства необходимо запланировать выделение 3 следующих областей:

Область переменных %I для входных данных

Область переменных %Q для выходов

Область переменных %Q для диагностики

Шаг 3. Выделение областей переменных %I и %Q для сети PROFIBUS с учетом будущего удаленного расширения. В разделе показан пример группового распределения этих областей, включая сети PROFIBUS и драйвер ввода-вывода (сервер MODBUS TCP).

6.5.7.3. Несовместимость приложений

Если области, которые будут использоваться в будущем, не будут спланированы должным образом, избыточные области памяти, возможно, придется изменить, что приведет к несовместимости между приложениями. Это приведет к тому, что только один ПЛК останется в активном состоянии, а другой ПЛК останется неактивным, без возможности синхронизации избыточных данных или приложений между двумя ПЛК.

Об этой несовместимости сообщает диагностика резервирования по адресу:

DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bПриложение несовместимо.

Эта диагностика активна, когда выполняется загрузка нового приложения в один из ПЛК, обычно неактивный, с одним из следующих изменений:

7. ОБСЛУЖИВАНИЕ

Изменения резервных областей памяти, настроенных в параметрах модуля NX4010

Изменения (создание или удаление) в символических избыточных переменных, объявленных в избыточных POU или избыточных GVL

Важно подчеркнуть, что при внесении изменений в символические избыточные переменные проблема несовместимости возникает только в том случае, если в один из ПЛК выполняется загрузка нового приложения. В случае, если изменения в символических избыточных переменных и все другие изменения, сделанные в проекте, входят в группу Модификаций, допускающих онлайн-загрузку, можно выполнить онлайн-загрузку модификаций без создания несовместимости приложений между ПЛК. .

6.5.7.4. Обновление проекта в связи с обновлением MasterTool IEC XE

Инструмент программирования MasterTool IEC XE постоянно совершенствуется, улучшаются его функции и добавляются новые. Когда необходимо обновить инструмент в системе с резервированием, используемый проект также необходимо обновить. Это обновление выполняется через меню Project/Project Update, доступное в инструменте. После обновления проекта можно выполнить автономную загрузку без прерывания управления процессом.

6.5.7.4.1. Обновление проекта с версий до 2.00 до версии 2.00 или выше

Среди изменений версии MasterTool IEC XE есть особый случай, который необходимо планировать более тщательно, чтобы избежать остановки процесса. Обновление проекта, созданного с помощью MasterTool IEC XE версии до 2.00, до версии 2.00 или выше вызывает реконфигурацию в области, выделенной для объекта Persistent Variables. Эта реконфигурация была осуществлена с целью оптимизации распределения этой области. Однако, если этот объект присутствует и помечен как избыточный в проекте, эта реконфигурация не позволит синхронизировать данные между двумя проектами, всегда переводя один из полукластеров в неактивное состояние.

Таким образом, если такая ситуация возникает, программное обеспечение ЦП NX3030 может обнаружить это и остановить синхронизацию данных объекта Persistent Variables до тех пор, пока проекты двух полукластеров не станут одинаковыми и, следовательно, не будут использовать проект с обновленным MasterTool IEC XE версия. Эта ситуация не остановит процесс, но если правильная последовательность обновления не будет соблюдена, данные объекта Persistent Variables могут быть перезапущены.

В этом случае необходимо выполнить приведенную ниже последовательность автономной загрузки:

Измените проект Half-Cluster in Active state, сняв пометку с объекта PersistentVars внутри объекта Redundancy Configuration. Эта загрузка должна быть выполнена как онлайн-изменение, и для этого необходимо внести другое изменение в проект, например. объявление новой переменной внутри NonSkippedPrg POU

После изменения Online необходимо запустить команду Create Boot Application в онлайн-режиме, когда ПЛК находится в активном состоянии. Это необходимо для того, чтобы приложение синхронизировалось с полукластером, перешедшим в состояние Not-Configured после загрузки

Обновите проект с версии ниже 2.00 до версии 2.00 или выше через меню Project/Project Update в MasterTool IEC XE.

Отключите синхронизацию проекта через меню Online/Redundancy Configuration.

Загрузите обновленный проект в полукластер, находящийся в состоянии ожидания. Появится сообщение, указывающее на реорганизацию области памяти объекта PersistentVars. Процедура должна продолжаться, и к концу загрузки проекта полукластер останется в состоянии STOP с состоянием резервирования как Not-Configured.

Переведите процессор в режим RUN. Half-Cluster перейдет в состояние Starting, а затем в Stand-by. Half-Cluster синхронизирует свои данные с данными, находящимися в активном состоянии.

Данные из объекта PersistentVars необходимо скопировать из активного полукластера в резервный вручную или использовать ресурс квитанции.

Переведите активный полукластер в режим ожидания. С этим действием другой полукластер перейдет в активный режим.

Включите синхронизацию проекта в меню Online/Redundancy Configuration. После этого полукластер в состоянии Standby перейдет в состояние Not-Configured и получит проект от полукластера в активном состоянии. К концу этого процесса состояние Half-Cluster перейдет в Starting, а затем обратно в Stand-by.

Измените проект полукластера в активном состоянии, отметив объект PersistentVars внутри объекта Redundancy Configuration. Эта загрузка должна быть выполнена как онлайн-изменение, и для этого

7. ОБСЛУЖИВАНИЕ

необходимо внести другое изменение в проект, например. удаление переменной, объявленной в начале этого процесса

После этого Полукластер, находившийся в Stand-by, перейдет в состояние Not-Configured и получит Проект от Полукластер в активном состоянии. К концу этого процесса состояние Half-Cluster изменится на Starting, а затем обратно на ПОДДЕРЖИВАЮЩИЙ

6.5.8. Изучение избыточности для автономной загрузки модификаций без прерывания управления технологическим процессом

В разделе «Автономная и онлайн-загрузка модификаций» сообщалось, что некоторые модификации требуют автономной загрузки в ПЛК, куда такие модификации должны быть загружены. В этих случаях у пользователя есть возможность прервать управление технологическим процессом в соответствии с процедурой, определенной в разделе Автономная загрузка модификаций с прерыванием управления технологическим процессом. Поэтому обычно необходимо заранее запрограммировать остановку процесса, что не всегда возможно, когда требуется срочная модификация.

Благодаря резервированию ПЛК и, в некоторых случаях, резервированию сети PROFIBUS можно выполнять автономные загрузки без прерывания управления процессом для большинства обычно необходимых модификаций. Для достижения этой цели необходимо тщательно следовать процедуре, шаги которой описаны в следующих подразделах.

6.5.8.1. Шаг 1. Проверка основных требований к участию

Чтобы автономная загрузка без прерывания управления процессом была возможной, должны быть соблюдены следующие основные требования:

Исходный проект должен быть создан в соответствии с рекомендациями раздела «Предыдущее планирование автономных модификаций без прерывания управления процессом».

ПЛК должен быть резервным

Если модификация затрагивает сеть PROFIBUS, необходимо, чтобы эта сеть была резервной. Такими модификациями могут быть:

- Вставка новых пультов
- Вставка модулей ввода/вывода в существующие пульты, в ранее зарезервированные позиции для соответствующих виртуальных модулей. Чтобы пульт не нужно было выключать, необходима база, совместимая с новым модулем ввода-вывода, в отведенном для него положении.
- Изменение параметров в удаленных или существующих модулях ввода-вывода.

Оба проекта ПЛК должны быть уравновешены, а службы резервной синхронизации данных и резервного списка форсировки должны работать должным образом без диагностики сбоя. Можно сказать, что эти условия выполняются, когда один ПЛК находится в активном состоянии, а другой — в резервном. Если неактивный ПЛК не находится в состоянии ожидания, можно наблюдать следующую диагностику:

- `DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bRedDataSync = TRUE`, указывает на успешное выполнение службы резервной синхронизации данных.
- `DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bRedForceSync = TRUE`, указывает на успешное выполнение службы резервного списка форсировки.
- `DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.dwApplicationCRC = DG_NX4010.tRedundancy.RedDgnRem.dwApplicationCRC`, указывает, что оба проекта ПЛК равны.

6.5.8.2. Шаг 2. Не загружайте в группу модификации, которые можно загрузить в Интернете.

Модификации, которые можно загрузить в режиме онлайн, нельзя загружать вместе с модификациями, которые необходимо загружать в режиме офлайн, без прерывания управления технологическим процессом. Когда эти два вида модификаций необходимы, их всегда нужно загружать отдельно.

Чтобы текущая процедура была успешной, абсолютно необходимо, чтобы выполненные модификации не вызвали никаких изменений в структуре резервных переменных, которыми обмениваются активный и неактивный ПЛК, посредством сервисов резервной синхронизации данных и резервного списка форсировки. Эти две службы должны продолжать работать должным образом даже при наличии временных различий между ПЛК. Модификации, которые должны быть загружены в автономном режиме и поддерживаются этой процедурой, не влияют на структуру избыточных переменных.

Однако некоторые модификации, которые можно загрузить онлайн, могут изменить структуру избыточных переменных, например:

Вставка символических переменных (избыточных или нет) в существующий POU или GVL или в новый POU или GVL

Удаление символьных переменных (избыточных или нет) в POU или в существующем GVL. Удаление POU или GVL может также включать удаление символьных переменных.

Изменение размера/структуры символьных переменных (избыточных или нет) в существующем POU или GVL

7. ОБСЛУЖИВАНИЕ

6.5.8.3. Шаг 3 – Предыдущая резервная копия проекта

Перед редактированием модификаций, которые необходимо загрузить в автономном режиме без прерывания управления процессом, в целях безопасности необходимо запустить резервную копию предыдущей версии проекта. Возможно, потребуется переустановить предыдущую версию, если при выполнении этой процедуры произойдет ошибка.

ВНИМАНИЕ

Рекомендации по резервному копированию для всех загруженных версий в ПЛК

могут не соблюдаться только в этой конкретной процедуре. Это должно быть обычной практикой.

6.5.8.4. Шаг 4 – Заботы при редактировании загруженных в автономном режиме модификаций

Автономно загруженные модификации через эту процедуру, как правило, следующие:

- Вставка новых устройств в дерево устройств
- Изменение свойства или параметра в устройствах, существующих в дереве устройств

Обычно это следующие устройства:

- Модули, такие как ведущий PROFIBUS (NX5001) или модули Ethernet (NX5000)
- Пульты PROFIBUS серии Ponto
- Модули ввода/вывода внутри пультов PROFIBUS серии Ponto
- Драйверы ввода/вывода связи MODBUS
- Сопоставление коммуникационных драйверов MODBUS

При редактировании этих модификаций проекта необходимо соблюдать следующие меры предосторожности:

- Если устройство существовало в предыдущей версии проекта и продолжает существовать в измененной версии, то выделенные для него переменные %I и %Q должны остаться прежними (командные, диагностические, входы и выходы). Необходимо следить за тем, чтобы вставленные модификации не изменяли такие распределения.
- Если устройство было вставлено в модифицированную версию проекта, выделенные для него переменные %I и %Q не должны размещаться в предыдущей версии проекта (командная, диагностическая, входы и выходы)

6.5.8.5. Шаг 5 – Отключение синхронизации неактивного проекта ПЛК

В процедурах, описанных в разделах «Онлайн-загрузка модификаций» и «Оффлайн-загрузка модификаций с прерыванием управления процессом», проект автоматически синхронизируется с активного ПЛК на неактивный ПЛК.

Однако при выполнении процедуры автономной загрузки без прерывания управления процессом синхронизация проекта должна быть временно отключена. Отключение синхронизации проекта объясняется в разделе Отключение синхронизации проекта и должно выполняться в неактивном ПЛК.

6.5.8.6. Шаг 6 – Выполнение физических модификаций

В этот момент могут быть выполнены физические модификации, такие как:

7. ОБСЛУЖИВАНИЕ

Установите новый модуль NX5000. Это можно сделать путем горячей установки модуля в каждую полукластерную стойку с последующим подключением к сети Ethernet.

Установите новую резервную сеть PROFIBUS. NX5001 можно вставлять в каждую полукластерную стойку в горячем состоянии. Затем к ним можно подключить резервную сеть PROFIBUS.

Установите новый резервный пульт Ponto Series. В этом случае удаленная головка должна быть установлена одновременно, т.е. сначала в сети В, а затем в сети А:

- Для установки головки в сеть В может потребоваться размыкание кабеля или контактов, что нарушит связь с другими головками, уже установленными в сети В. Перед этим все работающие активные головки должны быть помещены в сеть А и оперативный резерв в сети В
- Для установки головки в сеть А может потребоваться размыкание кабеля или контактов, что нарушит связь с другими головками, уже установленными в сети А. Перед этим все работающие активные головки должны быть помещены в сеть В и оперативные резервные головки в сети А. Установите модуль ввода-вывода в базу, ранее зарезервированную для него, в существующую удаленную

6.5.8.7. Шаг 7. Загрузите автономные модификации в неактивный ПЛК.

Сначала MasterTool должен быть подключен к неактивному ПЛК (см. Соединение MasterTool с ЦП NX3030 в разделе «Резервный ПЛК»).

Далее необходимо скачать оффлайн модификации. При этом приложение Non-Active PLC автоматически прерывается (выходит из режима Run).

6.5.8.8. Шаг 8. Установите неактивный ПЛК обратно в рабочий режим, чтобы вернуться в состояние ожидания.

Когда автономная загрузка завершена, неактивный ПЛК может вернуться в рабочий режим.

Через несколько секунд неактивный ПЛК должен перейти в состояние ожидания.

Если ПЛК не переходит в состояние ожидания, это может быть вызвано следующими проблемами:

Выполненные модификации изменили структуру избыточных переменных, что препятствует корректному выполнению службы избыточной синхронизации данных. Это можно проверить с помощью диагностики DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag .bRedDataSync (0 = сбой) в неактивном ПЛК. В этом случае изменения необходимо отменить, восстановив предыдущую резервную копию проекта и перезапустив эту процедуру.

Другие проблемы могут в конечном итоге помешать переходу в состояние ожидания, даже если это неожиданно. В этом случае необходимо соблюдать журнал диагностики и резервирования.

В случае, если ПЛК перешел в состояние ожидания, рекомендуется проверить, отличаются ли проекты между активным и неактивным ПЛК. Это можно сделать, сравнив диагностику DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag .dwApplicationCRC и DG_NX4010.tRedundancy.RedDgnRem.dwApplicationCRC в неактивном ПЛК (CRC должны быть разными).

Если оба проекта одинаковы в ПЛК, возможно, отключение синхронизации проекта (шаг 5) не выполняется должным образом. Это можно проверить с помощью диагностики

7. ОБСЛУЖИВАНИЕ

DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag .bProjectSyncDisable, которая должна быть истинной в неактивном ПЛК. Если это не так, процедура должна быть возвращена к шагу 5.

6.5.8.9. Шаг 9. Выполните переключение между активным и резервным ПЛК.

Переключение между ПЛК должно быть выполнено, например, нажав кнопку STAND-BY на активном ПЛК. Резервный ПЛК, у которого есть новый проект с изменениями, становится активным. Активный ПЛК со старым проектом становится резервным.

6.5.8.10. Шаг 10 – Включение синхронизации проектов в активном ПЛК

На шаге 5 синхронизация проекта была отключена в неактивном ПЛК. Можно заметить, что этот ПЛК теперь находится в активном состоянии.

На этом этапе необходимо снова включить синхронизацию проекта в этом ПЛК. Экран и используемая для него методология описаны в разделе Отключение синхронизации проекта. Но на этот раз нам нужно выбрать опцию «Включить» в поле со списком. MasterTool должен быть подключен к активному ПЛК (см. Соединение MasterTool с ЦП NX3030 в разделе «Резервный ПЛК»).

После включения синхронизации проекта в активном ПЛК пользователь должен проверить, была ли эта команда успешной, проверив, что DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bProjectSyncDisable = 0 в активном ПЛК.

Как только синхронизация проекта снова будет включена, ожидается следующая последовательность действий:

Неактивный ПЛК (состояние ожидания), который уже знает разницу между обоими проектами; выходит из состояния Stand-by и переходит в состояние Not-Configured

Измененный проект (новый) копируется из активного ПЛК в другой, временно в состоянии «Не настроен».

Как только проекты снова синхронизируются, ненастроенный ПЛК переходит в состояние запуска, а затем должен вернуться в состояние ожидания.

6.5.8.11. Шаг 11 – Необязательная реорганизация сетей ПЛК и PROFIBUS в активном состоянии

В конце процедуры, по причинам стандартизации или организации, пользователь может выполнить переключение для PLCA, которое считается активным, и для всех удаленных головок PROFIBUS, находящихся в активном состоянии, в сети А.

6.6. Техническое обслуживание резервирования

6.6.1. Горячая замена модулей в резервном ПЛК

В случае выхода из строя модуля одного из ПЛК (PLCA и PLCB) может потребоваться горячая замена модуля без прерывания управления процессом. Для этого необходимо выполнить следующие шаги:

Убедитесь, что полукластер, который не будет изменен, находится в активном или резервном состоянии, что позволяет ему взять на себя управление процессом. Перевести полукластер, модуль которого был изменен, в неактивное состояние с помощью панели управления резервированием PX2612 или Команды резервирования

7. ОБСЛУЖИВАНИЕ

Выполните необходимые обмены в Неактивном полукластере, как указано в разделе Конфигурация ЦП - Общие параметры - Как сделать Горячая замена

Вернуть полукластер в состояние Stand-by или Active, в зависимости от необходимости

6.6.2. Предупреждающие сообщения MasterTool

Когда в MasterTool открыт резервный проект или когда он подключен к ЦП NX3030, идентифицированному как PLCA или PLCB, могут появиться некоторые специальные предупреждающие сообщения, как описано в следующих подразделах.

6.6.2.1. Блокировка избыточной или неизбыточной загрузки проекта

MasterTool не позволяет загружать избыточный проект, если ЦП не является NX3030 и не идентифицирован как PLCA или PLCB (см. раздел «Идентификация ЦП NX3030»).

С другой стороны, MasterTool не позволяет загружать нерезервированный проект в ЦП NX3030, идентифицированный как PLCA или PLCB.

В случае попытки любого из этих незаконных действий MasterTool предупреждает соответствующим сообщением

6.6.2.2. Предупреждения перед командами, которые могут остановить активный ПЛК

Некоторые команды, такие как следующие, могут остановить ПЛК:

- Автономная загрузка после онлайн/входа
- Отладка/останов
- Отладка/новая точка останова
- Онлайн / Сброс (Тепло, Холодно, Исходное)

7. ОБСЛУЖИВАНИЕ

В случае, если MasterTool регистрируется в активном ПЛК и выполняется одна из этих команд, перед отправкой в активный ПЛК MasterTool отправляет следующее сообщение и ожидает авторизации:

«Если другой ПЛК находится в состоянии ожидания, он будет считаться активным и выключит этот ПЛК. В противном случае этого не произойдет, но автоматизированный процесс больше не будет контролироваться».

6.6.2.3. Оповещение перед входом в неактивный CP

В нормальных условиях MasterTool обычно не подключается к неактивному ПЛК. Таким образом, при попытке выполнить команду такого типа MasterTool отправляет следующее предупреждение:

«Вы входите в неактивный ПЛК, и это необычно. Вы уверены, что хотите выполнить эту команду?»

С другой стороны, бывают обстоятельства (не столь обычные), при которых необходимо войти в неактивный ПЛК, и в этих случаях пользователь должен авторизовать вход. Могут возникнуть такие обстоятельства, например:

Для первоначальных конфигураций, как описано в разделе «Первоначальная загрузка резервного проекта».

Для автономной загрузки другого проекта в неактивный ПЛК, как описано в разделе «Изучение резервирования для

Автономная загрузка Модификаций без прерывания раздела Управление технологическим процессом

Для контроля или форсирования нерезервированных переменных в неактивном ПЛК

6.6.3. Диагностика резервирования на графическом дисплее процессора NX3030

Многие диагностические данные, связанные с резервированием, можно наблюдать на дисплее ЦП NX3030.

6.6.3.1. Статус резервирования CP

Состояние резервирования ПЛК, описанное в разделе «Состояния резервирования ЦП», отображается тремя начальными символами во второй строке основного экрана, как показано в разделе «Графический дисплей». Экран дисплея отображается при инициализации и снова через несколько секунд после завершения навигации (без нажатия кнопки ЦП NX3030).

6.6.3.2. Экраны под меню ИЗБЫТОЧНОСТЬ

Есть меню ИЗБЫТОЧНОСТЬ, под которым есть несколько экранов. Описание и доступ к этим экранам доступны в разделе «Конфигурация» — «Информационное и конфигурационное меню ЦП».

6.6.4. Структура диагностики резервирования

Область диагностики модуля NX4010 сопоставляется с переменными %Q прямого представления и определяется символически через директиву AT в GVL Module_Diagnostics.

Этот раздел разделен на две части:

DG_NX4010.tGeneral: Общая диагностика работы NX4010. Они описаны в технических характеристиках модуля Redundancy Link Module — CE114900.

DG_NX4010.tRedundancy: Специфическая диагностика резервирования, описанная в этом разделе. Этот элемент разделен на другие 6 структур данных:

- RedDgnLoc: имеет резервную диагностику локального ПЛК (подключенного), например. состояние резервирования ПЛК. Этот раздел описан в разделе Диагностика резервирования.
- RedDgnRem: Копия с другого ПЛК RedDgnLoc, полученная по каналам синхронизации NETA / NETB.

Таким образом, локальный ПЛК имеет доступ к диагностике удаленного ПЛК. Этот раздел описан в разделе «Резервирование».

Диагностика

7. ОБСЛУЖИВАНИЕ

- RedCmdLoc: имеет команды резервирования, сгенерированные в этом ПЛК (локальные), например, посредством команд записи из системы SCADA или сгенерированные в ROU в этом ПЛК (ActivePrg или NonSkippedPrg). Этот раздел описан в разделе Команды резервирования.
- RedCmdRem: Копия RedCmdLoc другого ПЛК (удаленного), полученная по каналам синхронизации NETA/NETB. Этот раздел описан в разделе Команды резервирования.
- RedUsrLoc: используется, чтобы позволить пользователю обмениваться информацией между PLCA и PLCB. Этот раздел описан в разделе «Обмен пользовательской информацией между PLCA и PLCB».
- RedUsrRem: используется, чтобы позволить пользователю обмениваться информацией между PLCA и PLCB. Этот раздел описан в разделе «Обмен пользовательской информацией между PLCA и PLCB».

Важно подчеркнуть, что структуры диагностики избыточности обновляются только тогда, когда происходит успешная синхронизация новых данных. Поэтому, пока не произойдет новая синхронизация, диагностика останется с последним прочитанным значением.

Кроме того, диагностические структуры удаленного ПЛК доступны только для чтения, то есть значения, записанные в эти структуры, будут перезаписаны при следующей синхронизации. Таким образом, невозможно использовать структуру RedCmdRem для выполнения команды в удаленном ПЛК. Всегда используйте структуру RedCmdLoc для выполнения команд.

6.6.4.1. Диагностика резервирования

Диагностика резервирования может иметь несколько применений, например:

К ним можно обратиться, чтобы убедиться в наличии проблемы, которую необходимо решить.

Каждый раз, когда для них появляются изменения, такие изменения вставляются как события в журнал событий резервирования. Просматривая последовательность таких событий в истории, можно обнаружить, например, причину переключения.

На них можно сослаться в пользовательском приложении (ActivePrg или NonSkippedPrg). Например, можно проверить состояние ПЛК и, если он не активен, установить драйвер ввода-вывода MODBUS RTU master можно отключить в NonSkippedPrg.

ВНИМАНИЕ

DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync (определенный далее) должен быть протестирован, чтобы убедиться, что структура данных RedDgnRem была успешно считана из удаленного ПЛК в последнем цикле MainTask. Если это диагностическое значение равно 0 (FALSE), это означает, что структура данных RedDgnRem не была успешно прочитана из удаленного ПЛК, поэтому значения RedDgnRem могут быть недействительными или устаревшими.

Поскольку RedDgnRem является копией другого ПЛК RedDgnLoc, можно сделать вывод, что эти две структуры имеют одинаковый формат. Они разделены на другие четыре подструктуры:

- sGeneral_Diag: общая диагностика резервирования
- sNETA_Diag: диагностика канала синхронизации NETA
- sNETB_Diag: диагностика канала синхронизации NETB
- sNET_Stat: общая статистика для каналов синхронизации NETA и NETB, для подсчета ошибок и успехов в службах синхронизации
- sGeneral_DiagExt: Общая диагностика резервирования, расширенная часть (продолжение sGeneral_Diag)

Подструктура «sGeneral_Diag» имеет следующие поля для общей диагностики резервирования:

Прямая переменная		AT переменная	Описание
Переменная	Бит	DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.*	

7. ОБСЛУЖИВАНИЕ

	0	bConfigDone	TRUE — процесс настройки, выполняемый в состоянии Not-Configured, завершен.
			FALSE — процесс настройки, выполняемый в состоянии Not-Configured, еще не завершен или не был выполнен.
	1	bConfigError	TRUE — процесс настройки, выполненный в состоянии Not-Configured, завершился с ошибками. Это системная ошибка, обычно не ожидаемая. Свяжитесь со службой поддержки ALTUS, чтобы сообщить об этом. Также сообщите диагностическое значение ConfigErrorCode для поддержки ALTUS.
			FALSE — процесс настройки завершен успешно или не был выполнен.
	2	bTooManyRedAreas	TRUE — количество избыточных областей превышает максимально допустимое. Это системная ошибка, обычно не ожидаемая. Свяжитесь со службой поддержки ALTUS, чтобы сообщить об этом.
			FALSE — количество избыточных областей находится в пределах ожидаемого.

Прямая переменная		AT переменная	Описание
переменная	Бит	DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.*	
%QB(n+4)	3	bTemporaryBufferTooSmall	TRUE — промежуточная структура данных недостаточного размера. Это системная ошибка, обычно не ожидаемая. Свяжитесь со службой поддержки ALTUS, чтобы сообщить об этом.
			FALSE — Промежуточная структура данных соответствует ожидаемой.
	4	bExchangeSync	TRUE — служба синхронизации диагностики и обмена командами была успешно выполнена в этом цикле MainTask.
			FALSE — структура RedDgnRem имеет устаревшие или недопустимые значения, так как она не была прочитана из другого ПЛК (удаленного) в этом цикле.

7. ОБСЛУЖИВАНИЕ

5	bRedDataSync	TRUE — служба резервной синхронизации данных была успешно выполнена в этом цикле MainTask.
		FALSE — служба резервной синхронизации данных не была успешно выполнена в этом цикле MainTask.
6	bRedForceSync	TRUE — служба синхронизации избыточного списка форсирования была успешно выполнена в этом цикле MainTask.
		FALSE — служба резервной принудительной синхронизации списка не была успешно выполнена в этом цикле MainTask.
7	bApplicationIncompatible	TRUE — Приложение несовместимо между ПЛК. Произведена загрузка нового приложения с одним из следующих изменений: Изменения в избыточной области памяти; Изменения в символьных избыточных переменных. Пока это диагностическое сообщение истинно, один из ПЛК будет оставаться в неактивном состоянии до тех пор, пока одно и то же приложение не будет присутствовать в двух ПЛК. Это подразумевает перезагрузку старого приложения или загрузку нового приложения в оба ПЛК. Дополнительную информацию о том, как действовать, можно найти в разделе Загрузка программы резервного ЦП.
		FALSE — приложение совместимо между ПЛК.
0	Reserved	Зарезервированный бит.

Прямая переменная	AT переменная DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Описание
переменная	Бит	

7. ОБСЛУЖИВАНИЕ

%QB(n+5)	1	bProjectSyncDisable	TRUE – Приложение проекта и архив проекта не будут синхронизированы между ПЛК. Это копия энергонезависимой переменной, используемой для включения или отключения синхронизации проекта, как описано в разделе «Отключение синхронизации проекта». Синхронизация проекта Выключено в локальном или удаленном ПЛК. Таким образом, достаточно выполнить команду отключения в одном ПЛК, чтобы синхронизация проекта была Выключено. Команды включения и отключения синхронизации проекта описаны в разделе «Отключение синхронизации проекта».
			FALSE – Приложение проекта и архив проекта будут синхронизированы между ПЛК.
	2	blncompatibleFirmware	TRUE — версия микропрограммы несовместима между этим ЦП и удаленным ЦП.
			FALSE — версия микропрограммы совместима между этим ЦП и удаленным ЦП.
	3	bApplicationProjectDiff	TRUE — Приложение проекта между этим ЦП и удаленным ЦП отличается.
			FALSE — Приложение проекта между этим ЦП и удаленным ЦП одинаково.
4	bProjectArchiveDiff	TRUE – Архив проекта между этим CPU и удаленным отличается.	
		FALSE – Архив проекта между этим и удаленным процессором одинаков.	
5	bOnlineChangeApply	TRUE – некоторые изменения были внесены онлайн в приложение, и оно еще не было синхронизировано с резервным ПЛК.	
		FALSE – В приложении не было онлайн-изменений или они уже были синхронизированы с резервным ПЛК.	
6	bFailedRED	TRUE — сбой в модуле NX4010. ЦП NX3030 не может обмениваться данными с этим модулем по шине, или произошел сбой в микропроцессоре NX4010.	

7. ОБСЛУЖИВАНИЕ

		FALSE — Модуль NX4010 работает нормально.
--	--	---

Прямая переменная		АТ переменная DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Описание
Переменная	Бит		
%QB(n+6)	7	bFailedPBUS1A	<p>TRUE — Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 1 A. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.</p> <p>FALSE — в сети PROFIBUS 1 A нет сбоев.</p>
	0	bFailedPBUS1B	<p>TRUE — Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 1 B. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.</p> <p>FALSE — в сети PROFIBUS 1 B нет сбоев.</p>
	1	bFailureProfibus_1	<p>TRUE — Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 1. В случае резервирования сети PROFIBUS 1 отказ PROFIBUS_1 является результатом логического И между FailedPBUS1A и FailedPBUS1B. В случае, если сеть PROFIBUS 1 не является избыточной, FailurePROFIBUS_1 является копией FailedPBUS1A.</p> <p>FALSE — в сети PROFIBUS нет сбоев.</p>

7. ОБСЛУЖИВАНИЕ

2	bFailedPBUS2A	TRUE – Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 2 A. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.
		FALSE — в сети PROFIBUS 2 A нет сбоев.

Direct Variable		AT variable DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Description
Variable	Bit		
	3	bFailedPBUS2B	TRUE – Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 2 B. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.
			FALSE — в сети PROFIBUS 2 B нет сбоев.
	4	bFailureProfibus_2	TRUE – Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 2. В случае резервирования сети PROFIBUS 2, FailurePROFIBUS_2 является результатом логического И между FailedPBUS2A и FailedPBUS2B. В случае, если сеть PROFIBUS 2 не является избыточной, FailurePROFIBUS_2 является копией FailedPBUS2A.
	5	bProfibusVitalFailureAny	TRUE – этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) по крайней мере с одной из сетей PROFIBUS, сконфигурированных в режиме серьезного отказа.

7. ОБСЛУЖИВАНИЕ

			FALSE — в сетях PROFIBUS нет сбоев, настроенных на существенный сбой.
6	bProfibusVitalFailureAll		TRUE — этот ПЛК не может обмениваться данными в состоянии ведущего (активном или пассивном) со всеми сетями PROFIBUS, сконфигурированными в режиме критического отказа. FALSE — в сетях PROFIBUS нет сбоев, настроенных на существенный сбой.
7	bTurnOffOtherPLC_Normal		TRUE — этот ПЛК замыкает реле PX2612, чтобы другой ПЛК оставался выключенным в нормальных условиях, а не из-за проверки панели PX2612. FALSE — реле PX2612 включено (bTurnOffOtherPLC_TestMode) или выключено.
0	bTurnOffOtherPLC_TestMode		TRUE — этот ПЛК замыкает реле PX2612, чтобы другой ПЛК оставался выключенным из-за тестового режима панели PX2612. FALSE — реле PX2612 включено (bTurnOffOtherPLC_Normal) или выключено.
1	bActiveLED		TRUE — индикатор PX2612 ACTIVE горит.

Прямая переменная		АТ переменной DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Описание
Переменная	Бит		
%QB(n+7)			FALSE — индикатор PX2612 ACTIVE мигает (bBlinkActiveLED) или не горит.
	2	bBlinkActiveLED	TRUE — индикатор PX2612 ACTIVE мигает. FALSE — PX2612 ACTIVE горит (bActiveLEDI) или не горит.
	3	bStandbyLED	TRUE — индикатор STAND-BY PX2612 горит. FALSE — индикатор PX2612 ACTIVE мигает (bBlinkStandbyLED) или не горит.
	4	bBlinkStandbyLED	TRUE — индикатор STAND-BY PX2612 мигает. FALSE — индикатор STAND-BY PX2612 горит (bStandbyLED) или не горит.
	5	bInactiveLED	TRUE — индикатор PX2612 НЕАКТИВЕН горит.

7. ОБСЛУЖИВАНИЕ

			FALSE — индикатор PX2612 INACTIVE не горит или мигает (bBlinkInactiveLED).
	6	bBlinkInactiveLED	TRUE — индикатор PX2612 НЕАКТИВЕН мигает. FALSE — индикатор PX2612 INACTIVE горит (bInactiveLED) или не горит.
	7	bRedPanelTestMode	TRUE — панель PX2612 находится в тестовом режиме. FALSE — панель PX2612 находится в нормальном режиме.
%QB(n+8)	-	ePLC_ID	Эта диагностика информирует эту идентификацию ПЛК: - 0 = без резервирования - 2 = ПЛКА - 3 = ПЛКБ Это копия энергонезависимой переменной, используемой для идентификации ПЛК, как описано в разделе «Идентификация ЦП NX3030». В разделе Начальная загрузка избыточного проекта описана команда MasterTool, используемая для записи в эту энергонезависимую переменную.
%QB(n+9)	-	eRedState	Информирует о состоянии резервирования этого ПЛК: - Не настроено = 0 - Пуск = 2 - В режиме ожидания = 3 - Активный = 4 - Неактивен = 5
%QB(n+10)	-	ePreviousRedState	Предыдущее значение RedState перед переходом данных.
%QW(n+11)	-	wRedStateDuration	Измеряет, как долго (в миллисекундах) предполагается текущее состояние избыточности. Это время прекращает увеличиваться, когда достигает 65535 мс.
Прямая переменная		AT переменная DG_NX4010.tRedundancy. RedDgnLoc.sGeneral_Diag.*	Описание
Переменная	Бит		
%QW(n+13)	-	wConfigErrorCode	Код ошибки, обнаруженный в процессе настройки в состоянии Not-Configured. См. диагностику ConfigError, описанную ранее.
%QD(n+15)	-	dwApplicationCRC	32-битный CRC проекта приложения, используемый для обнаружения

			различий между проектами приложений двух ПЛК.
%QD(n+19)	-	dwArchiveCRC	32-битный CRC архива проекта, используемый для обнаружения различий между архивом проекта двух ПЛК.
%QD(n+23)	-	dwFirmwareVersion	Эта версия прошивки ПЛК используется для проверки совместимости обеих прошивок ПЛК.
%QD(n+27)	-	dwIECTimer	Синхронизация таймера IEC необходима для бесперебойной работы некоторых функциональных блоков, таких как TON и TOF. Благодаря этой диагностике таймер IEC от активного ПЛК принимается и обновляется в неактивном ПЛК, поскольку служба диагностики и обмена командами выполнена успешно. Счет начинается с 0 и увеличивается до 4294967295.
%QW(n+31)	-	wCycleCounter	После подсчета переполнение перезапускается с 0. 16-битный счетчик, используемый в качестве вспомогательной информации о последовательности в журнале событий резервирования. В активном ПЛК он увеличивается в каждом цикле MainTask. В неактивном ПЛК получает копию значения, существующего в активном ПЛК, поскольку служба диагностики и обмена командами выполнена успешно. Счет начинается с 0 и увеличивается до 65535. После подсчета переполнение перезапускается с 0.

Таблица 209: Общая диагностика резервирования

Примечания:

Визуализация диагностических структур: диагностические структуры, добавленные в проект, можно визуализировать при доступе к диспетчеру библиотек из древовидного представления в окне MasterTool IEC XE. При этом можно визуализировать все типы данных, определенные в структуре.

Переменные прямого представления: «n» представляет значение, сконфигурированное в модуле NX4010 с помощью программного обеспечения MasterTool IEC XE, в качестве %Q начального адреса диагностики модуля. Это определение справедливо для всей структуры диагностики.

Директива AT: Директива AT — это слово, зарезервированное в программном обеспечении, которое связано с переменным адресом. В случае модуля NX4010 переменная DG_NX4010 связана с начальным адресом диагностики модуля.

Подструктура sNETA_Diag имеет следующие поля для диагностики каналов синхронизации NETA:

DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync: когда эта диагностическая переменная имеет значение FALSE, невозможно определить состояние некоторых других диагностических данных, таких как bIncompilerFirmware, bApplicationProjectDiff и bProjectArchiveDiff. Там не будет правильного значения, потому что они зависят от правильного функционирования связи между двумя процессорами, чтобы информация могла быть правильно сгенерирована. Примечания:

7. ОБСЛУЖИВАНИЕ

Визуализация диагностических структур: диагностические структуры, добавленные в проект, можно визуализировать при доступе к диспетчеру библиотек из древовидного представления в окне MasterTool IEC XE. При этом можно визуализировать все типы данных, определенные в структуре.

Переменные прямого представления: «n» представляет значение, сконфигурированное в модуле NX4010 с помощью программного обеспечения MasterTool IEC XE, в качестве %Q начального адреса диагностики модуля. Это определение справедливо для всей структуры диагностики.

Директива AT: Директива AT — это слово, зарезервированное в программном обеспечении, которое связано с переменным адресом. В случае модуля NX4010 переменная DG_NX4010 связана с начальным адресом диагностики модуля.

Подструктура sNETA_Diag имеет следующие поля для диагностики каналов синхронизации NETA:

DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync: когда эта диагностическая переменная имеет значение FALSE, невозможно определить состояние некоторых других диагностических данных, таких как bIncompilerFirmware, bApplicationProjectDiff и bProjectArchiveDiff. Там не будет правильного значения, потому что они зависят от правильного функционирования связи между двумя процессорами, чтобы информация могла быть правильно сгенерирована.

Прямая переменная		AT переменная DG_NX4010.tRedundancy. RedDgnLoc.sNETA_Diag.*	Описание
Переменная	Бит		
%QB(n+33)	0	bGeneralFailure	TRUE – в канале синхронизации произошел какой-либо сбой. 3 следующих диагностических сообщения укажут конкретную неисправность. FALSE – Канал синхронизации работает правильно.
	1	bInternalFailure	TRUE – причина обнаруженного сбоя находится в этом ПЛК. Такие сбои лечатся особым образом. FALSE — Модуль NX4010 работает нормально.
	2	bLinkDownFailure	TRUE — Кабель AL-2319A отсоединен от модуля NX4010 или поврежден в одном из ПЛК. FALSE — кабель AL-2319A подключен к модулю NX4010.
	3	bTimeoutFailure	TRUE — об этом сбое сообщается в случае, если служба синхронизации не была успешно завершена в течение определенного времени ожидания и не были найдены сбои типа bInternalFailure или bLinkDownFailure, оправдывающие это. FALSE — Модуль NX4010 работает нормально.
	4 .. 7	bReserved[4..7]	Сдержанный.

Табл. 210: Специальная диагностика интерфейса NETA

Подструктура «sNETB_Diag» имеет следующие поля для диагностики каналов синхронизации NETB:

7. ОБСЛУЖИВАНИЕ

Прямая переменная		AT переменная DG_NX4010.tRedundancy. RedDgnLoc.sNETB_Diag.*	Описание
Переменная	Бит		
%QB(n+34)	0	bGeneralFailure	TRUE – в канале синхронизации произошел какой-либо сбой. 3 следующих диагностических сообщения укажут конкретную неисправность. FALSE – Канал синхронизации работает правильно.
	1	bInternalFailure	TRUE – причина обнаруженного сбоя находится в этом ПЛК. Такие сбои лечатся особым образом. FALSE — Модуль NX4010 работает нормально.
	2	bLinkDownFailure	TRUE — Кабель AL-2319B отсоединен от модуля NX4010 или поврежден в одном из ПЛК. FALSE — кабель AL-2319B подключен к модулю NX4010.
Прямая переменная		AT переменной DG_NX4010.tRedundancy. RedDgnLoc.sNETB_Diag.*	Описание
Переменная	Бит		
	3	bTimeoutFailure	TRUE — об этом сбое сообщается в случае, если служба синхронизации не была успешно завершена в течение определенного времени ожидания и не были найдены сбои типа bInternalFailure или bLinkDownFailure, оправдывающие это. FALSE — Модуль NX4010 работает нормально.
	4..7	bReserved[4..7]	Сдержанный.

Таблица 211: Специальная диагностика интерфейса NETB

Подструктура «sNET_Stat» содержит статистику успешных и неудачных служб. Статистика локальных и удаленных ПЛК может быть перезапущена с помощью команд:

```
//Локальный ПЛК
DG_NX4010.tRedundancy.RedCmdLoc.bResetNETStatisticsLocal := TRUE;
//Удаленный ПЛК
DG_NX4010.tRedundancy.RedCmdLoc.bResetNETStatisticsRemote := TRUE;
```

Подструктура имеет следующие счетчики:

Прямая переменная	AT переменная DG_NX4010.tRedundancy. DgnLoc.sNET_Stat.*	Описание
%QW(n+35)	wSuccessExchDgCmdSync	Успешность подсчета диагностики и Служба команд (от 0 до 65535)
%QW(n+37)	wFailedExchDgCmdSync	Успешность подсчета диагностики и Служба команд (от 0 до 65535)
%QW(n+39)	wSuccessRedDataSync	Успешность подсчета диагностики и Служба команд (от 0 до 65535)
%QW(n+41)	wFailedRedDataSync	Успешность подсчета диагностики и Служба команд (от 0 до 65535)
%QW(n+43)	wSuccessRedForceSync	Подсчет успешных операций службы синхронизации резервного списка форсирования (от 0 до 65535)
%QW(n+45)	wFailedRedForceSync	Подсчет отказов резервного форсирования Служба синхронизации списка (от 0 до 65535)

Табл. 212: Специальная диагностика интерфейса

Примечание:

Счетчики: Все счетчики возвращаются к нулю при превышении их предельного значения.

Подструктура «sGeneral_DiagExt» содержит следующие поля для общей диагностики резервирования:

Прямая переменная		AT переменная DG_NX4010.tRedundancy .RedDgn- Loc.sGeneral_DiagExt.*	Описание
Переменная	Бит		
%QB(n+47)	0	bFailedPBUS3A	TRUE – этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 3 A. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.
			FALSE – в сети PROFIBUS 3 A нет сбоев.

7. ОБСЛУЖИВАНИЕ

1	bFailedPBUS3B	TRUE – Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 3 В. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.
		FALSE – В сети PROFIBUS 3 В не бывает сбоев.
2	bFailureProfibus_3	TRUE – этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 3. В случае резервирования сети PROFIBUS 3, FailurePROFIBUS_3 является результатом логического И между FailedPBUS3A и FailedPBUS3B. В случае, если сеть PROFIBUS 3 не является избыточной, FailurePROFIBUS_3 является копией FailedPBUS3A.
		FALSE – в сети PROFIBUS 3 нет сбоев.
3	bFailedPBUS4A	TRUE – Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 4 А. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.
		FALSE – В сети PROFIBUS 4 А нет сбоев.

Прямая переменная		AT переменная DG_NX4010.tRedundancy .RedDgn- Loc.sGeneral_DiagExt.*	Описание
Переменная	Бит		

7. ОБСЛУЖИВАНИЕ

	4	bFailedPBUS4B	<p>TRUE – Этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 4 В. Ведущий режим (связь с ведомыми) принимается активным ПЛК. Пассивный режим (связь с активным мастером) принимается неактивным ПЛК. Этот сбой также может указываться в случае сбоя микропроцессора модуля NX5001 или в случае, если он не может связаться с ЦП NX3030 через шину.</p> <p>FALSE — в сети PROFIBUS 4 В нет сбоев.</p>
	5	bFailureProfibus_4	<p>TRUE — этот ПЛК не может обмениваться данными в состоянии ведущего (активного или пассивного) в сети PROFIBUS 4. В случае резервирования сети PROFIBUS 4 отказ PROFIBUS_4 является результатом логического И между FailedPBUS4A и FailedPBUS4B. В случае, если сеть PROFIBUS 4 не является избыточной, FailurePROFIBUS_4 является копией FailedPBUS4A.</p> <p>FALSE — в сети PROFIBUS 4 нет сбоев.</p>
	6	bFailureCommBusAny	<p>TRUE – Ошибка в какой-либо коммуникационной полевой шине. Как пример порт Ethernet с протоколом MODBUS настроен на жизненный отказ.</p> <p>FALSE – нет отказов полевой шины.</p>
	7	bFailureCommBusAll	<p>TRUE – Нарушена связь по всей полевой шине. Например, NET 1 и NET 2 настроены как жизненно важные, и обе находятся в состоянии сбоя.</p> <p>FALSE – По крайней мере одна коммуникационная полевая шина работает без сбоев.</p>
%QB(n+48)	-	byFailedCommBusCount	Количество коммуникационных полевых шин, которые сообщают об ошибках.
%QB(n+49)	0	bRemCpuKeepAliveNet1	<p>TRUE — ЦП NX3030 получает пакеты проверки активности от ЦП другого полукластера через NET 1. Пакеты проверки активности отправляются только в проектах, не использующих панель PX2612 и не имеющих сети PROFIBUS.</p> <p>FALSE — пакеты Keep Alive не принимаются.</p>

	1	bRemCpuKeepAliveNet2	TRUE — ЦП NX3030 получает пакеты Keep Alive от ЦП другого полукластера через NET 2. FALSE — пакеты Keep Alive не принимаются.
Прямая переменная		AT переменная DG_NX4010.tRedundancy .RedDgn- Loc.sGeneral_DiagExt.*	Описание
Переменная	Бит		
	2..7	(Occulted reserved bits)	Биты зарезервированы для использования в будущем. В символической структуре они не показаны (скрыты).
%QB(n+50)	-	abyReservedBytes[1..5]	5 зарезервированных байтов для будущего использования. В символической структуре они не показаны (скрыты).

Таблица 213: Общая диагностика резервирования, расширенная часть

6.6.4.2. Команды резервирования

Поля команды структуры RedCmdLoc и RedCmdRem имеют суффикс, который может быть локальным или удаленным. Например, существуют командные поля StandbyLocal и StandbyRemote, действие которых эквивалентно кнопке STAND-BY на панели PX2612.

Команда с суффиксом Local, сгенерированная в RedCmdLoc, должна выполняться в локальном ПЛК (local). С другой стороны, команда с суффиксом Remote, сгенерированная в RedCmdLoc, будет выполняться в другом ПЛК (удаленном). Это работает следующим образом:

Удаленный ПЛК каждый цикл MainTask получает копию RedCmdLoc от локального ПЛК через NETA/NETB, и эта копия называется в нем RedCmdRem

Удаленный ПЛК выполняет только команды RedCmdRem с суффиксом Remote.

Пример 1: если локальный ПЛК находится в активном состоянии, и его желательно переключить в состояние ожидания, в нем можно включить бит DG_NX4010.tRedundancy.RedCmdLoc.bStandbyLocal.

Пример 2: если удаленный ПЛК находится в активном состоянии и требуется переключить его в состояние ожидания, в локальном ПЛК можно установить бит DG_NX4010.tRedundancy.RedCmdLoc.bStandbyRemote. Это может быть полезно, например, если связь системы SCADA с удаленным ПЛК временно отсутствует. В этом случае команда записывается SCADA в локальный ПЛК, который ретранслирует на удаленный ПЛК через NETA/NETB.

ВНИМАНИЕ

Если диагностика DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bExchangeSync указывает на отказ службы диагностики и обмена командами, команда с суффиксом Remote не может быть передана на удаленный ПЛК, поэтому она не будет выполнена.

7. ОБСЛУЖИВАНИЕ

Чтобы запустить команду, соответствующий бит RedCmdLoc должен быть включен. Это можно сделать через систему SCADA, выполнив запись через MasterTool или даже включив бит внутри POU как ActivePrg или NonSkippedPrg.

Пользователю не нужно беспокоиться об отключении командного бита, что автоматически выполняется менеджером резервирования:

В случае команд, выполняемых в локальном ПЛК (RedCmdLoc + команды с суффиксом Local), бит отключается, как только команда видна и выполняется.

В случае команд, выполняемых в удаленном ПЛК (RedCmdRem + команды с суффиксом Remote):

- В удаленном ПЛК команда выполняется, когда диспетчер резервирования видит восходящий фронт в командном бите.
- В локальном ПЛК, где была сгенерирована команда, бит автоматически отключается в следующем цикле MainTask.

ВНИМАНИЕ

Есть два командных бита, которые обычно должны быть отключены пользователем: DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal and DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal.

Более подробная информация об этих командах описана в этом разделе. На случай, если пользователь забудет их отключить, вместо этого это должны сделать автоматические механизмы.

Важно подчеркнуть, что любая команда активации или деактивации регистрируется в журнале событий резервирования, что важно для анализа истории, например. определить причину переключения.

Затем определяются поля структуры RedCmdLoc и RedCmdRem:

Прямая переменная		АТ переменная DG_NX4010.tRedundancy. RedCmdLoc.*	Описание
Переменная	Бит		
%QB(n+55)	0	bButtonTurnOnLocal	TRUE — это обработанная копия кнопки TURN ON PLCX, написанной на панели PX2612. Этот бит активируется через 1 секунду после нажатия кнопки и деактивируется сразу же при ее отпускании. Важно подчеркнуть, что этот бит будет активирован при нажатии кнопки TURN ON на удаленном ПЛК, так как этот тип команды всегда отправляется удаленным ПЛК. FALSE — кнопка ВКЛЮЧИТЬ PLCX не нажата.
	1	bButtonStandbyLocal	TRUE — это обработанная копия с кнопки STAND-BY, написанной на панели PX2612. Этот бит активируется через 1 секунду после нажатия кнопки и деактивируется сразу же при ее отпускании. FALSE — кнопка STAND-BY не нажата.

7. ОБСЛУЖИВАНИЕ

2	bButtonInactiveLocal	TRUE — это обработанная копия кнопки INACTIVE, написанной на панели PX2612. Этот бит активируется через 1 секунду после нажатия кнопки и деактивируется сразу же при ее отпускании.
		FALSE – кнопка INACTIVE не нажата.
3	bAutoConfigLocal	TRUE — эта диагностика информирует об автоматическом запросе конфигурации, необходимом для разрешения состояния «Не настроено» в некоторых ситуациях.
		FALSE — автоматический запрос конфигурации отключен.
4	bTurnOnLocal	TRUE — эта команда производит действие, эквивалентное действию кнопки TURN ON PLCX на PX2612 в локальном ПЛК.
		FALSE — кнопка TURN ON PLCX на локальном ПЛК не нажата.
5	bStandbyLocal	TRUE — эта команда производит действие, эквивалентное нажатию кнопки STAND-BY на PX2612 в локальном ПЛК.
		FALSE – Кнопка STAND-BY на локальном ПЛК не нажата.
6	bInactiveLocal	TRUE — эта команда производит действие, эквивалентное нажатию кнопки INACTIVE на PX2612 в локальном ПЛК.
		FALSE — кнопка INACTIVE на локальном ПЛК не нажата.
7	bResetNETStatisticsLocal	TRUE — эта команда сбрасывает статистику NETA/NETB (см. подструктуру SNET_Stat в RedDgnLoc и RedDgnRem). Такой статистикой являются счетчики неудач и успехов в службах синхронизации.

Прямая переменная		AT переменная DG_NX4010.tRedundancy. RedCmdLoc.*	Описание
Переменная	Бит		
			FALSE — команды сброса статистики NETA/NETB в локальном ПЛК не были активированы.

%QB(n+56)	0	bTestModeLocal	<p>TRUE — эта команда переводит панель PX2612 в тестовый режим, позволяя тестировать ее компоненты (светодиоды, реле и кнопки), как описано в разделе «Тестирование панели PX2612». Тестовый режим PX2612 принимается, только если этот бит установлен на обоих ПЛК. Поэтому, чтобы PX2612 находился в тестовом режиме, ПЛК проверяет наличие RedCmdLoc. TestModeLocal и RedCmdRem.TestModeLocal включены.</p> <p>Диагностика RedDgnLoc.RedPanelTestMode включена, чтобы сообщить, что PX2612 действительно находится в тестовом режиме. Обычно пользователь должен отключить бит TestModeLocal на обоих ПЛК сразу после завершения тестов PX2612, но если он забудет это сделать, бит будет отключен автоматически через 15 минут после включения.</p> <p>FALSE – команда, которая помещает Панель PX2612 в тестовом режиме деактивирована.</p>
	1	bTestRelayLocal	<p>TRUE – Эта команда используется для проверки реле PX2612 NO и, следовательно, внешнего реле NC, используемого для отключения другого ПЛК. Эта команда принимается только тогда, когда PX2612 находится в тестовом режиме, автоматически отключается и игнорируется, если PX2612 не находится в этом режиме. Обычно пользователь должен отключить бит TestRelayLocal, как только завершится тест реле, но если он забудет, бит отключается, как только тестовый режим завершается. Важно подчеркнуть, что эта команда принимается только в активном ПЛК, чтобы предотвратить ее отключение неактивным ПЛК.</p> <p>FALSE — команда, используемая для проверки реле PX2612 NO, деактивирована.</p>
	2	bStandbyRemote	<p>TRUE — эта команда производит действие, эквивалентное кнопке STAND-BY на PX2612 в удаленном ПЛК.</p>

7. ОБСЛУЖИВАНИЕ

			FALSE – Кнопка STAND-BY на удаленном ПЛК не нажата.
	3	bInactiveRemote	TRUE – Эта команда производит действие, эквивалентное нажатию кнопки НЕАКТИВНО на PX2612 в удаленном ПЛК. FALSE – кнопка INACTIVE на удаленном ПЛК не нажата.
Прямая переменная		АТ переменной DG_NX4010.tRedundancy.RedCmdLoc.*	Описание
Переменная	Бит		
	4	bResetNETStatisticsRemote	TRUE — эта команда производит действие, эквивалентное кнопке ResetNETStatisticsLocal на PX2612 в удаленном ПЛК. FALSE — команды сброса статистики NETA/NETB в удаленном ПЛК не были активированы.
	5..7	bReserved[5..7]	Зарезервировано.

Таблица 214: Команды резервирования

6.6.4.3. Обмен пользовательской информацией между PLCA и PLCB

Служба Синхронизация Обмена Диагностикой и Командами в каждом цикле MainTask осуществляет обмен следующими структурами данных между обоими ПЛК, используя каналы синхронизации NETA/NETB:

Диагностика резервирования (RedDgnLoc и RedDgnRem), уже описанная в разделе Структура диагностики резервирования.

Команды резервирования (RedCmdLoc и RedCmdRem), уже описанные в разделе «Команды резервирования».

Структуры RedUsrLoc и RedUsrRem представляют собой просто 128-байтовый массив, использование которого может свободно определяться пользователем. Они позволяют пользователю передавать в каждом цикле 128 байтов информации из PLCA в PLCB и другие 128 байтов из PLCB в PLCA.

RedUsrRem — это копия с другого ПЛК RedUsrLoc, полученная через NETA/NETB. Конкретный ПЛК записывает информацию в RedUsrLoc, которая считывается в RedUsrRem другого ПЛК.

Эти структуры данных имеют множество полезных свойств. Например, предположим, что система SCADA подключена только к активному ПЛК и требуется визуализировать некоторую информацию из неактивного ПЛК. Неактивный ПЛК может поместить эту информацию в эти структуры данных. Среди такой информации, например, могут быть неотображенные диагностики в RedDgnLoc и RedDgnRem.

6.6.4.4. Диагностика Modbus, используемая при резервировании

Для проверки сбоев на всех серверах MODBUS, сконфигурированных в экземпляре клиента MODBUS, в каждом сконфигурированном экземпляре клиента MODBUS предусмотрена специальная диагностика. В таблице ниже показана диагностика для этого типа сбоя в экземпляре клиента MODBUS с именем MODBUS_Symbol_Client. t.

Переменные DG_MODBUS_Symbol_Client.tDiag.*	Описание
--	----------

bAllDevicesCommFailure	TRUE — все серверы, настроенные на этом клиенте, показывают ошибку.
	FALSE — в этом клиенте настроен как минимум один сервер, который не показывает ошибку.

Табл. 215: Диагностика клиента Modbus

Когда сконфигурирован режим жизненно важного отказа, эта диагностика используется, и через 3 секунды после изменения ее состояния с FALSE на TRUE происходит переключение, если другие условия для этого события удовлетворены.

6.6.4.5. Журнал событий резервирования

MasterTool позволяет просматривать несколько журналов ПЛК Nexto, в том числе журнал событий резервирования. Эти сообщения, специфичные для резервирования, регистрируются в системном журнале соответствующие изменения в полях структуры данных диагностики и данных структуры команд резервирования.

Каждая строка, представленная в журнале, имеет следующие столбцы:

Отметка времени: время и дата события с разрешением в миллисекундах.

Серьезность: информация, предупреждение, ошибка или исключение

Описание: текст, описывающий событие

Компонент: компонент, вызвавший событие, а в случае с журналом событий резервирования — «Управление резервированием».

Текст столбца «Описание» содержит информацию о произошедшем событии.

Пример столбца Описание может быть следующим:

Новое состояние резервирования (локальное): Запуск

Для доступа к этому экрану необходимо дважды щелкнуть устройство (NX3030) в дереве устройств, а затем выбрать вкладку «Журнал». Есть фильтр, позволяющий выбрать только компонент «Управление резервированием», чтобы отображались только события резервирования.

ВНИМАНИЕ

Некоторые диагностики могут указывать на возможные сбои при инициализации резервной системы и в первых циклах задач. Но при правильной работе системы эти диагностики больше не указывают на ошибки сразу после инициализации системы.

6.6.5. Тест панели PX2612

Панель PX2612 состоит из кнопок, светодиодов и реле. Многие из этих ресурсов используются не очень часто, поэтому редко тестируются и дефекты могут не обнаруживаться. Важно время от времени запускать тесты, чтобы убедиться, что эти ресурсы работают должным образом, чтобы избежать скрытых сбоев и предотвратить использование PX2612, когда это необходимо.

6.6.5.1. Вход в тестовый режим

Первым шагом для тестирования PX2612 является перевод его в тестовый режим. Это делается включением командного бита DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal на обоих ПЛК.

ПЛК определяет, что он находится в тестовом режиме, когда установлены следующие два бита:

DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal (RedCmdLoc.bTestModeLocal включен в этом ПЛК)

DG_NX4010.tRedundancy.RedCmdRem.bTestModeLocal (RedCmdLoc.bTestModeLocal включен в другом ПЛК)

Когда оба бита включены, ПЛК включает диагностику DG_NX4010.tRedundancy.RedDgnLoc.sGeneral_Diag.bRedPanelTestMode, чтобы сообщить, что PX2612 находится в тестовом режиме.

6.6.5.2. Ручной и автоматический режим тестирования

Пользователь может выйти из тестового режима вручную, отключив бит DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal в обоих ПЛК. На самом деле достаточно отключить его

7. ОБСЛУЖИВАНИЕ

только в одном ПЛК, так как тестовый режим требует, чтобы этот бит был включен в обоих ПЛК. Тем не менее, эта практика рекомендуется.

Если пользователь забывает выключить бит `DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal`, он автоматически отключается через 15 минут после включения, автоматически завершая тестовый режим.

6.6.5.3. Тестирование светодиодов

Таким образом, во время тестового режима 6 светодиодов должны мигать, теряя свою обычную полезность, которая показывает состояние резервирования.

6.6.5.4. Кнопки Теста

При нажатии кнопки в тестовом режиме соответствующий светодиод перестает мигать и горит постоянно. В следующей таблице представлена связь между нажатой кнопкой и светодиодом, который остается включенным.

Проверенная кнопка	Соответствующий светодиод
TURN ON PLC A	ACTIVE PLC B
STAND-BY PLC A	STAND-BY PLC A
INACTIVE PLC A	INACTIVE PLC A
TURN ON PLC B	ACTIVE PLC A
STAND-BY PLC B	STAND-BY PLC B
INACTIVE PLC B	INACTIVE PLC B

Таблица 216: Связь между кнопками и светодиодами в тесте кнопок PX2612

Можно заметить, что обычно светодиод находится со стороны нажатой кнопки, за исключением TURN ON PLCx. Прежде чем светодиод загорится, необходимо удерживать кнопку не менее 1 секунды. Светодиод снова начнет мигать после того, как его отпустят. В тестовом режиме кнопки не позволяют выполнять функции, которые были бы выполнены вне этого режима, например, чтобы вызвать изменение состояния резервирования.

6.6.5.5. Тест реле

Для тестирования реле был создан бит `DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal`. При включении этого бита, если ПЛК находится в тестовом режиме и в активном состоянии, он активирует реле, которое должно вызвать отключение другого ПЛК (неактивного). При отключении `DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal` реле освобождается, позволяя повторно активировать другой ПЛК.

Команда не действует в неактивном ПЛК, чтобы предотвратить отключение активного ПЛК. Пользователь должен вызвать переключение между ПЛК (активный x неактивный), чтобы проверить реле в обоих ПЛК. Когда ПЛК, который был выключен, повторно активируется и перезапускается, он возвращается с отключенным параметром `DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal`, таким образом, тестовый режим отменяется. Бит `DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal` необходимо снова включить в этом ПЛК и установить его в активное состояние перед проверкой его реле. По завершении тестового режима командный бит `DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal` автоматически отключается, если пользователь забыл его включить.

6.6.5.6. Предлагаемая последовательность выполнения теста PX2612

7. ОБСЛУЖИВАНИЕ

Для выполнения тестов PX2612 предлагается следующая последовательность:

- Включите бит команды DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal в обоих ПЛК (PLCA и PLCB).
- Необходимо следить за миганием 6 светодиодов.
- Нажмите по одной на 6 кнопок и убедитесь, что соответствующий светодиод перестает мигать и продолжает гореть, пока кнопка нажата. Следует помнить, что кнопка должна оставаться нажатой не менее одной секунды, прежде чем светодиод перестанет мигать и останется включенным, и что светодиод снова начнет мигать после того, как кнопка будет отпущена.
- Включите бит команды DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal в активном ПЛК. Необходимо наблюдать за выключением неактивного ПЛК.
- Отключите бит команды DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal в активном ПЛК. Необходимо наблюдать включение неактивного ПЛК.
- Подождите, пока неактивный ПЛК не перезапустится и не перейдет в состояние ожидания. Режим тестирования активен, так как бит DG_NX4010 .tRedundancy.RedCmdLoc.bTestModeLocal выключен при перезапуске ПЛК в режиме ожидания.
- Вызвать переключение между ПЛК, нажав кнопку РЕЖИМ ОЖИДАНИЯ Активного ПЛК. Нормальное использование кнопки STAND-BY возможно, поскольку тестовый режим не активен.
- Включите бит команды DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal в новом активном ПЛК, который только что вышел из состояния ожидания. Таким образом, тестовый режим снова активируется.
- Включите бит команды DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal в активном ПЛК. Необходимо наблюдать за выключением неактивного ПЛК.
- Отключите бит команды DG_NX4010.tRedundancy.RedCmdLoc.bTestRelayLocal в активном ПЛК. Необходимо наблюдать за повторной активацией неактивного ПЛК.
- Выключите бит команды DG_NX4010.tRedundancy.RedCmdLoc.bTestModeLocal в активном ПЛК, чтобы завершить тестовый режим. Нет необходимости делать это в резервном ПЛК, поскольку он только что инициализировался со снятым битом DG_NX4010.tRedundancy .RedCmdLoc.bTestModeLocal.

7. Обслуживание

Одной из особенностей серии Nexto является генерация диагностики аномалий, будь то сбои, ошибки или режимы работы, что позволяет оператору легко выявлять и решать проблемы, возникающие в системе.

ЦП Nexto позволяют визуализировать диагностические данные, генерируемые системой, различными способами, а именно:

Диагностика одним касанием

Диагностика через светодиод

Диагностика через WEB

Диагностика через переменные

Диагностика через функциональные блоки

Первая — это инновационная функция серии Nexto, которая обеспечивает быстрый доступ к нештатным условиям приложения. Второй — чисто визуальный, формируется двумя светодиодами, размещенными на панели (DG и WD), а также светодиодами, размещенными в разьеме RJ45 (исключительно для подключения Ethernet). Следующей особенностью является графическая визуализация на веб-странице стойки и соответствующих настроенных модулей с возможностью индивидуального доступа к рабочему состоянию и активной диагностике. Диагностика также хранится непосредственно в переменных ЦП, либо в прямом представлении (%Q), либо в атрибутах (переменная AT), и может использоваться пользовательским приложением, например, в системе диспетчерского управления. Последние представляют специфические условия функционирования системы.

Эта функция диагностики предназначена для выявления возможных проблем с установкой или конфигурацией системы, а также проблем или недостатков сети связи.

7.1. Диагностика модуля

7.1.1. Диагностика одним касанием

One Touch Diag (OTD) или диагностика одним касанием — это эксклюзивная функция серии Nexto для программируемых контроллеров. С помощью этой новой концепции пользователь может проверять диагностику любого модуля, подключенного к системе, прямо на графическом дисплее ЦП одним касанием диагностического переключателя модуля. Это мощный диагностический инструмент, который можно использовать в автономном режиме (без использования программного обеспечения для контроля или программирования), что упрощает поиск и быстрое решение возможных проблем.

Клавиша диагностики расположена на верхней части ЦП, в легкодоступном месте и, помимо предоставления активной диагностики, позволяет получить доступ к навигационному меню, описанному в разделе Конфигурация – Информационное и конфигурационное меню ЦП.

На рисунке ниже показано расположение переключателя ЦП:

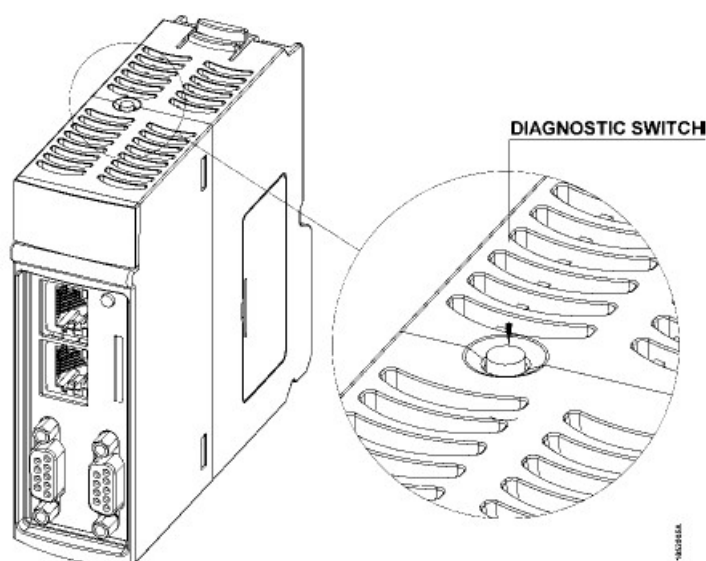


Рисунок 189: Диагностический переключатель

Одним кратким касанием ЦП начинает отображать диагностику шины (когда активен, в противном случае показывает сообщение «NO DIAG»). Первоначально тег визуализируется (настраивается в свойствах модуля в программном обеспечении MasterTool IEC XE в соответствии со стандартом IEC 61131-3), другими словами, имя, присвоенное ЦП, и после этого отображаются все диагностические данные через дисплей ЦП. Сообщения. Этот процесс выполняется дважды на дисплее. Все происходит автоматически, так как пользователю нужно выполнить только первое короткое касание, а ЦП отвечает за отображение диагностики. Диагностика других модулей, присутствующих на шине, также отображается на графическом дисплее ЦП кратким нажатием кнопки диагностического модуля в той же презентационной модели диагностики.

На рисунке ниже показан процесс, начинающийся с короткого касания, а условия и время ЦП представлены в прямоугольниках меньшего размера. Важно подчеркнуть, что диагностика может иметь более одного экрана, другими словами, указанное время на блок-схеме ниже действительно для одного из них.

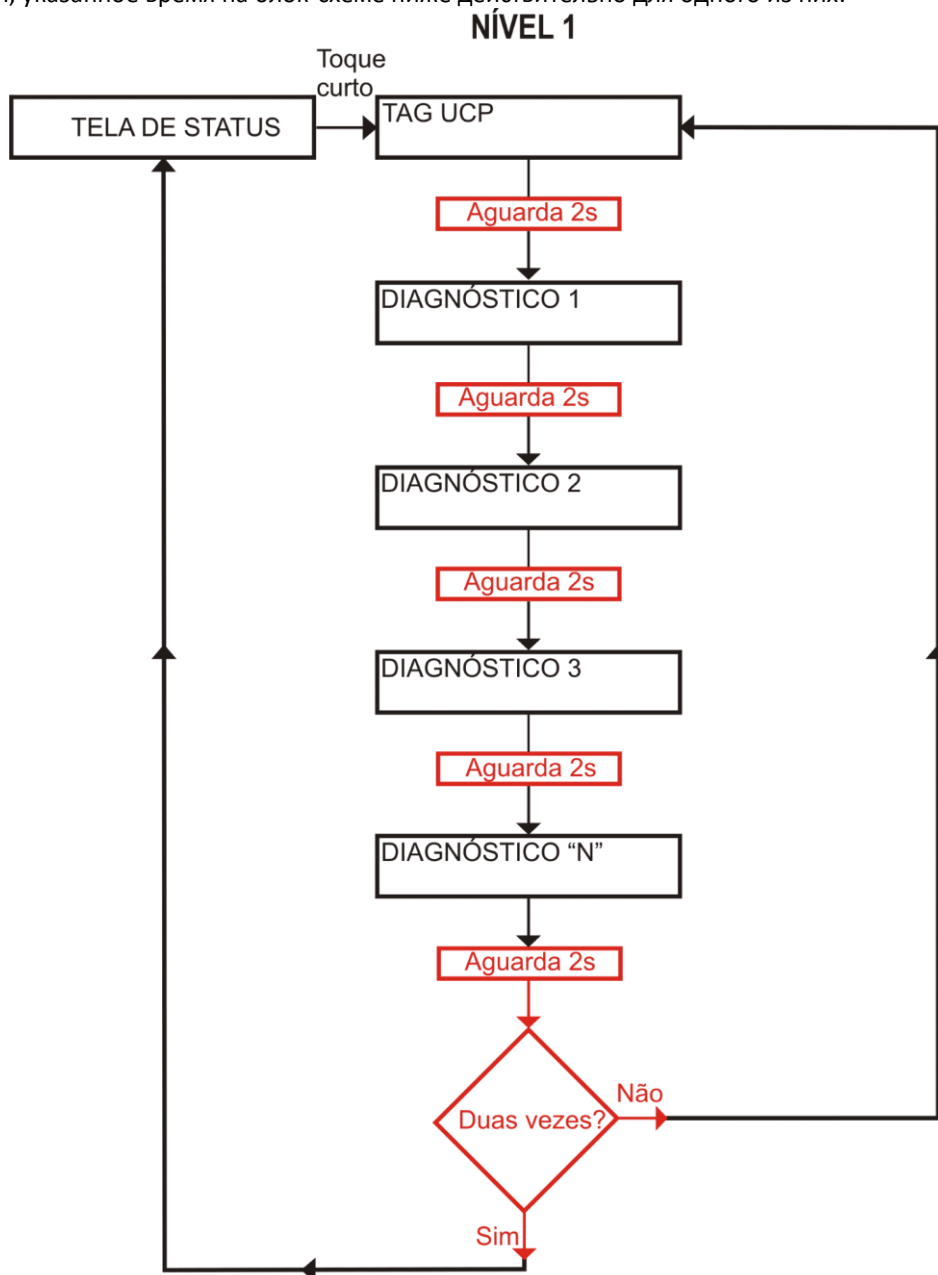


Рисунок 190: Визуализация диагностики ЦП

7. ОБСЛУЖИВАНИЕ

Перед завершением всего процесса визуализации достаточно коротко коснуться диагностического переключателя в любой момент или нажать диагностический переключатель на любом модуле ввода/вывода, подключенном к шине. Кроме того, важно отметить, что One Touch Diag может быть доступен, когда модуль находится в рабочем режиме.

В случае выполнения длинного касания ЦП переходит в меню навигации, которое описано в разделе Конфигурация – Информационное и конфигурационное меню ЦП.

В таблице ниже показана разница между коротким временем касания, длительным временем касания и залившей кнопкой.

Сенсорный тип	Минимальное время	Максимальное время	Состояние индикации
Без касаний	-	59.99 мс	-
Короткое касание	60 ммс	0.99 с	Выпускать
Длинное касание	1 с	20 с	Более 1 с до 20 с
Заблокированный переключатель	20.01 с	(∞)	Индикация диагностики, см. Таблицу 222

Таблица 217: Время одного касания

Сообщения, отображаемые на графическом дисплее ЦП Nexto, соответствующие диагностике, описаны в разделе «Диагностика с помощью переменных» в Таблице 222.

Если в одном из модулей ввода-вывода произойдет какая-либо ситуация с заеданием кнопки, диагностическая кнопка этого модуля остановится или отобразит диагностику на графическом дисплее ЦП при ее нажатии. В этом случае CPU укажет, что имеется модуль с активной диагностикой. Чтобы удалить эту диагностику из ЦП, необходимо выполнить горячую замену в модуле, где диагностика активна.

Дополнительные сведения о процедуре просмотра диагностики ЦП или других шинных модулей см. в описании в Руководстве пользователя серии Nexto — MU214600.

7.1.2. Диагностика через светодиод

Этот продукт имеет светодиод для диагностической индикации (светодиод DG) и светодиод для индикации событий сторожевого таймера (светодиод WD). Таблицы 218 и 219 показывают значение каждого состояния и его соответствующие описания.

7.1.2.1. ДГ (диагностический)

Зеленый	Красный	Описание	Случаи	Приоритетность
Выкл	Выкл	Не используется	Нет источника питания. Аппаратная проблема	-
Вкл	Выкл	Все приложения в режиме выполнения (Выполнить)	-	3 (низкий)
Выкл	Вкл	Все приложения в режиме остановки (стоп)	-	3 (низкий)
Мигает 2x	Выкл	Шинные модули с диагностикой	По крайней мере, шинный модуль, включая ЦП,	1

7. ОБСЛУЖИВАНИЕ

			имеет активную диагностику.	
Мигает 3x	Выкл	Форсирование данных	Некоторая область памяти форсируется пользователем через MasterTool IEC XE	2
Выкл	Мигает 4x	Конфигурация или аппаратная ошибка в шине	Шина повреждена или неправильно настроена	0 (высокая)

Таблица 218: Описание состояний диагностических светодиодов

7.1.2.2. WD (сторожевой таймер)

Красный экран	Описание	Случаи	Приоритетность
Выкл	Нет индикации сторожевого таймера	Нормальная операция	3 (низкий)
Мигает 1x	Программный сторожевой таймер	Контрольный таймер пользовательского приложения	2
Вкл	Аппаратный сторожевой таймер	Поврежденный модуль и/или поврежденная операционная система	1 (высокий)

Таблица 219: Описание состояний светодиода сторожевого таймера

Примечания:

Программный сторожевой таймер: Чтобы удалить индикатор сторожевого таймера, выполните сброс приложения или выключите и снова включите ЦП. Этот сторожевой таймер возникает, когда время выполнения пользовательского приложения превышает настроенное время сторожевого таймера.

Диагностику можно проверить в переменной Exception.wExceptionCode, см. Таблицу 226.

Аппаратный сторожевой таймер: Для сброса любой индикации сторожевого таймера, например, светодиода WD или операнда Reset.bWatchdogReset, модуль должен быть отключен от источника питания.

Чтобы проверить условия приложения при перезапуске модуля, см. конфигурации в Таблице 42.

7.1.2.3. Индикаторы разъема RJ45

Оба светодиода, размещенные в разъемах RJ45, помогают пользователю в обнаружении проблем с установленной физической сетью, указывая скорость сетевого соединения и наличие трафика связи интерфейса. Значение светодиодов представлено в таблице ниже.

Желтый	Зеленый	Значение
◦	◦	Сетевая ссылка отсутствует
•	◦	10 Мбайт/с сеть ССЫЛКА
•	•	Сеть 100 Мбайт/с ССЫЛКА

X	-	Передача или прием по сети Ethernet для этого IP-адреса или на него. Мигает по запросу процессора Nexto, а не при каждой передаче или приеме, другими словами, может мигать на более низкой частоте, чем реальная частота передачи или приема
---	---	---

Таблица 220: Значение индикатора Ethernet

7.1.3. Диагностики через WEB

Помимо ранее представленных функций, серия Nexto предоставляет пользователю инновационный инструмент доступа к диагностике системы и рабочим состояниям через веб-страницу.

Использование, помимо того, что является динамичным, очень интуитивно понятно и облегчает работу пользователя. Использование системы контроля можно заменить, если оно ограничено проверкой состояния системы.

Чтобы получить доступ к нужной веб-странице ЦП, достаточно использовать стандартный навигатор (Internet Explorer 7 или выше, Mozilla Firefox 3.0 или выше и Google Chrome 8 или выше) и ввести в адресной строке IP-адрес ЦП (пример. : <http://192.168.1.1>).

Во-первых, представлена информация о ЦП, как показано на рис. 191:

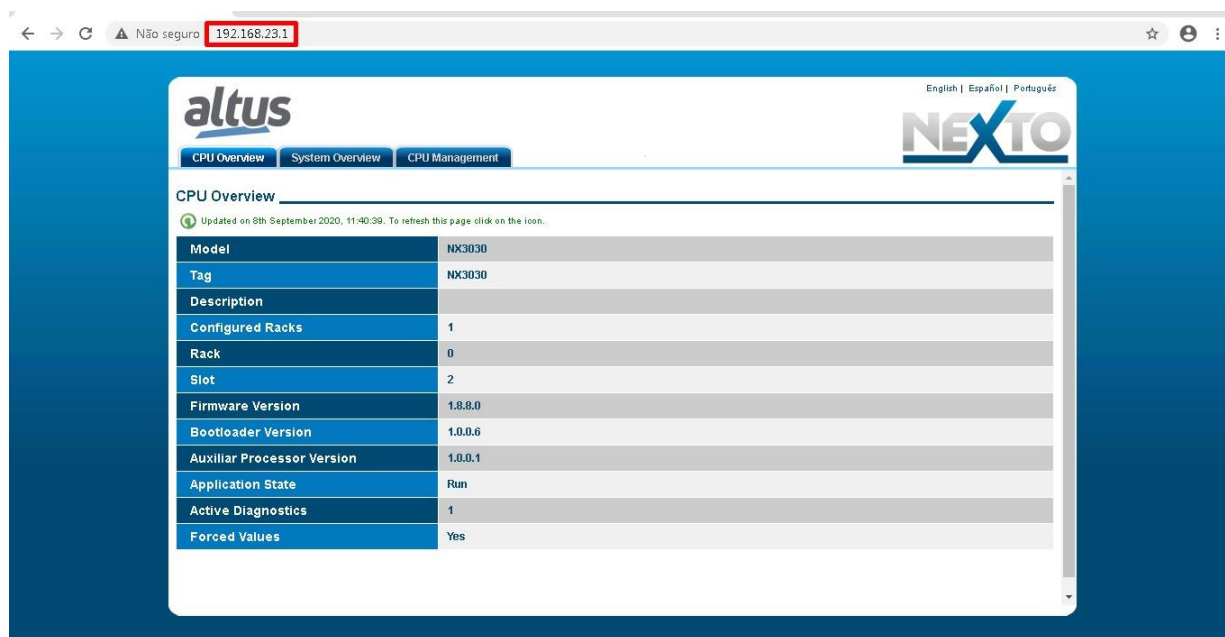


Рисунок 191: Начальный экран

Существует также вкладка «Обзор системы», которую можно просмотреть в стойке или в списке имеющихся модулей (опция в правой части экрана). Пока на ЦП нет приложения, на этой странице будет отображаться конфигурация с самой большой доступной стойкой и стандартным блоком питания, подключенным к ЦП. При использовании визуализации Rack модули с диагностикой мигают и окрашиваются в красный цвет, как показано на рис. 192. В противном случае отображается список с подключенными к системе модулями, тегами и номером активной диагностики:

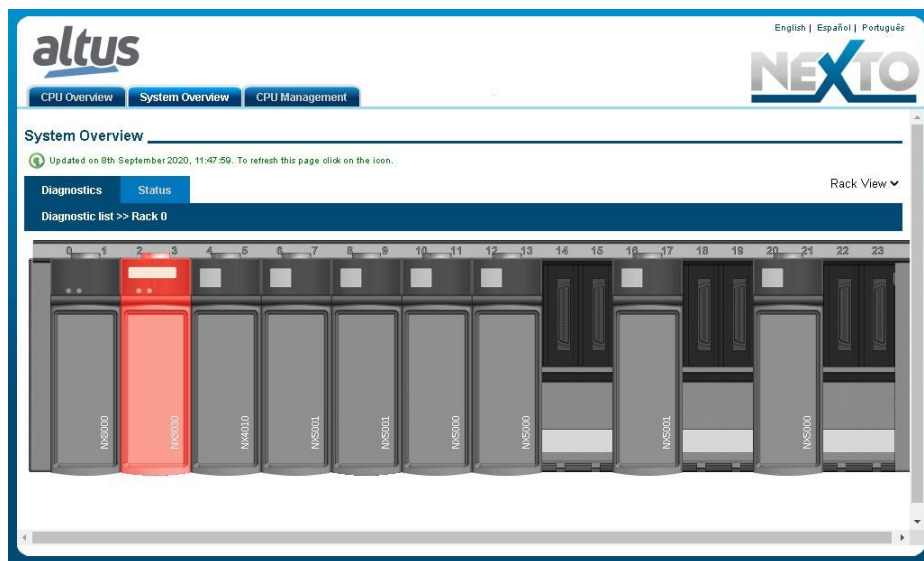


Рисунок 192: Системная информация

Когда модуль с диагностикой нажат, отображается активный(е) модуль(и) с диагностикой(ями), как показано на Рисунке 193:

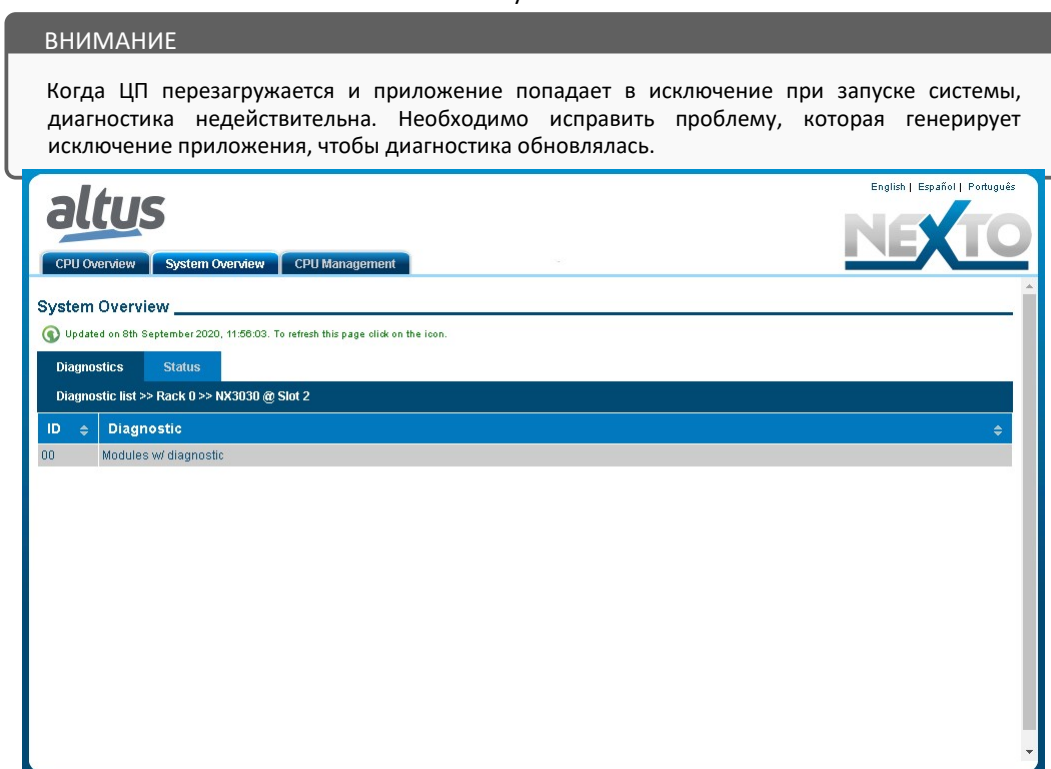


Рисунок 193: Диагностики системы

Если выбрана вкладка «Статус», на экране отображается состояние всей подробной диагностики, как показано на рис. 194:

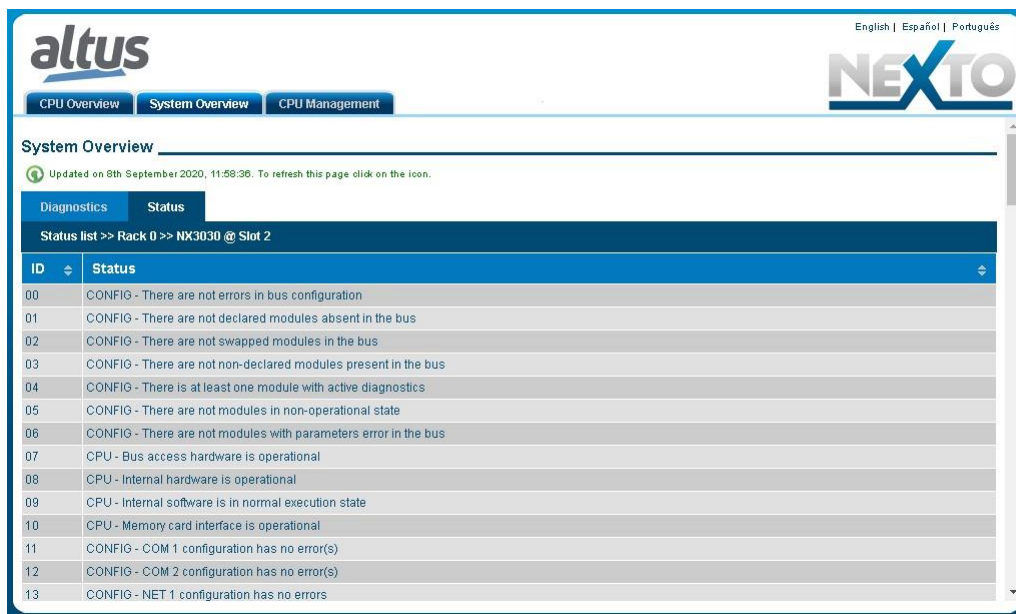


Рисунок 194: Статус системы

Пользователь может выбрать для визуализации три варианта языка: португальский, английский и испанский. Это просто изменить верхнее правое меню на нужный язык. Кроме того, на вкладке «Управление ЦП» есть другие функции, такие как обновление прошивки и SNMP.

Вкладка «Обновление прошивки» доступна только пользователю, то есть только для внутреннего использования Altus. В случаях, когда обновление выполняется удаленно (например, по радио или спутниковому соединению), минимальная скорость соединения должна быть 128 Кбит/с.

7.1.4. Обзорщик диагностики

Diagnostic Explorer — это включение диагностики через WEB в MasterTool IEC XE, чтобы сделать процесс более быстрым и прямым.

Доступ к этой функции происходит двумя способами:

- Доступ к опции «Diagnostic Explorer» в дереве устройств, размещенной на экране MasterTool IEC XE слева, и указание правильного IP-адреса в поле, указанном на рисунке 195. Помните, что для отображения страницы диагностики пользователь должен быть подключен к ЦП (начальное программирование – раздел входа в систему).

7. ОБСЛУЖИВАНИЕ

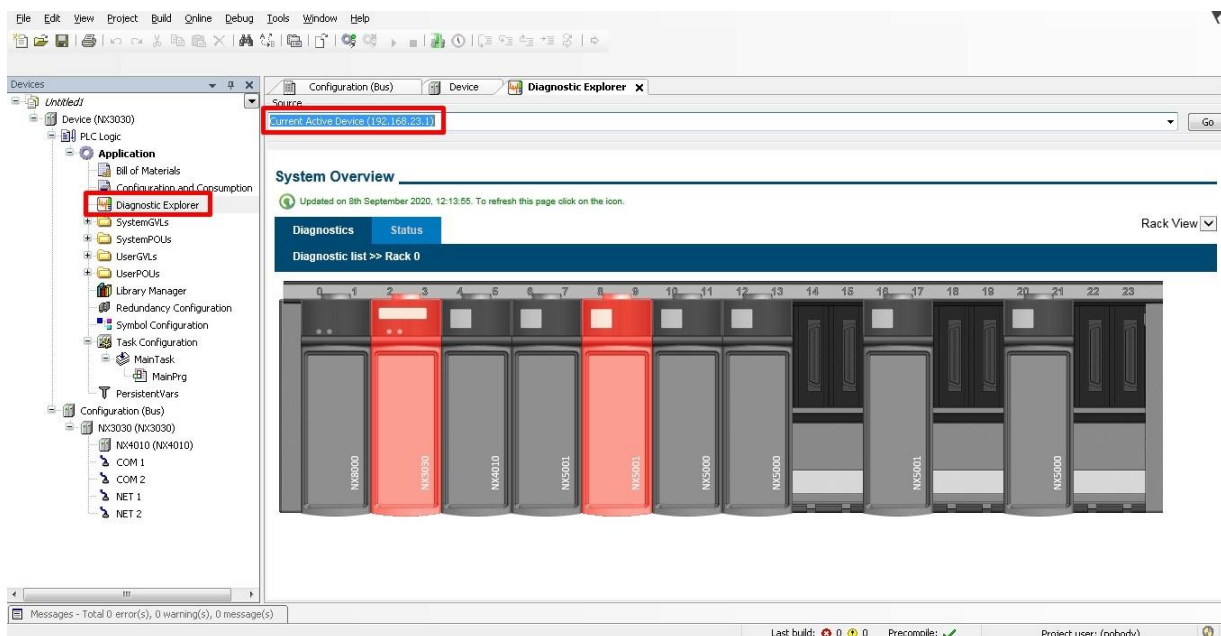


Рисунок 195: Экран проводника диагностики

-Щелкнув правой кнопкой мыши на модуле и выбрав «Диагностика», откроется Проводник диагностики, направляющий на страницу состояния модуля.

7.1.5. Диагностика через переменные

ЦП серии Nexto имеют множество переменных для диагностической индикации. Имеются структуры данных с диагностикой всех модулей, объявленных на шине, отображенных на переменные прямого представления %Q и определенные символически через директиву AT, в GVL System_Diagnostics, созданные автоматически MasterTool IEC XE.

В таблице ниже показано разделение диагностических байтов/слов:

Байт	Описание
0 до 3	Сводная диагностика процессора.
4 до 693	Детальная диагностика процессора.

Таблица 221: Отдел диагностики ЦП

7.1.5.1. Обобщенная диагностика

В таблице ниже показано значение каждого суммарного диагностического бита ЦП:

Прямая переменная		Сообщение диагностики	AT переменная DG_Module.tSummarized.*	Описание
Переменная	Бит			
-	-	NO DIAG	-	Активной диагностики нет.
%QB(n)	0	CONFIG. MISMATCH	bConfigMismatch	TRUE – в шине возникла проблема с конфигурацией, так как модуль вставлен в неправильное положение.
				FALSE – шина сконфигурирована правильно.

7. ОБСЛУЖИВАНИЕ

	1	ABSENT MODULES	bAbsentModules	TRUE — один или несколько объявленных модулей отсутствуют.
				FALSE — в шине обнаружены все заявленные модули.
	2	SWAPPED MODULES	bSwappedModules	TRUE – в шине есть измененные модули.
				FALSE – в шине нет измененных модулей.
	3	NON-DECLARED MODULES	bNonDeclaredModules	TRUE — один или несколько модулей на шине не были объявлены в конфигурации.
				FALSE — объявлены все модули.
	4	MODULES W/ DIAGNOSTICS	bModulesWithDiagnostic	TRUE – один или несколько модулей на шине имеют активную диагностику.
		FALSE – в модулях на шине нет активной диагностики.		
5	MODULES W/ FATAL ERROR	bModuleFatalError	TRUE — один или несколько модулей на шине находятся в фатальной ошибке.	
			FALSE – Все модули работают правильно.	
6	MODULES W/ PARAM. ERROR	bModuleParameterError	TRUE – Один или несколько модулей на шине имеют ошибку параметризации.	
			FALSE – Все модули параметризованы.	
7	BUS ERROR	bWHSBBusError	TRUE – Индикация ведущего устройства о сбое в шине WHSB.	
			FALSE – Шина WHSB работает правильно.	
%QB(n+1)	0	HARDWARE FAILURE	bHardwareFailure	TRUE — аппаратный сбой процессора.
				FALSE — оборудование работает нормально.
	1	SOFTWARE EXCEPTION	bSoftwareException	TRUE — одно или несколько исключений, созданных программным обеспечением.
				FALSE — исключения не генерируются в программном обеспечении.
	2	HARDWARE WATCHDOG	bHwWatchdogReset	TRUE — ЦП хотя бы один раз был перезапущен аппаратным сторожевым таймером.
				FALSE — ЦП работает нормально.
	3	ERROR IN MEMORY CARD	bMemoryCardError	TRUE – карта памяти вставлена в ЦП, но не работает должным образом.

7. ОБСЛУЖИВАНИЕ

			FALSE — карта памяти работает правильно.
4	COM1 CONF. ERROR	bCOM1ConfigError	TRUE — произошла ошибка во время или после настройки последовательного интерфейса COM 1.

Прямая переменная		Сообщение диагностики	AT переменная DG_Module.tSummarized.*	Описание
Переменная	Бит			
				FALSE — конфигурация последовательного интерфейса COM 1 верна.
	5	COM2 CONF. ERROR	bCOM2ConfigError	TRUE — произошла ошибка во время или после настройки последовательного интерфейса COM 2.
				FALSE — конфигурация последовательного интерфейса COM 2 верна.
	6	NET1 CONF. ERROR	bNET1ConfigError	TRUE — произошла ошибка во время или после настройки интерфейса NET 1 Ethernet.
				FALSE — конфигурация интерфейса Ethernet NET 1 верна.
	7	NET2 CONF. ERROR	bNET2ConfigError	TRUE — произошла ошибка во время или после настройки интерфейса NET 2 Ethernet.
				FALSE — конфигурация интерфейса Ethernet NET 2 верна.
%QB(n+2)	0	INVALID DATE/TIME	bInvalidDateTime	TRUE — дата или час недействительны.
				FALSE — дата и час указаны правильно.
	1	RUNTIME RESET	bRTSReset	TRUE — RTS (система выполнения) была перезапущена хотя бы один раз. Эта диагностика сбрасывается только при перезапуске системы.

7. ОБСЛУЖИВАНИЕ

				FALSE — RTS (система выполнения) работает нормально.
	2	OTD SWITCH ERROR	bOTDSwitchError	TRUE – Истинно, если ключ OTD был заблокирован более чем на 20 с хотя бы один раз при подаче питания на ЦП. Эта диагностика сбрасывается только при перезапуске системы. FALSE – Ключ в настоящее время не заблокирован или был заблокирован, когда ЦП был включен.
	0	ABSENT RACK	bAbsentRacks	TRUE — одна или несколько заявленных стоек отсутствуют. FALSE – Отсутствующих стоек нет.
%QB(n+3)	1	DUPLICATED RACK	bDuplicatedRacks	TRUE – имеются стойки с повторяющимся идентификационным номером. FALSE – Нет стоек с повторяющимся идентификационным номером.
	2	INVALID RACK	bInvalidRacks	TRUE – имеются стойки с недопустимым идентификационным номером. FALSE – нет стоек с неверным идентификационным номером.
	3	NON DECLARED RACK	bNonDeclaredRacks	TRUE – имеются стойки с незадекларированным идентификационным номером. FALSE – Стойки с незадекларированным идентификационным номером отсутствуют.
	4	DUPLICATED SLOT	bDuplicatedSlots	TRUE- есть несколько повторяющихся адресов слотов.
Прямая переменная	Сообщение диагностики	AT переменная DG_Module.tSummarized.*	Описание	Прямая переменная

7. ОБСЛУЖИВАНИЕ

Переменная	Бит			
				FALSE — нет повторяющихся адресов слотов.

Табл. 222: Сводная диагностика ЦП

Примечания:

Переменная прямого представления: «n» представляет собой значение, установленное в ЦП с помощью программного обеспечения MasterTool IEC XE, например, при диагностике исходного адреса.

Директива AT: В описании символических переменных, которые используют директиву AT для выполнения отображения в переменных с прямой адресацией, перед желаемой сводной диагностикой должен быть указан синтаксис DG_Module.tSummarized, когда слово модуля заменяется используемым ЦП. Например, для диагностики несовместимой конфигурации необходимо использовать переменную: DG_NX3010.tSummarized.bConfigMismatch. Директива AT — это слово, зарезервированное в программном обеспечении для программирования и используемое только для диагностической индикации.

Несоответствие конфигурации: Диагностика несовместимой конфигурации формируется, если один или несколько модулей стойки не соответствуют заявленным, то есть при отсутствии или различных состояниях модулей. Вставленные в шину модули, не заявленные в проекте, не учитываются.

Переставленные модули: Если только два модуля заменены между собой на шине, то можно идентифицировать измененную диагностику. В противном случае проблема трактуется как «Несовместимая конфигурация».

Модули с неустранимой ошибкой: В случае, если диагностика модулей с неустранимой ошибкой верна, необходимо проверить, какой модуль в шине является проблемным, и отправить его в службу технической поддержки Altus, так как он имеет аппаратный сбой.

Модуль с ошибкой параметризации: Если диагностика ошибки параметризации верна, необходимо убедиться, что модуль на шине сконфигурирован правильно, а версия микропрограммы и программного обеспечения MasterTool IEC XE верна. Если проблема возникла при вставке модуля в шину, убедитесь, что модуль поддерживает горячую замену.

Ошибка шины: Считается фатальной ошибкой, прерывающей доступ к модулям в шине. Если диагностика ошибки шины верна, возможно, возникла нештатная ситуация из-за выбранной конфигурации горячего обмена или аппаратная проблема в линиях связи шины, обратитесь в службу технической поддержки Altus.

Аппаратный сбой: если диагностика аппаратного сбоя верна, ЦП необходимо отправить в службу технической поддержки Altus, так как у него есть проблемы с RTC, вспомогательным процессором или другими аппаратными ресурсами.

Программное исключение: если диагностика программного исключения верна, пользователь должен проверить свое приложение, чтобы гарантировать, что оно не обращается к памяти неправильно. Если проблема остается, необходимо обратиться в сектор технической поддержки Altus. Коды исключений программного обеспечения описаны далее в таблице подробной диагностики ЦП.

Диагностическое сообщение: диагностические сообщения можно просмотреть на графическом дисплее ЦП с помощью клавиши OTD или с помощью веб-интерфейса на странице диагностики ЦП.

7.1.5.2. Детальная диагностика

В приведенных ниже таблицах содержится подробная диагностика процессоров серии Nexto. Важно иметь в виду следующие наблюдения, прежде чем обращаться к ним:

Визуализация диагностических структур: Диагностические структуры, добавленные в проект, можно увидеть в элементе «Диспетчер библиотек» древовидного представления MasterTool IEC XE. Там можно увидеть все типы данных, определенные в структуре.

Счетчики: все счетчики диагностики ЦП возвращаются к нулю при превышении их предельного значения.

Переменная прямого представления: «n» представляет собой значение, настроенное в ЦП через MasterTool IEC XE в качестве начального диагностического адреса.

Директива AT: при описании символических переменных, которые используют директиву AT для отображения в переменные прямого отображения, синтаксис, который следует использовать перед желаемой сводной диагностикой, — DG_Module.tDetailed., где слово «модуль» должно быть заменено используемым ЦП. Директива AT является зарезервированным словом программиста, и некоторые символьные переменные, использующие эту директиву, указывают на диагностику.

7. ОБСЛУЖИВАНИЕ

Прямое представительство	Размер	AT переменная DG_Modulo.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 прошивки.
%QB(n+12)	BYTE ARRAY(4)	Target. abyBootloaderVersion	Версия загрузчика.
%QB(n+16)	BYTE ARRAY(4)	Target. abyAuxprocVersion	Версия вспомогательного процессора.

Таблица 223: Описание целевой группы подробной диагностики

Прямое представительство	Размер	AT переменная DG_Modulo.tDetailed.*	Описание
%QX(n+20).0	BIT	Hardware. bAuxprocFailure	Сбой связи между вспомогательным процессором и основным процессором.
%QX(n+20).1	BIT	Hardware. bRTCFailure	Основной процессор не может обмениваться данными с RTC (часами ЦП).
%QX(n+20).2	BIT	Hardware. bThermometerFailure ¹	Сбой связи между термометром и главным процессором.
%QX(n+20).3	BIT	Hardware. bLCDFailure	Сбой связи между ЖК-экраном и главным процессором.

Таблица 224: Описание группы подробной диагностики оборудования

Прямое представительство	Размер	AT переменная DG_Modulo.tDetailed.*	Описание
%QW(n+21)	WORD	Exception. wExceptionCode	Код исключения, сгенерированный RTS. См. Таблицу 226.
%QB(n+23)	BYTE	Exception. byProcessorLoad	Уровень, в процентах (%), заряда процессора.

Table 225: Группа подробной диагностики исключений

Примечание:

Код исключения: код исключения, сгенерированный RTS (Runtime System), можно посмотреть ниже.:

Код	Описание	Код	Описание
0x0000	Кода исключения нет.	0x0051	Нарушение доступа.
0x0010	Время сторожевого таймера задачи IEC истекло (программный сторожевой таймер).	0x0052	Привилегированная инструкция.
0x0012	Ошибка конфигурации ввода/вывода.	0x0053	Сбой страницы.

7. ОБСЛУЖИВАНИЕ

0x0013	Ошибка контрольной суммы после загрузки программы.	0x0054	Переполнение стека.
0x0014	Ошибка полевой шины.	0x0055	Неверное расположение.
0x0015	Ошибка обновления ввода/вывода.	0x0056	Неверный маневр.
0x0016	Время цикла (выполнение) превышено.	0x0057	Защищенная страница.
0x0017	Онлайн-обновление программы слишком долгое.	0x0058	Двойной провал.
0x0018	Внешние ссылки не разрешены.	0x0059	Неверный опкод.
Код	Описание	Код	Описание
0x0019	Загрузка отклонена.	0x0100	Несоответствие типа данных.
0x001A	Проект не загружен, так как сохраняемые переменные не могут быть перераспределены.	0x0101	Превышен лимит массивов.
0x001B	Проект не загружен и удален.	0x0102	Деление на ноль.
0x001C	Недостаточно памяти.	0x0103	Переполнение.
0x001D	Ретенционная память повреждена и не может быть сопоставлена.	0x0104	Нельзя продолжать.
0x001E	Проект может быть загружен, но позже вызывает падение.	0x0105	Сторожевой таймер загрузки процессора для всех обнаруженных задач IEC.
0x0021	Цель запускаемого приложения не соответствует текущей цели.	0x0150	FPU: не указана ошибка.
0x0022	Ошибка запланированных задач.	0x0151	FPU: операнд не является нормальным.
		0x0152	FPU: деление на ноль.
0x0023	Скачанный файл Checksum с ошибкой.	0x0153	FPU: Неточный результат.
0x0024	Сохраняемый идентификатор не соответствует текущему идентификатору программы загрузочного проекта.	0x0154	FPU: неверная операция.
0x0025	Сбой конфигурации задачи IEC.	0x0155	FPU: переполнение.
0x0026	Приложение работает с неправильной целью.	0x0156	FPU: проверка стека.
0x0050	Незаконная инструкция.	0x0157	FPU: недолив.

Таблица 226: Коды исключений RTS

Прямое представительство	Размер	AT DG vari-able_Modulo.tDetailed.*	Описание
--------------------------	--------	------------------------------------	----------

7. ОБСЛУЖИВАНИЕ

%QB(n+25)	BYTE	RetainInfo. byCPUInitStatus	Статус запуска процессора: 01: Горячий старт 02: Теплый старт 03: Холодный пуск Примечание. Эти переменные сбрасываются при каждом включении питания.
%QW(n+26)	WORD	RetainInfo. wCPUColdStartCounter	Счетчик холодного запуска: он будет увеличиваться только из-за горячего удаления ЦП на шине, а не из-за команды холодного сброса MasterTool IEC XE. (от 0 до 65535)
%QW(n+28)	WORD	RetainInfo. wCPUWarmStartCounter	Счетчик теплого запуска: он будет увеличиваться только во время последовательности включений питания системы, а не из-за команды Hot Reset MasterTool IEC XE. (от 0 до 65535).
%QW(n+30)	WORD	RetainInfo. wCPULotStartCounter	Счетчик помех меньше, чем время поддержки сбоя питания ЦП. (от 0 до 65535).
%QW(n+32)	WORD	RetainInfo. wRTSResetCounter	Счетчик сбросов, выполненных RTS (Runtime System). (от 0 до 65535).

Таблица 227: Подробная диагностика группы RetainInfo

Прямое представительство	Размер	AT DG variable_Modulo.tDetailed.*	Описание
%QX(n+36).0	BIT	Reset. bBrownOutReset	ЦП был перезапущен из-за отказа источника питания при последнем запуске.
Прямое представительство	Размер	AT DG variable_Modulo.tDetailed.*	Описание
%QX(n+36).1	BIT	Reset. bWatchdogReset	ЦП был перезапущен из-за активного сторожевого таймера при последнем запуске.

Таблица 228: Группа подробной диагностики сброса Описание

Примечание:

Сброс пониженного напряжения: Диагностика сброса пониженного напряжения верна только тогда, когда источник питания превышает минимальный предел, требуемый его техническими характеристиками, оставаясь низковольтным, т. е. не подвергаясь прерыванию. ЦП идентифицирует падение напряжения питания и выдаст диагностическое сообщение о сбое питания. Когда напряжение восстанавливается, ЦП автоматически перезагружается и отображает диагностику сброса пониженного напряжения.

Прямое представительство	Размер	AT DG variable_Modulo.tDetailed.*	Описание
%QX(n+37).0	BIT	Thermometer. bOverTemperatureAlarm ¹	Аварийный сигнал сработал из-за внутренней температуры 85 °C или выше.
%QX(n+37).1	BIT	Thermometer. bUnderTemperatureAlarm ¹	Аварийный сигнал генерируется из-за внутренней температуры 0°C или ниже.
%QD(n+38)	DINT	Thermometer. diTemperature ¹	Температура, считываемая внутренним датчиком ЦП.

Таблица 229: Описание группы подробной диагностики термометра

7. ОБСЛУЖИВАНИЕ

Примечание:

Температура: Чтобы увидеть температуру непосредственно в адресе памяти, необходимо сделать преобразование, так как размер данных DINT и мониторинг ведется в 4 байта. Поэтому рекомендуется использовать связанную символическую переменную, так как она уже обеспечивает конечное значение температуры.

Прямое представительство	Размер	AT DG Variable_Module.tDetailed.*	Описание
%QB(n+42)	BYTE	Serial.COM1. byProtocol	Протокол, выбранный в COM 1: 00: Без протокола 01: Ведущее устройство MODBUS RTU 02: Ведомое устройство MODBUS RTU 03: Другой протокол
%QD(n+43)	DWORD	Serial.COM1. dwRXBytes	Счетчик символов, полученных от 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 в следующих случаях: Энергетика Конфигурация 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	

Табл. 230: Описание группы подробной диагностики Serial COM 1

Примечание:

Счетчик ошибок четности: когда последовательный COM 1 настроен без четности, этот счетчик ошибок не будет увеличиваться при получении сообщения с другой четностью. В этом случае будет указана ошибка кадра.

Прямая переменная	Размер	AT DG Variable_Module.tDetailed.*	Описание
%QB(n+67)	BYTE	Serial.COM2. byProtocol	Протокол, выбранный в COM 2: 00: Без протокола 01: Ведущее устройство MODBUS RTU 02: Ведомое устройство MODBUS RTU 03: Другой протокол
%QD(n+68)	DWORD	Serial.COM2. dwRXBytes	Счетчик символов, полученных от COM 2 (от 0 до 4294967295).
%QD(n+72)	DWORD	Serial.COM2. dwTXBytes	Счетчик символов, передаваемых через COM 2 (от 0 до 4294967295).
%QW(n+76)	WORD	Serial.COM2. wRXPendingBytes	Количество символов, оставшихся в буфере чтения в COM 2 (от 0 до 65535).
%QW(n+78)	WORD	Serial.COM2. wTXPendingBytes	Количество символов, оставшихся в буфере передачи в COM 2 (от 0 до 65535).

7. ОБСЛУЖИВАНИЕ

%QW(n+80)	WORD	Serial.COM2. wBreakErrorCounter	Счетчики ошибок COM 2 (от 0 до 65535). Эти счетчики перезапускаются при следующих условиях: Запуск Конфигурация последовательного порта COM 2 Удаление очередей RX и TX
%QW(n+82)	WORD	Serial.COM2. wParityErrorCounter	
%QW(n+84)	WORD	Serial.COM2. wFrameErrorCounter	
%QW(n+86)	WORD	Serial.COM2. wRXOverrunCounter	

Табл. 231: Группа подробной диагностики Serial COM 2

Примечание:

Счетчик ошибок четности: когда последовательный COM 2 настроен без четности, этот счетчик ошибок не будет увеличиваться при получении сообщения с другой четностью. В этом случае будет указана ошибка кадра.

Прямое представление	Размер	AT DG variable_Module.tDetailed.*	Описание
%QX(n+92).0	BIT	Ethernet.NET1. bLinkDown	Указывает состояние канала в сети NET 1.
%QW(n+93)	WORD	Ethernet.NET1. wProtocol	Протокол выбран в NET 1: 00: Нет протокола
%QX(n+93).0	BIT	Ethernet.NET1. wProtocol. bMODBUS_RTU_ETH_Client	Клиент MODBUS RTU через TCP.
%QX(n+93).1	BIT	Ethernet.NET1. wProtocol. bMODBUS_ETH_Client	Клиент MODBUS TCP.
%QX(n+93).2	BIT	Ethernet.NET1. wProtocol. bMODBUS_RTU_ETH_Server	Сервер MODBUS RTU через TCP.
%QX(n+93).3	BIT	Ethernet.NET1. wProtocol. bMODBUS_ETH_Server	Сервер MODBUS TCP.
%QB(n+95)	STRING (15)	Ethernet.NET1. szIP	NET IP-адрес 1.
%QB(n+111)	STRING (15)	Ethernet.NET1. szMask	NET 1 Маска подсети.
%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	NET IP-адрес 1.
%QB(n+165)	BYTE ARRAY(4)	Ethernet.NET1. abyMask	NET 1 Маска подсети.
%QB(n+169)	BYTE ARRAY(4)	Ethernet.NET1. abyGateway	Адрес шлюза NET 1.
Прямая переменная	Размер	AT DG_Module.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).

7. ОБСЛУЖИВАНИЕ

%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	Количество байтов, отправленных через порт NET 1 (от 0 до
%QD(n+191)	DWORD	Ethernet.NET1. dwBytesReceived	4294967295).
%QW(n+195)	WORD	Ethernet.NET1. wTXErrors	Количество байтов, полученных через порт NET 1 (от 0 до 4294967295).
%QW(n+197)	WORD	Ethernet.NET1. wTXFIFOErrors	Счетчик ошибок передачи через порт NET 1 (от 0 до 65535).
%QW(n+199)	WORD	Ethernet.NET1. wTXDropErrors	Счетчик ошибок в буфере передачи через порт NET 1 (от 0 до 65535).
%QW(n+201)	WORD	Ethernet.NET1. wTXCollisionErrors	Счетчик потери соединения при передаче через порт NET 1 (от 0 до 65535).
%QW(n+203)	WORD	Ethernet.NET1. wTXCarrierErrors	Счетчик ошибок коллизий при передаче через порт NET 1 (от 0 до 65535).
%QW(n+205)	WORD	Ethernet.NET1. wRXErrors	Счетчик ошибок передачи при передаче через порт NET 1 (от 0 до 65535).
%QW(n+207)	WORD	Ethernet.NET1. wRXFIFOErrors	Счетчик ошибок, полученных через порт NET 1 (от 0 до 65535).
%QW(n+209)	WORD	Ethernet.NET1. wRXDropErrors	Счетчик ошибок в приемном буфере через порт NET 1 (от 0 до 65535).
%QW(n+211)	WORD	Ethernet.NET1. wRXFrameErrors	Счетчик потери соединения при приеме через порт NET 1 (от 0 до 65535).
%QW(n+213)	WORD	Ethernet.NET1. wMulticast	Счетчик ошибок кадров при приеме через порт NET 1 (от 0 до 65535).

Таблица 232: Подробная диагностика группы Ethernet NET1

Прямая переменная	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QX(n+219).0	BIT	Ethernet.NET2. bLinkDown	Указывает состояние ссылки в NET 2.
%QW(n+220)	WORD	Ethernet.NET2. wProtocol	Протокол выбран в NET 2: 00: Нет протокола
%QX(n+220).0	BIT	Ethernet.NET2. wProtocol. bMODBUS_RTU_ETH_Client	Клиент MODBUS RTU через TCP.
%QX(n+220).1	BIT	Ethernet.NET2. wProtocol. bMODBUS_ETH_Client	Клиент MODBUS TCP.
%QX(n+220).2	BIT	Ethernet.NET2. wProtocol. bMODBUS_RTU_ETH_Server	Сервер MODBUS RTU через TCP.
%QX(n+220).3	BIT	Ethernet.NET2. wProtocol. bMODBUS_ETH_Server	Сервер MODBUS TCP.
%QB(n+222)	STRING (15)	Ethernet.NET2. szIP	NET2 IP-адрес.

7. ОБСЛУЖИВАНИЕ

%QB(n+238)	STRING (15)	Ethernet.NET2. szMask	Маска подсети NET 2.
%QB(n+254)	STRING (15)	Ethernet.NET2. szGateway	Адрес шлюза NET 2.
Прямая переменная	Размер	AT DG_Module.tDetailed.*	Variable Описание
%QB(n+270)	STRING (17)	Ethernet.NET2. szMAC	MAC-адрес NET 2.
%QB(n+288)	BYTE ARRAY(4)	Ethernet.NET2. abyIP	NET2 IP-адрес.
%QB(n+292)	BYTE ARRAY(4)	Ethernet.NET2. abyMask	Маска подсети NET 2.
%QB(n+296)	BYTE ARRAY(4)	Ethernet.NET2. abyGateway	Адрес шлюза NET 2.
%QB(n+300)	BYTE ARRAY(6)	Ethernet.NET2. abyMAC	MAC-адрес NET 2.
%QD(n+306)	DWORD	Ethernet.NET2. dwPacketsSent	Счетчик пакетов, отправленных через порт NET 2 (от 0 до 4294967295).
%QD(n+310)	DWORD	Ethernet.NET2. dwPacketsReceived	Счетчик пакетов, полученных через порт NET 2 (от 0 до 4294967295).
%QD(n+314)	DWORD	Ethernet.NET2. dwBytesSent	Количество байтов, отправленных через порт NET 2 (от 0 до
%QD(n+318)	DWORD	Ethernet.NET2. dwBytesReceived	4294967295).
%QW(n+322)	WORD	Ethernet.NET2. wTXErrors	Количество байтов, полученных через порт NET 2 (от 0 до 4294967295).
%QW(n+324)	WORD	Ethernet.NET2. wTXFIFOErrors	Счетчик ошибок передачи через порт NET 2 (от 0 до 65535).
%QW(n+326)	WORD	Ethernet.NET2. wTXDropErrors	Счетчик ошибок в буфере передачи через порт NET 2 (от 0 до 65535).
%QW(n+328)	WORD	Ethernet.NET2. wTXCollisionErrors	Счетчик потери соединения при передаче через порт NET 2 (от 0 до 65535).
%QW(n+330)	WORD	Ethernet.NET2. wTXCarrierErrors	Счетчик ошибок коллизии при передаче через порт NET 2 (от 0 до 65535).
%QW(n+332)	WORD	Ethernet.NET2. wRXErrors	Счетчик ошибок передачи при передаче через порт NET 2 (от 0 до 65535).
%QW(n+334)	WORD	Ethernet.NET2. wRXFIFOErrors	Счетчик ошибок, полученных через порт NET 2 (от 0 до 65535).
%QW(n+336)	WORD	Ethernet.NET2. wRXDropErrors	Счетчик ошибок в приемном буфере через порт NET 2 (от 0 до 65535).
%QW(n+338)	WORD	Ethernet.NET2. wRXFrameErrors	Счетчик потери соединения при приеме через порт NET 2 (от 0 до 65535).
%QW(n+340)	WORD	Ethernet.NET2. wMulticast	Счетчик ошибок кадров при приеме через порт NET 2 (от 0 до 65535).

Табл. 233: Подробная диагностика группы Ethernet NET2

Прямая переменная	Размер	AT	Variable	Описание
-------------------	--------	----	----------	----------

7. ОБСЛУЖИВАНИЕ

		DG_Module.tDetailed.*	
%QB(n+346)	BYTE	UserFiles. byMounted	Указывает, может ли память, используемая для записи пользовательских файлов, принимать данные.
%QD(n+347)	DWORD	UserFiles. dwFreeSpacekB	Свободное место в памяти файлов пользователя в килобайтах.
%QD(n+351)	DWORD	UserFiles. dwTotalSizekB	Емкость памяти файлов пользователя в Кбайтах.

Таблица 234: Файлы пользователя группы подробной диагностики

Примечание:

Пользовательский раздел: Пользовательский раздел — это область памяти, зарезервированная для хранения данных в ЦП. Например: файлы с расширением PDF, файлы с расширением DOC и другие данные.

Прямая переменная	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QB(n+356)	BYTE	UserLogs. byMounted	Состояние памяти, в которую вставлены пользовательские журналы.
%QW(n+357)	WORD	UserLogs. wFreeSpacekB	Свободное место в памяти журнала пользователя в килобайтах.
%QW(n+359)	WORD	UserLogs. wTotalSizekB	Пользователь регистрирует объем памяти в килобайтах.

Таблица 235: Журналы пользователей группы подробной диагностики

Прямая переменная	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QB(n+362)	BYTE	MemoryCard. byMounted	Состояние карты памяти: 00: карта памяти не установлена 01: Карта памяти вставлена и установлена
%QX(n+363).0	BIT	MemoryCard. bMemcardtoCPUвключено	Уровень защиты карты памяти: Чтение данных с карты памяти авторизованным процессором.
%QX(n+363).1	BIT	MemoryCard. bCPUtoMemcardвключено	Запись данных на карту памяти авторизованным ЦП.
%QD(n+364)	DWORD	MemoryCard. dwFreeSpacekB	Свободное место на карте памяти в килобайтах.
%QD(n+368)	DWORD	MemoryCard. dwTotalSizekB	Емкость карты памяти в килобайтах

Таблица 236: Группа подробной диагностики карты памяти

Прямая переменная	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QB(n+372)	BYTE	WHSB. byHotSwapAndStartupStatus	Сообщает о нештатной ситуации в шине, вызвавшей остановку приложения для каждого режима горячей замены. См. Таблицу 238 для получения дополнительной информации.

7. ОБСЛУЖИВАНИЕ

%QB(n+373)	BYTE	WHSB. byReserved_0	Сдержанный.
%QD(n+374)	DWORD ARRAY (32)	WHSB. adwRackIOErrorStatus	Идентификация ошибок в модулях ввода/вывода по отдельности. Дополнительные сведения об этой диагностике см. в примечаниях ниже.
%QD(n+502)	DWORD ARRAY (32)	WHSB. adwModulePresenceStatus	Статус наличия заявленных модулей ввода/вывода в шинах, по отдельности. Дополнительные сведения об этой диагностике см. в примечаниях ниже.
%QB(n+630)	BYTE	WHSB. byWHSBBusErrors	Счетчик отказов в шине WHSB. Этот счетчик перезапускается при подаче питания (от 0 до 255).

Таблица 237: Группа подробной диагностики WHSB

Примечания:

Диагностика ошибок модулей шины: Каждое двойное слово из этого диагностического массива представляет собой стойку, позиции которой представлены битами этих двойных слов. Итак, Bit-0 DWORD-0 эквивалентен нулевой позиции стойки с нулевым адресом. Каждый из этих битов является результатом логической операции ИЛИ между диагностикой несовместимой конфигурации (bConfigMismatch), отсутствующими модулями (bAbsentModules), замененными модулями (bSwappedModules), модулем с фатальной ошибкой (bModuleFatalError) и рабочим состоянием модуля в определенное положение.

Состояние присутствия модуля: Каждое DWORD из этого диагностического массива представляет собой стойку, позиции которой представлены битами этих DWORD. Итак, бит-0 из DWORD-0 эквивалентен нулевой позиции стойки с нулевым адресом. Таким образом, если модуль присутствует, этот бит будет истинным. Важно отметить, что эта диагностика действительна для всех модулей, кроме блоков питания, ЦП и незадекларированных модулей, например, те, которые не находятся в стойке на соответствующей позиции (бит остается в ложном состоянии).

Ситуации, в которых приложение останавливается: Коды для возможных ситуаций, в которых приложение останавливается, можно посмотреть ниже:

Код	Исчисляемый	Описание
00	INITIALIZING	Это состояние представлено, пока другие состояния не готовы.
01	RESET_WATCHDOG	Приложение в режиме остановки из-за сброса аппаратного сторожевого таймера или сброса среды выполнения, когда параметр «Запустить пользовательское приложение после сброса сторожевого таймера» не отмечен.
02	ABSENT_MODULES_HOT_SWAP_ ВЫКЛЮЧЕНО	Приложение в режиме остановки из-за установки диагностики отсутствия модулей, когда для режима горячей замены установлено значение «Отключено» или «Отключено, только для объявленных модулей».
03	CFG_MISMATCH_HOT_SWAP_ ВЫКЛЮЧЕНО	Приложение в режиме остановки из-за установки диагностики несоответствия конфигурации, когда режим горячей замены «Отключен» или «Отключен, только для объявленных модулей».
04	ABSENT_MODULES_HOT_SWAP_ STARTUP_CONSISTENCY	Приложение в режиме остановки из-за установки диагностики «Отсутствующие модули», когда для режима «горячей» замены установлено значение «Включено, с согласованностью запуска» или

7. ОБСЛУЖИВАНИЕ

		«Включено, с согласованностью запуска только для объявленных модулей».
05	CFG_MISMATCH_HOT_SWAP_STARTUP_CONSISTENCY	Приложение в режиме остановки из-за установки диагностики несовместимой конфигурации, когда для режима горячей замены установлено значение «Включено, с согласованностью запуска» или «Включено, с согласованностью запуска только для объявленных модулей».
06	APPL_STOP_ALLOWED_TO_RUN	Приложение в режиме остановки, и все согласования выполнены успешно. Приложение можно перевести в режим работы.
07	APPL_STOP_MODULES_NOT_READY	Приложение в режиме остановки и все согласования выполнены успешно, но модули ввода-вывода не могут запустить систему. Невозможно перевести приложение в режим выполнения.
08	APPL_STOP_MODULES_GETTING_READY_TO_RUN	Приложение в режиме остановки, и все согласования выполнены успешно. Модули ввода-вывода готовятся к запуску системы. Невозможно перевести приложение в режим выполнения.
09	NORMAL_OPERATING_STATE	Приложение в режиме выполнения.
10	MODULE_CONSISTENCY_OK	Внутреннее использование.
11	APPL_STOP_DUE_TO_EXCEPTION	Приложение в режиме остановки из-за исключения в ЦП.
12	DUPLICATED_SLOT_HOT_SWAP_ВЫКЛЮЧЕНО	Приложение в режиме остановки из-за установки диагностики дублированных слотов, когда режим горячей замены «Отключен» или «Отключен, только для объявленных модулей».
13	DUPLICATED_SLOT_HOT_SWAP_STARTUP_CONSISTENCY	Приложение в режиме остановки из-за настройки диагностики дублированных слотов, когда для режима горячей замены установлено значение «Включено, с согласованностью запуска» или «Включено, с согласованностью запуска только для объявленных модулей».
14	DUPLICATED_SLOT_HOT_SWAP_ВКЛЮЧЕНО	Приложение в режиме остановки из-за установки диагностики дублированных слотов, когда режим горячей замены «Включен, без согласованности при запуске».
15	NON_DECLARED_MODULE_HOT_SWAP_STARTUP_CONSISTENCY	Приложение в режиме остановки из-за установки диагностики недеklarированных модулей, когда режим горячей замены «включен, с согласованностью запуска».
16	NON_DECLARED_MODULE_HOT_SWAP_ВЫКЛЮЧЕНО	Приложение в режиме остановки из-за установки диагностики «Недекларированные модули», когда режим «горячей замены» отключен.

Таблица 238: Коды ситуаций, при которых приложение останавливается

7. ОБСЛУЖИВАНИЕ

Прямая переменная	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QB(n+631)	BYTE	Application. byCPUState	Информирует о рабочем состоянии ЦП: 01: все пользовательские приложения находятся в режиме выполнения. 03: Все пользовательские приложения находятся в режиме остановки.
%QX(n+632).0	BIT	Application. bForcedIOs	Имеется одна или несколько точек принудительного ввода/вывода.

Таблица 239: Описание группы подробной диагностики приложения

Прямая переменная	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QX(n+633).0	BIT	SNTP. bServiceВключено	Служба SNTP включена.
%QB(n+634)	BYTE	SNTP. byActiveTimeServer	Указывает, какой сервер активен: 00: Нет активного сервера. 01: Основной сервер активен. 02: Дополнительный сервер активен.
%QW(n+635)	WORD	SNTP. wPrimaryServerDownCount	Количество раз, когда основной сервер был недоступен (от 0 до 65535).
%QW(n+637)	WORD	SNTP. wSecondaryServerDownCount	Количество раз, когда дополнительный сервер был недоступен (от 0 до 65535).
%QD(n+639)	DWORD	SNTP. dwRTCtimeUpdatedCount	Количество обновлений RTC службой SNTP (от 0 до 4294967295).
%QB(n+643)	BYTE	SNTP. byLastUpdateSuccessful	Указывает статус последнего обновления: 00: Не обновляется. 01: Последнее обновление не удалось. 02: Последнее обновление прошло успешно.
%QB(n+644)	BYTE	SNTP. byLastUpdateTimeServer	Указывает, какой сервер использовался при последнем обновлении: 00: Нет обновлений. 01: Основной сервер. 02: Дополнительный сервер.
%QB(n+645)	BYTE	SNTP.sLastUpdateTime. byDayOfMonth	День последнего обновления RTC.
%QB(n+646)	BYTE	SNTP.sLastUpdateTime. byMonth	Месяц последнего обновления RTC.
%QW(n+647)	WORD	SNTP.sLastUpdateTime. wYear	Год последнего обновления RTC.
%QB(n+649)	BYTE	SNTP.sLastUpdateTime. byHours	Время последнего обновления RTC.
%QB(n+650)	BYTE	SNTP.sLastUpdateTime. byMinutes	Минута последнего обновления RTC.
%QB(n+651)	BYTE	SNTP.sLastUpdateTime. bReservedAlign	Зарезервировано для выравнивания.
%QB(n+652)	BYTE	SNTP.sLastUpdateTime. bySeconds	Согласно последнему обновлению RTC.
%QW(n+653)	WORD	SNTP.sLastUpdateTime. wMilliseconds	Миллисекунды последнего обновления RTC.

Таблица 240: Подробная диагностика группы SNTP

7. ОБСЛУЖИВАНИЕ

Прямая переменная	Размер	AT DG_Module.tDetailed.* Variable	Описание
%QX(n+659).0	BIT	SOE[1]. bConnectionStatus	Статус подключения клиента 01
%QX(n+659).1	BIT	SOE[1]. bOverflowStatus	Статус очереди событий клиента 01: FALSE — нет переполнения TRUE – превышен предел очереди
%QB(n+660)	BYTE	SOE[1]. byReserved_0	Зарезервирован
Прямая переменная	Размер	AT DG_Module.tDetailed.* Variable	Описание
%QW(n+661)	WORD	SOE[1]. wEventsCounter	Счетчик событий очереди клиентов 01
%QX(n+663).0	BIT	SOE[2]. bConnectionStatus	Статус подключения клиента 02
%QX(n+663).1	BIT	SOE[2]. bOverflowStatus	Статус очереди событий клиента 02: FALSE — Нет переполнения. TRUE – превышен предел очереди
%QB(n+664)	BYTE	SOE[2]. byReserved_0	Зарезервирован.
%QW(n+665)	WORD	SOE[2]. wEventsCounter	Счетчик событий клиентской очереди 02.

Таблица 241: Подробная диагностика группы SOE

Примечания:

Синхронизация диагностики группы SOE в системе, работающей с резервированием полукластера: если проект настроен с резервированием полукластера, диагностика группы SOE не синхронизируется между двумя полукластерами.

Обновление диагностики группы SOE при переходе в активное состояние: когда полукластер переходит из состояния ожидания в активное состояние, диагностика группы SOE обновляется с третьего цикла.

Прямая переменная	Размер	AT DG_Module.tDetailed.* Variable	Описание
%QD(n+667)	DWORD	Rack. dwAbsentRacks	Каждый бит представляет собой идентификационный номер стойки, если какой-либо бит равен TRUE, это означает, что стойка с таким идентификационным номером отсутствует.
%QD(n+671)	DWORD	Rack. dwDuplicatedRacks	Каждый бит представляет собой идентификационный номер стойки. Если какой-либо бит равен TRUE, это означает, что более одной стойки настроены с одним и тем же идентификационным номером.
%QD(n+675)	DWORD	Rack. dwNonDeclaredRacks	Каждый бит представляет собой идентификационный номер стойки, если какой-либо бит равен TRUE, это означает, что имеется стойка, сконфигурированная с идентификационным номером, не объявленным в проекте.

Табл. 242: Подробная диагностика группы стоек

7. ОБСЛУЖИВАНИЕ

Прямая переменная	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QD(n+681)	DWORD	ApplicationInfo. dwApplicationCRC	32-битный CRC приложения. Когда приложение модифицируется и отправляется в ЦП, вычисляется новый CRC.

Таблица 243: Подробная диагностика группы ApplicationInfo

7.1.6. Диагностика через функциональные блоки

Функциональные блоки позволяют визуализировать некоторые параметры, к которым иначе нельзя получить доступ. Функция расширенной диагностики находится в библиотеке NextoStandard и описана ниже.

7.1.6.1. GetTaskInfo

Эта функция возвращает информацию о задаче конкретного приложения.



Рисунок 196: Функция GetTaskInfo

Ниже описаны параметры, которые необходимо передать функции, чтобы она вернула информацию о приложении.

Входной параметр	Тип	Описание
psAppName	POINTER TO STRING	Имя приложения.
psTaskName	POINTER TO STRING	Название задачи.
pstTaskInfo	POINTER TO stTask-Info	Указатель для получения информации о приложении.

Таблица 244: Входные параметры GetTaskInfo

Данные, возвращаемые функцией через указатель, указанный во входных параметрах, описаны в таблице ниже.

Возвращаемые параметры	Размер	Описание
dwCurScanTime	DWORD	Время цикла задачи (выполнение) с разрешением 1 мкс.
dwMinScanTime	DWORD	Минимальное время цикла задачи с разрешением 1 мкс.
dwMaxScanTime	DWORD	Максимальное время цикла задачи. Разрешение 1 мкс.
dwAvgScanTime	DWORD	Среднее время цикла задачи с разрешением 1 мкс.
dwLimitMaxScan	DWORD	Максимальное время цикла задачи до появления сторожевого таймера.
dwIECCycleCount	DWORD	Счетчик циклов IEC.

Таблица 245: Возвращаемые параметры GetTaskInfo

Возможный КОД ОШИБКИ:

NoError: успешное выполнение;

TaskNotPresent: нужной задачи не существует.

7. ОБСЛУЖИВАНИЕ

Пример использования на языке ST:

```
PROGRAM UserPrg VAR sAppName : STRING;
psAppName : POINTER TO STRING; sTaskName :
STRING; psTaskName : POINTER TO STRING;
pstTaskInfo : POINTER TO stTaskInfo;
TaskInfo : stTaskInfo;
Info : ERRORCODE;
END_VAR //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 названием задачи.
pstTaskInfo := ADR(TaskInfo); //Указатель, который получает информацию о задаче.
//ФУНКЦИЯ:
//Вызов функции.
Информация := GetTaskInfo (psAppName, psTaskName, pstTaskInfo); //Variable Info получает
возможные ошибки функции.
```

7.2. Графический дисплей

Графический дисплей, доступный в этом продукте, имеет важный инструмент для управления процессом, так как через него можно распознать возможные состояния ошибки, активные компоненты или наличие диагностики. Кроме того, вся диагностика, включая модули ввода/вывода, представлена пользователю через графический дисплей. Дополнительные сведения об использовании диагностического ключа и его визуализации см. в разделе «Диагностика одним касанием».

На рисунке ниже можно увидеть доступные символы на графическом дисплее этого продукта, а затем их соответствующие значения.

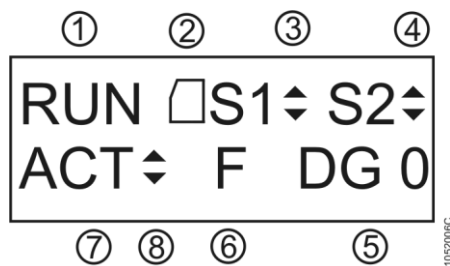


Рис. 197: Экран состояния ЦП

Расшифровка:

1. Индикация работы состояния ЦП. В случае, если приложение CPU работает, состояние RUN. В случае, если приложение ЦП остановлено, состояние будет STOP, а когда оно остановлено в метке очистки приложения, состояние будет BRKP. Дополнительные сведения см. в разделе «Рабочие состояния процессора».
2. Индикация наличия карты памяти. Дополнительные сведения об установке см. в разделе «Установка карты памяти».
3. Индикация трафика COM 1. Стрелка вверх (▲) указывает на передачу данных, а стрелка вниз (▼) указывает на прием данных. Дополнительную информацию об интерфейсе COM 1 см. в разделе «Последовательные интерфейсы».

7. ОБСЛУЖИВАНИЕ

4. Индикация трафика COM 2. Стрелка вверх (▲) указывает на передачу данных, а стрелка вниз (▼) указывает на прием данных. Дополнительную информацию об интерфейсе COM 2 см. в разделе «Последовательные интерфейсы».

5. Индикация количества активных диагностик CPU. Если отображаемое число отличается от 0 (нуля), в ЦП выполняется активная диагностика. Подробнее об их отображении на графическом дисплее ЦП с помощью диагностической клавиши см. в разделе Диагностика одним касанием.

6. Принудительные переменные в индикации CPU. Если на графическом дисплее отображается символ «F», переменная принудительно задается пользователем, будь то символическое, прямое представление или AT. Для получения дополнительной информации о форсировании переменных см. раздел «Запись и форсирование переменных».

7. Идентификация состояния резервирования ЦП (сообщение действительно только для NX3030 в режиме резервирования). Если ЦП является активным ПЛК, будет представлена информация ACT. Другими возможными состояниями являются NCF (не настроено), STR (запуск), INA (неактивно) и SBY (ожидание).

8. Индикация того, что выполняется синхронизация проекта. Стрелка вверх (▲) указывает на передачу данных проекта, а стрелка вниз (▼) указывает на прием данных проекта. Дополнительные сведения о синхронизации проектов см. в разделе «Синхронизация проектов».

Помимо символов, описанных выше, процессоры Nexto могут отображать на графическом дисплее некоторые сообщения, соответствующие процессу, который выполняется в данный момент.

В таблице ниже представлены сообщения и их соответствующие описания.:

Сообщение	Описание
FORMATTING...	Указывает, что процессор форматирует карту памяти.
FORMATTING ERROR	Указывает, что произошла ошибка при форматировании карты памяти процессором.
WRONG FORMAT	Указывает, что формат карты памяти неверен.
INCORRECT PASSWORD	Указывает, что введенный пароль отличается от настроенного пароля.
TRANSFERRING...	Указывает, что проект передается.
Сообщение	Описание
TRANSFERRING ERROR	Указывает на то, что при переносе проекта произошла ошибка, вызванная какой-либо проблемой с картой памяти или ее удалением во время переноса.
TRANSFERRING COMPLETE	Указывает, что передача была выполнена успешно.
TRANSFERRING TIMEOUT	Указывает на тайм-аут (время связи истекло) во время переноса проекта.
CPU TYPE MISMATCH	Указывает, что модель ЦП отличается от модели, настроенной в проекте на карте памяти.
VERSION MISMATCH	Указывает, что версия процессора отличается от версии, настроенной в проекте на карте памяти.
APPLICATION CORRUPTED	Указывает, что приложение на карте памяти повреждено.
APPLICATION NOT FOUND	Указывает, что на карте памяти нет приложения для передачи в ЦП.
CRC NOT FOUND	Указывает, что приложение CRC не существует.
MCF FILE NOT FOUND	Указывает на отсутствие файла MCF на карте памяти.


NO TAG	В MasterTool IEC XE нет сконфигурированного тега для ЦП.
NO DESC	В MasterTool IEC XE нет сконфигурированного описания ЦП.
MSG. ERROR	Указывает на наличие ошибок в диагностических сообщениях запрошенного модуля (модулей).
SIGNATURE MISSING	Указывает, что продукт вызвал непредвиденную проблему. Свяжитесь с сектором технической поддержки Altus.
APP. ERROR RESTARTING	Указывает, что в приложении произошла ошибка, и среда выполнения перезапускает приложение.
APP. NOT LOADED	Указывает, что среда выполнения не будет загружать приложение.
LOADING APP.	Указывает, что среда выполнения загрузит приложение.
WRONG SLOT	Указывает, что ЦП находится в неправильном положении в стойке.
FATAL ERROR	Указывает на наличие серьезных проблем при запуске ЦП, таких как неправильно смонтированные разделы ЦП. Пожалуйста, свяжитесь со службой поддержки Altus.
HW-SW MISMATCH	Указывает, что аппаратное и программное обеспечение ЦП несовместимы, так как продукт вызвал непредвиденную проблему. Пожалуйста, свяжитесь со службой поддержки Altus.
UPDATING FIRMWARE	Указывает, что микропрограмма обновляется в ЦП.
RECEIVING FIRMWARE	Указывает, что файл обновления передается в ЦП.
UPDATED	Показывает обновленную версию микропрограммы в ЦП.
UPDATE ERROR	Указывает, что во время обновления микропрограммы ЦП произошла ошибка, вызванная сбоем связи или проблемами конфигурации.
REBOOTING SYSTEM...	Указывает, что ЦП перезагружается, чтобы обновление вступило в силу.


Таблица 246: Другие сообщения графического дисплея

7.3. Журнал системы

Системный журнал — это функция, доступная в программаторе MasterTool IEC XE. Это важный инструмент для управления процессом, так как он позволяет находить события на ЦП, которые могут указывать на состояние ошибки, наличие активных компонентов или активную диагностику. Такие события можно просматривать в хронологическом порядке с точностью до миллисекунд, с объемом хранения до одной тысячи записей журнала, хранящихся во внутренней памяти ЦП, которые невозможно удалить.

Чтобы получить доступ к этим журналам, просто перейдите в дерево устройств и дважды щелкните устройство, затем перейдите на вкладку «Журнал», где можно увидеть сотни операций, таких как: максимальное количество циклов задач, доступ пользователей, онлайн-изменения, загрузка приложений. и загрузка, синхронизация приложений между ЦП, обновление прошивки между другими событиями и действиями.

Чтобы просмотреть журналы, просто нужно подключиться к ЦП (выбран активный путь) и нажать на . При нажатии этой кнопки журналы отображаются и мгновенно обновляются. Когда кнопка не нажата, журналы будут удерживаться на экране, это означает, что эта кнопка имеет две стадии: одна удерживает состояние журнала при

обновлении, а состояние обновления отключено. Чтобы журналы больше не отображались, нажмите .

7. ОБСЛУЖИВАНИЕ

Журналы можно фильтровать по четырем типам: предупреждения, ошибки, исключения и информация.

Еще один способ отфильтровать сообщения, отображаемые пользователю, — это выбрать компонент для просмотра.

Отметка времени на вкладке «Журнал» отображается MasterTool после информации, предоставленной устройством (ЦП). MasterTool может отображать отметку времени по местному времени (время компьютера) или по Гринвичу, если установлен флажок времени UTC.

ВНИМАНИЕ

Если параметры времени или часового пояса устройства неверны, отметка времени, отображаемая в Master-Tool, также не будет правильной.

Для получения дополнительной информации о системных журналах см. Руководство пользователя MasterTool IEC XE — MU299609 и подраздел «Часы RTC и синхронизация времени» данного руководства.

ВНИМАНИЕ

Системные журналы ЦП серии Nexto, начиная с версии прошивки 1.4.0.33, перезагружаются в случаях перезапуска ЦП или перезагрузки системы выполнения, то есть можно будет проверить более старые журналы, когда один таких ситуаций возникает.

7.4. Не загружается приложение при запуске

1. При необходимости пользователь может не загружать существующее приложение на ЦП во время его запуска. Просто включите ЦП с нажатой кнопкой диагностики и удерживайте ее до тех пор, пока не появится сообщение «APP. НЕ ЗАГРУЖЕНО». При попытке входа в систему программное обеспечение MasterTool IEC XE укажет, что в ЦП нет приложения. Для перезагрузки приложения необходимо перезагрузить ЦП или выполнить новую загрузку приложения.

7.5. Отказ источника питания

Блок питания серии Nexto (NX8000) имеет систему обнаружения отказов в соответствии с уровнями, определенными в его технических характеристиках (см. Технические характеристики блока питания 30 Вт, 24 В пост. тока — CE114200). Есть два способа диагностировать неисправность:

1. В случае, если источник питания NX8000 включен с напряжением ниже требуемого минимального предела, генерируется диагностическое сообщение об отказе источника питания, которое распознается ЦП, и на дисплее отображается сообщение «POWER FAILURE». Когда питание находится в установленных пределах, ЦП распознает это и автоматически перезапускается с пользовательским приложением. Диагностика по-прежнему будет активной, чтобы показать пользователю, что при последней инициализации произошел сбой источника питания.

2. В случае, если NX8000 имеет падение напряжения ниже минимально необходимого значения и возвращается к более высокому значению в течение 10 мс, ЦП не распознает сбой питания, и диагностика не генерируется, так как система продолжает работать. в целости и сохранности за это время. Но если падение напряжения длится более 10 мс, на экране ЦП отображается сообщение «POWER FAILURE» и активируется диагностика..



Рисунок 198: Сообщение об отказе источника питания

Пользователь может изменить значение переменной, относящейся к отказу источника питания, на FALSE во время выполнения приложения, облегчая проверку и обработку этой диагностики.

Диагностика POWER FAILURE уже сопоставлена с определенной областью памяти, определенной как Подробная диагностика ЦП. Таким образом, это просто использовать его как глобальную переменную. Имя переменной описано в подробном списке диагностики в разделе Диагностика через переменные.

7.6. Общие проблемы

Если при включении ЦП он не работает, необходимо проверить следующие пункты:

Находится ли температура в помещении в пределах поддерживаемого устройством диапазона?

На блок питания стойки подается правильное напряжение?

Вставлен ли модуль питания в крайний левый угол стойки (если смотреть на стойку спереди), за которым следует ЦП серии Nexto?

Имеются ли сетевые устройства, такие как концентраторы, коммутаторы или маршрутизаторы, запитанные, взаимосвязанные, настроенные и работающие должным образом?

Правильно ли подключен сетевой кабель Ethernet к порту NET 1 или NET 2 процессора Nexto и к сетевому устройству? ЦП Nexto Series включен, находится в режиме выполнения (Run) и не проводит диагностику оборудования?

Если ЦП Nexto указывает режим выполнения (Выполнение), но не отвечает на запрошенные сообщения, будь то через MasterTool IEC XE или протоколы, необходимо проверить следующие элементы:

Конфигурация параметров CPU Ethernet правильная?

Правильно ли настроен соответствующий коммуникационный протокол в CPU? Правильно ли включены переменные, которые разрешают отношения MODBUS?

Если проблема не обнаружена, обратитесь в службу технической поддержки Altus.

7.7. Поиск неисправностей

В таблице ниже показаны симптомы некоторых проблем с их возможными причинами и решениями. Если проблема не устранена, обратитесь в службу технической поддержки Altus.

Симптом	Возможные причины	Решение
Не включает процессор	Отсутствие питания или неправильное питание.	Убедитесь, что ЦП правильно подключен к стойке. Выключите питание и снимите все модули с шины, кроме источника питания и процессора. Включите питание шины и проверьте работу источника питания, внешнего и того, что находится в стойке. Проверьте, попадает ли напряжение питания на контакты источника питания Nexto и правильно ли оно поляризовано.
На экране отображается сообщение НЕПРАВИЛЬНЫЙ СЛОТ.	ЦП в неправильном положении.	ЦП должен быть размещен в слотах 2 и 3 стойки 0. Установите его в правильные слоты. ЦП должны быть размещены в слотах 2 и 3 стойки 0. Установите их в правильные слоты.
Не общается	Плохой контакт или неправильная конфигурация.	Проверьте каждое соединение кабеля связи. Проверьте конфигурацию последовательного интерфейса и интерфейса Ethernet в программе MasterTool IEC XE.
Симптом	Возможные причины	Решение
Не распознает карту памяти	Плохое соединение или не установлен	Убедитесь, что карта памяти правильно вставлена в отсек. Убедитесь, что карта памяти вставлена правильно, как указано на передней панели ЦП. Проверить, не размонтировалась ли карта памяти, с помощью кнопки MS, расположенной на передней панели, визуализируя индикацию на графическом дисплее процессора.

Таблица 247: Поиск и устранение неисправностей

7.8.Профилактическое обслуживание

Необходимо ежегодно проверять, надежно ли подключены соединительные кабели, без скопления пыли, в основном защитные устройства.

В средах, подверженных чрезмерному загрязнению, оборудование необходимо периодически очищать от пыли, мусора и т.п.

Диоды TVS, используемые для защиты от переходных процессов, вызванных атмосферными разрядами, должны периодически проверяться, так как они могут быть повреждены или разрушены, если поглощенная энергия превышает предел. Во многих случаях неисправность может быть не визуальной. В критических случаях рекомендуется периодическая замена диодов TVS, даже если они не показывают визуальных сигналов неисправности.

Автобусная герметичность и чистота каждые полгода.

Для получения дополнительной информации см. Руководство по серии Nexto — MU214600.

8. Приложение. DNP3 Совместимость

8.1. Профиль устройства DNP3

DNP3 ДОКУМЕНТ ПРОФИЛЯ УСТРОЙСТВА	
Идентификация устройства	
Имя поставщика	Altus S/A
Имя устройства	NX3030
Функция устройства	подчиненное устройство
Поддерживаемые уровни DNP для	Запросы: Нет Ответы: нет
Поддерживаемые соединения	IP Networking
Методы установки настраиваемых параметров	Программное обеспечение: MasterTool IEC XE
IP Networking	
Тип конечной точки:	Прослушивание TCP (только Outstation)
Принимает TCP-соединения от	Разрешить все
IP-адреса, с которых принимаются TCP-соединения:	*.*.*.*
Номер порта прослушивания TCP	Конфигурируемый, диапазон от 1 до 65535
Таймер проверки активности TCP	Настраиваемый, диапазон от 0 до 4294967295
Несколько основных соединений	Поддерживает до двух мастеров На основе номера порта TCP
Поддержка синхронизации времени	SNTP
Связующий слой	
Адрес канала передачи данных	Настраиваемый, диапазон от 0 до 65519
Поддержка собственного адреса с использованием адреса 0xFFFC	Нет
Требуется подтверждение канального уровня	Никогда
Максимальное количество октетов, передаваемых в кадре канала передачи данных	Зафиксировано при 292
Максимальное количество октетов, которое может быть получено в кадре канала передачи данных	Зафиксировано при 292

Прикладной уровень	
Максимальное количество октетов, передаваемых во фрагменте прикладного уровня	Зафиксировано при 2048
Максимальное количество октетов, которое может быть получено во фрагменте прикладного уровня	Исправлено в 2048
Тайм-аут ожидания подтверждения приложения запрошенного ответного сообщения	Фиксировано на 10000 мс
Бит неисправности устройства IIN1.6	Этот бит будет установлен, если ПЛК не находится в режиме RUN.
Поведение при переполнении буфера событий	Отменить самое старое событие
Отправляет многофрагментные ответы	Да
Поддержка незапрашиваемых ответов Outstation	
Поддерживает нежелательные отчеты	Нет

Таблица 248: Пофайл устройства DNP3

8. ПРИЛОЖЕНИЕ. ВЗАИМОДЕЙСТВИЕ DNP3

8.2. Таблица реализации DNP V3.0

ГРУППА ОБЪЕКТОВ DNP И ВАРИАЦИЯ			ЗАПРОС		ОТВЕТ	
Group Num	Var Num	Описание	Функциональный код (dec)	Коды квалификаторов (hex)	Функциональный код (dec)	Коды квалификаторов (hex)
1	0	Двоичный ввод – любая вариация	1 (чтение)	00, 01 (старт-стоп) 06 (нет диапазона или все)		
1	1	Двоичный ввод – упакованный формат	1 (чтение)	00, 01 (старт-стоп) 06 (нет диапазона или все)	129 (ответ)	00, 01 (старт-стоп)

2	0	Двоичное входное событие – любая вариация	1 (чтение)	06 (нет диапазона или все) 07, 08 (ограниченное кол-во)		
2	2	Событие бинарного ввода – с абсолютным временем	1 (чтение)	06 (нет диапазона или все) 07, 08 (ограниченное кол-во)	129 (ответ)	17, 28 (индекс)
60	1	Объекты класса – данные класса 0	1 (чтение)	06 (нет диапазона или все)		
60	2	Объекты класса – данные класса 1	1 (чтение)	06 (нет диапазона или все) 07, 08 (ограниченное кол-во)		
80	1	Внутренние показания – упакованный формат	1 (чтение)	00, 01 (старт-стоп)	129 (ответ)	00 (старт-стоп)
			2 (записать)	00 (старт-стоп) индекс=7		

Таблица 249: Таблица реализации DNP V3.0