

HIMax®

Руководство по связи

SAFETY
NONSTOP



КОММУНИКАЦИЯ

Все названные в данном руководстве HIMA изделия защищены товарным знаком. То же самое распространяется, если не указано другое, на прочих упоминаемых изготовителей и их продукцию.

Все технические данные и указания настоящего руководства разработаны с особой тщательностью и составлены при использовании эффективного контроля. При возникновении вопросов обращайтесь в компанию HIMA. Фирма HIMA с благодарностью принимает предложения по внесению в руководство необходимой дополнительной информации.

Право на внесение технических изменений сохраняется. Фирма HIMA оставляет за собой также право обновлять написанный материал без предварительного уведомления.

Подробная информация содержится на компакт-диске и на нашем сайте <http://www.hima.de> и <http://www.hima.com>.

© Copyright 2008, HIMA Paul Hildebrandt GmbH

Все права сохраняются

Контактные данные

Адрес компании HIMA:

HIMA Paul Hildebrandt GmbH

Postfach 1261

D-68777 Brühl

Тел.: +49(06202) 709-0

Факс: +49(06202) 709-107

Эл. почта: info@hima.com

Содержание

1	Введение	11
1.1	Структура и использование руководства	11
1.2	Целевая группа	12
1.3	Варианты написания текста	12
1.3.1	Указания по безопасности	12
1.3.2	Указания по использованию	13
2	Безопасность	14
2.1	Условия использования	14
2.1.1	Климатические условия	14
2.1.2	Механические условия	16
2.1.3	Условия по ЭМС	16
2.1.4	Подача напряжения	16
2.1.5	Защитные меры от воздействия электростатического разряда	17
2.2	Остаточный риск	18
2.3	Меры предосторожности	18
2.4	Информация для экстренных случаев	18
3	Описание изделия	19
3.1	Безопасный протокол (safeethernet)	19
3.2	Стандартные протоколы	20
3.3	Избыточность	21
3.4	Интерфейсы Ethernet	22
3.4.1	Свойства интерфейсов Ethernet	22
3.4.2	Конфигурация интерфейсов Ethernet	23
3.4.3	Используемые сетевые порты для связи Ethernet	27
3.5	Интерфейсы полевой шины	27
3.5.1	Назначение штырьковых выводов гнезд D-SUB FB1 и FB2	27
3.5.2	Протоколы и интерфейсы	29
3.5.3	Структура номера модуля COM	30
4	safeethernet	31
4.1	Что представляет собой safeethernet	32
4.2	Конфигурация избыточного соединения safeethernet	34
4.3	Редактор safeethernet	37
4.4	Детальный вид редактора safeethernet	39
4.5	Возможные соединения safeethernet	42
4.5.1	Одиночное соединение safeethernet (канал 1)	42
4.5.2	Избыточное соединение safeethernet (канал 1 и канал 2)	42
4.5.3	Разрешенные комбинации	43

4.6	Параметры safeethernet.....	44
4.6.1	Максимальное время цикла системы управления HIMax	44
4.6.2	Время ожидания приема (Receive Timeout)	44
4.6.3	Время ответа (ResponseTime).....	45
4.6.4	Настройка Sync/Async	45
4.6.5	Время ожидания повторной передачи (ResendTMO)	46
4.6.6	Время ожидания подтверждения (Acknowledge Timeout)	46
4.6.7	Интервал передачи пакетов (Prod-Rate)	46
4.6.8	Память	47
4.7	Максимальное время реакции	47
4.7.1	Расчет максимального времени реакции двух ПЭС HIMax	47
4.7.2	Расчет макс. времени реакции в соединении с ПЭС HIMatrix	48
4.7.3	Расчет макс. времени реакции с двумя системами управления HIMatrix или удаленными устройствами ввода/вывода	48
4.7.4	Расчет макс. времени реакции с двумя системами HIMax и одной ПЭС HIMatrix	49
4.8	Профили Safeethernet.....	50
4.8.1	Профиль I (Fast & Cleanroom).....	50
4.8.2	Профиль II (Fast & Noisy)	51
4.8.3	Профиль III (Medium & Cleanroom).....	51
4.8.4	Профиль IV (Medium & Noisy)	52
4.8.5	Профиль V (Slow & Cleanroom)	52
4.8.6	Профиль VI (Slow & Noisy)	53
4.9	Межпроектная связь	54
4.9.1	Варианты межпроектной связи.....	55
4.10	Межпроектная связь между SILworX и ELOP II Factory	56
4.10.1	Конфигурация HIMax в проекте SILworX	56
4.10.2	Конфигурация HIMatrix в ELOP II Factory	60
4.11	Панель управления (safeethernet)	62
4.11.1	Поле индикации (соединение safeethernet).....	63
5	PROFIBUS DP	64
5.1	Ведущее устройство HIMA PROFIBUS DP	65
5.1.1	Создание ведущего устройства HIMA PROFIBUS DP	65
5.2	Пример PROFIBUS DP	66
5.2.1	Конфигурирование ведомого устройства PROFIBUS DP	66
5.2.2	Конфигурирование ведущего устройства PROFIBUS DP	68
5.3	Функции меню ведущего устройства PROFIBUS DP	75
5.3.1	Редактирование (Edit)	75
5.3.2	Функция меню Свойства	76
5.4	Метод доступа к шине PROFIBUS-DP	80
5.4.1	Протокол ведущего/ведомого устройства	80
5.4.2	Протокол маркера.....	80
5.4.3	Контрольное время обращения маркера (Ttr).....	80
5.4.4	Расчет контрольного времени обращения маркера Ttr.....	81
5.5	Изохронный цикл PROFIBUS DP (начиная с DP-V2)	83
5.5.1	Изохронный режим (начиная с DP-V2)	84
5.5.2	Изохронный режим Sync (начиная с DP-V2)	84
5.5.3	Изохронный режим Freeze (начиная с DP-V2)	84

5.6	Функции меню ведомого устройства PROFIBUS DP (в ведущем устройстве).....	85
5.6.1	Создание ведомого устройства PROFIBUS DP (в ведущем устройстве).....	85
5.6.2	Редактирование (Edit).....	85
5.6.3	Свойства	86
5.7	Считывание файла GSD	92
5.8	Обработка параметров пользователя	93
5.9	Функциональные блоки PROFIBUS	95
5.9.1	Функциональный блок MSTAT	96
5.9.2	Функциональный блок RALRM	99
5.9.3	Функциональный блок RDIAG	103
5.9.4	Функциональный блок RDREC	107
5.9.5	Функциональный блок SLACT	110
5.9.6	Функциональный блок WRREC	113
5.10	Вспомогательные функциональные блоки PROFIBUS	116
5.10.1	Вспомогательный функциональный блок ACTIVE	116
5.10.2	Вспомогательный функциональный блок ALARM	117
5.10.3	Вспомогательный функциональный блок DEID	118
5.10.4	Вспомогательный функциональный блок ID	119
5.10.5	Вспомогательный функциональный блок NSLOT	120
5.10.6	Вспомогательный функциональный блок SLOT	120
5.10.7	Вспомогательный функциональный блок STDDIAG	121
5.11	Коды ошибок функциональных блоков	123
5.12	Панель управления (ведущее устройство PROFIBUS DP)	124
5.12.1	Контекстное меню (ведущее устройство PROFIBUS)	124
5.12.2	Поле индикации (ведущее устройство PROFIBUS)	125
5.12.3	Состояние ведущего устройства PROFIBUS DP	126
5.12.4	Поведение ведущего устройства PROFIBUS DP	126
5.12.5	Функция светодиода FBx у ведущего устройства PROFIBUS DP	127
5.13	Ведомое устройство HIMA PROFIBUS DP	128
5.13.1	Создание ведомого устройства HIMA PROFIBUS DP	128
5.14	Функции меню ведомого устройства PROFIBUS DP	129
5.14.1	Редактирование (Edit).....	129
5.14.2	Свойства	131
5.15	Панель управления (ведомое устройство PROFIBUS DP)	132
5.15.1	Контекстное меню (ведомое устройство PROFIBUS DP)	132
5.15.2	Поле индикации (ведомое устройство PROFIBUS DP).....	133
5.16	Функция светодиода FBx у ведомого устройства PROFIBUS DP	133
6	Modbus	134
6.1	Ведущее устройство Modbus HIMA.....	135
6.2	Пример Modbus	136
6.2.1	Конфигурация ведущего устройства Modbus TCP	137
6.2.2	Конфигурация ведомого устройства Modbus TCP	139
6.3	Функции меню ведущего устройства Modbus HIMA	140
6.3.1	Редактирование (Edit).....	140
6.3.2	Свойства	141

6.4	Функции Modbus (телеграмма-запрос).....	143
6.4.1	Телеграмма-запрос для считывания	145
6.4.2	Телеграмма-запрос для записи.....	147
6.4.3	Ведомые устройства Ethernet (ведомые устройства TCP/UDP).....	149
6.4.4	Системные переменные ведомого устройства TCP/UDP	150
6.4.5	Свойства ведомого устройства TCP/UDP	151
6.4.6	Шлюз Modbus (шлюз TCP/UDP)	152
6.4.7	Свойства шлюза	154
6.4.8	Системные переменные ведомого устройства-шлюза	154
6.4.9	Свойства шлюза ведомого устройства-шлюза	154
6.4.10	Последовательный Modbus	155
6.4.11	Свойства последовательного Modbus	156
6.4.12	Системные переменные ведомого устройства Modbus	156
6.4.13	Свойства ведомого устройства Modbus	157
6.5	Панель управления (ведущее устройство Modbus).....	157
6.5.1	Контекстное меню (ведущее устройство Modbus).....	158
6.5.2	Поле индикации (ведущее устройство Modbus)	158
6.6	Панель управления (ведущее устройство > ведомое устройство Modbus)	158
6.7	Функция светодиода FBx у ведущего устройства Modbus.....	159
6.8	Ведомое устройство Modbus HIMA	160
6.9	Функции меню ведомого устройства Modbus HIMA	161
6.9.1	Свойства.....	161
6.9.2	Системные переменные	165
6.9.3	Регистровая переменная	165
6.9.4	Битовые переменные	165
6.9.5	Присвоение переменных передачи/приема	166
6.9.6	Функции Modbus.....	167
6.9.7	Специальные коды функций HIMA:.....	169
6.9.8	Режим адресации Modbus (режим совместимости).....	171
6.9.9	Режим адресации Modbus (адресация через бит и регистр)	172
6.9.10	Смещения для альтернативной адресации Modbus	174
6.10	Панель управления (ведомое устройство Modbus).....	176
6.10.1	Контекстное меню (ведомое устройство Modbus)	176
6.10.2	Поле индикации (ведомое устройство Modbus).....	177
6.10.3	Поле индикации (данные ведущего устройства)	177
6.11	Функция светодиода FBx у ведомого устройства Modbus.....	178
6.11.1	Коды ошибок соединения TCP/IP Modbus	178
7	Send & Receive TCP	179
7.1	Требования к системе.....	179
7.1.1	Создание протокола S&R-TCP	180
7.2	Пример: Конфигурация S&R TCP.....	180
7.2.1	Конфигурация S&R TCP системы управления SIMATIC 300 Siemens	182
7.2.2	S&R TCP Конфигурирование системы управления HIMAх	186
7.3	Функции меню протокола S&R-TCP	188
7.3.1	Редактирование (Edit)	188
7.3.2	Свойства.....	188

7.4	Функции меню соединения TCP	190
7.4.1	Редактирование (Edit).....	190
7.4.2	Системные переменные.....	190
7.4.3	Свойства.....	191
7.5	Обмен данными	193
7.5.1	Соединения TCP.....	193
7.5.2	Циклический обмен данными.....	194
7.5.3	Ациклический обмен данными с помощью функциональных блоков.....	194
7.5.4	Одновременный циклический и ациклический обмен данными.....	194
7.5.5	Контроль потока.....	195
7.6	Сторонние системы с холостыми байтами	195
7.7	Функциональные блоки S&R-TCP	196
7.7.1	TCP_Reset.....	197
7.7.2	TCP_Send.....	200
7.7.3	TCP_Receive.....	203
7.7.4	TCP_ReceiveLine.....	207
7.7.5	TCP_ReceiveVar.....	211
7.8	Панель управления (Send/Receive over TCP)	216
7.8.1	Контекстное меню (Send/Receive Protocol).....	216
7.8.2	Поле индикации (Send/Receive Protocol).....	216
7.8.3	Поле индикации (Send/Receive Server).....	216
7.8.4	Код ошибки соединения TCP.....	217
7.8.5	Таблица дополнительных кодов ошибок функциональных блоков.....	218
7.8.6	Состояние соединения.....	218
7.8.7	Состояние соединения участника коммуникации.....	218
8	SNTP-Protocol (протокол SNTP)	219
8.1	SNTP Client (клиент SNTP).....	219
8.2	Клиент SNTP (информация по серверу).....	221
8.3	SNTP Server (сервер SNTP).....	222
9	Сервер X-OPC	223
9.1	Конфигурирование соединения сервера OPC	226
9.1.1	Необходимое программное обеспечение:.....	226
9.1.2	Условия для работы сервера X-OPC:.....	226
9.1.3	Установка сервера OPC.....	227
9.1.4	Конфигурирование интерфейса DCOM.....	228
9.1.5	Создание сервера OPC в SILworX.....	229
9.1.6	Настройки сервера OPC в редакторе safeethernet.....	230
9.1.7	Конфигурация в качестве сервера доступа к данным X-OPC.....	231
9.1.8	Конфигурация в качестве сервера аварийных сигналов и событий X-OPC.....	233
9.2	Параметрирование	235
9.2.1	Набор серверов OPC.....	235
9.2.2	Сервер OPC.....	239
9.2.3	Alarm & Events (аварийные сигналы и события).....	240
9.3	Удаление сервера X-OPC	243

10	Конфигурирование функциональных блоков	244
10.1	Создание библиотек функциональных блоков	244
10.2	Конфигурирование функциональных блоков в прикладной программе	244
10.3	Конфигурирование функциональных блоков в структуре SILworX	245
	Приложение	249
	Глоссарий	249
	Список изображений	250
	Список таблиц	252
	Алфавитный указатель	258

1 Введение

В руководстве по связи описаны свойства и конфигурация протоколов связи безопасных систем управления HIMax.

С помощью предоставленных протоколов Вы можете соединять системы управления HIMax между собой и с системами управления других изготовителей.

Знание предписаний и технически безупречное выполнение содержащихся в данном руководстве указаний квалифицированным персоналом является обязательным условием при планировании, проектировании, программировании, установке, вводе в эксплуатацию, эксплуатации и ремонте систем управления HIMax.

Неквалифицированное вмешательство в устройства, отключение или невыполнение функций безопасности (шунт), или несоблюдение указаний данного руководства (и возникшие в результате этого неисправности или ухудшение функций безопасности) могут привести к травмам, материальному и экологическому ущербу, за которые фирма HIMA не несет никакой ответственности.

Устройства автоматизации HIMax разрабатываются, изготавливаются и испытываются при соблюдении соответствующих норм по безопасности. Использовать только в предусмотренных в описаниях случаях применения в указанных условиях окружающей среды.

1.1 Структура и использование руководства

Содержание данного руководства является частью описания аппаратного обеспечения программируемой электронной системы HIMax.

Руководство разделено на следующие основные главы:

- Введение
- Безопасность
- Описание изделия
- Ввод в эксплуатацию
- Эксплуатация
- Техобслуживание
- Вывод из эксплуатации
- Транспортировка
- Утилизация

Также следует использовать следующую документацию:

Название	Содержание	№ документа
HIMax Руководство по системе	Описание аппаратного обеспечения системы HIMax	HI 801 060 R
HIMax Руководство по функциональной безопасности	Функции безопасности системы HIMax	HI 801 061 R
HIMax Руководство по связи	Описание линий связи и протоколов	HI 801 162 R
Первые шаги	Введение в SILworX	HI 801 061 E

Таблица 1: Дополнительные руководства

Актуальные версии руководств Вы найдете на сайте компании HIMA www.hima.de. С помощью номера версии в нижнем колонтитуле Вы можете определить актуальность имеющихся руководств по сравнению с версией, выложенной в интернете.

1.2 Целевая группа

Данный документ предназначен для планировщиков, проектировщиков и программистов автоматических установок, а также специалистов, выполняющих ввод в эксплуатацию, эксплуатацию и техническое обслуживание устройств и систем. Условием для эксплуатации является наличие специальных знаний в области безопасных автоматизированных систем.

1.3 Варианты написания текста

Для более удобного прочтения и наглядности в данном документе используются следующие варианты написания:

Полужирный	Выделение важных частей текста. Обозначения кнопок, разделов меню и вкладок в SILworX, на которые Вы можете щелкнуть
<i>Курсив</i>	Параметры и системные переменные
<code>Courier</code>	Текстовый ввод пользователей
RUN	Обозначения рабочих состояний большими буквами
Гл. 1.2.3	Перекрестные ссылки являются гиперссылками, даже если они не имеют особого обозначения. Если Вы наведете на ссылку указатель мыши, то она поменяет свой вид. Если щелкнуть на ссылку происходит переход в соответствующее место документа.

Указания по безопасности и использованию имеют особое обозначение.

1.3.1 Указания по безопасности

Указания по безопасности представлены в документе следующим образом. Эти указания должны обязательно соблюдаться, чтобы максимально уменьшить степень риска. Структура содержания:

- Сигнальное слово: опасно, предупреждение, осторожно, указание
- Вид и источник опасности
- Последствия
- Избежание опасности

СИГНАЛЬНОЕ СЛОВО



Вид и источник опасности!

Последствия

Избежание опасности

Значение сигнальных слов:

- **Опасно:** Несоблюдение указаний ведет к серьезным телесным повреждениям и летальному исходу
- **Предупреждение:** Несоблюдение указаний ведет к серьезным телесным повреждениям и летальному исходу
- **Осторожно:** При несоблюдении указаний возможные легкие телесные повреждения
- **Указание:** При несоблюдении указаний возможен материальный ущерб

УКАЗАНИЕ

Вид и источник повреждения!
Избежание повреждения

- 1.3.2 Указания по использованию
Дополнительная информация предлагается, как показано на следующем примере:

i

Здесь располагается текст дополнительной информации.

Полезные советы и рекомендации появляются в такой форме:

СОВЕТ Здесь располагается текст совета.

В основе всех данных по SILworX лежит версия на английском языке.

2 Безопасность

Следует обязательно прочесть следующую информацию по безопасности, указания и инструкции. Использовать изделие только при соблюдении всех директив и правил безопасности.

Данный продукт эксплуатируется с БСНН или ЗСНН. Модуль сам по себе не является источником опасности. Использование во взрывоопасных областях разрешено только в том случае, если приняты дополнительные меры безопасности.

2.1 Условия использования

Устройства были разработаны при соблюдении следующих стандартов по ЭМС, климатических и экологических требований:

Стандарт	Содержание
ЕС/EN 61131-2: 2006	Программируемые логические контроллеры, часть 2 Требования к ресурсам и испытания
IEC/EN 61000-6-2: 2005	ЭМС Отраслевая норма, часть 6-2 Помехоустойчивость, промышленная сфера
IEC/EN 61000-6-4: 2006	Электромагнитная совместимость (ЭМС) Отраслевая норма, эмиссия помех, промышленная сфера

Таблица 2: Нормы для ЭМС, климатических и экологических требований

При использовании безопасных систем управления HiMax необходимо соблюдать следующие общие условия:

Условие	Содержание
Класс защиты	Класс защиты II согл. МЭК/EN 61131-2
Степень загрязнения	Степень загрязнения II согл. МЭК/EN 61131-2
Высота над уровнем моря	< 2000 м
Корпус	Стандарт: IP 20 Если того требуют соответствующие нормы применения (напр., EN 60204, EN 954-1), устройство необходимо встраивать в корпус с необходимой степенью защиты (напр., IP 54).

Таблица 3: Общие условия

2.1.1 Климатические условия

Наиболее важные испытания и предельные значения для климатических условий перечислены в таблице ниже:

IEC/EN 61131-2	Климатические испытания
	Рабочая температура: 0...+60 °C (Предельные значения при испытании: -10...+70 °C)
	Температура хранения: -40...+85 °C
	Сухое тепло и холод; испытания на прочность: +70 °C/-25 °C, 96 ч, электропитание не подключено
	Смена температуры; испытание на прочность и нечувствительность: -25 °C/+70 °C и 0 °C/+55 °C, электропитание не подключено
	Циклы с влажным теплом; испытания на прочность: +25 °C/+55 °C, 95% относительная влажность, электропитание не подключено

Таблица 4: Климатические условия

2.1.2 Механические условия

Наиболее важные испытания и предельные значения для механических условий перечислены в таблице ниже:

IEC/EN 61131-2	Механические испытания
	Испытание на нечувствительность по отношению к вибрациям: 5...9 Гц/3,5 мм 9...150 Гц, 1 г, испытуемый образец в эксплуатации, 10 циклов на ось
	Нечувствительность по отношению к ударам: 15 г, 11 мс, испытуемый образец в эксплуатации, 2 цикла на ось

Таблица 5: Механические испытания

2.1.3 Условия по ЭМС

Для безопасных систем требуются повышенные уровни при воздействии возмущений. Системы HiMax отвечают данным требованиям согласно МЭК 62061 и МЭК 61326-3-1 (DIS). См. столбец «Критерий ФБ» (функциональная безопасность).

IEC/EN 61131-2	Испытания на помехоустойчивость	Критерий ФБ
IEC/EN 61000-4-2	Испытание на воздействие электростатических разрядов: 6 кВ контактный разряд, 8 кВ воздушный разряд	-
IEC/EN 61000-4-3	Испытание RFI (10 В/м): 26 МГц...1 ГГц, 80% AM Испытание RFI (20 В/м): 26 МГц...2,7 ГГц, 80% AM: EN 298	- 20 В/М
IEC/EN 61000-4-4	Испытание на устойчивость к наносекундным импульсным помехам: 2 кВ линия снабжения, 1 кВ сигнальная линия	4 кВ
IEC/EN 61000-4-12	Испытание затухающими колебаниями: 2,5 кВ L-, L+/PE 1 кВ L+/L -	

Таблица 6: Испытания на помехоустойчивость

IEC/EN 61000-6-2	Испытания на помехоустойчивость	Критерий ФБ
IEC/EN 61000-4-6	Высокая частота, асимметрично: 10 В, 150 кГц...80 МГц, AM 20 В, 150 кГц...80 МГц, AM: EN 298	20 В
IEC/EN 61000-4-3	Импульсы 900 МГц	
IEC/EN 61000-4-5	Импульсное напряжение: 2 кВ, 1 кВ	2 кВ/1 кВ

Таблица 7: Испытания на помехоустойчивость

IEC/EN 61000-6-4	Испытания на эмиссию помех
EN 55011 Класс А	Эмиссия помех: излуч., проводной

Таблица 8: Испытания на эмиссию помех

2.1.4 Подача напряжения

Наиболее важные испытания и предельные значения для подачи напряжения на устройства перечислены в таблице ниже:

IEC/EN 61131-2	Дополнительная проверка характеристик подачи постоянного напряжения
	Подача напряжения альтернативно должна отвечать следующим стандартам: МЭК/EN 61131-2 или БСНН (Safety Extra Low Voltage, SELV) или ЗСНН (Protective Extra Low Voltage, PELV)
	Защита устройств HiMax должна осуществляться согласно данным, содержащимся в этом руководстве.
	Проверка диапазона напряжений: 24 В DC, -20%...+25% (19,2 В...30,0 В)
	Испытание на нечувствительность в случае краткого прерывания подачи электропитания от внешнего источника: пост. т, PS 2: 10 мс
	Изменение полярности питающего напряжения: Указание в соответствующей главе Руководства по системе или в таблице параметров для линии электропитания.
	Буферное время, испытание на прочность: Испытание В, 1000 ч

Таблица 9: Дополнительная проверка характеристик подачи постоянного напряжения

2.1.5 Защитные меры от воздействия электростатического разряда

Изменение и расширение системы или замену модуля может выполнять только персонал, ознакомленный с защитными мерами от воздействия электростатического разряда.

УКАЗАНИЕ



Электростатические разряды могут повредить встроенные в системы управления электронные конструктивные элементы!

- Выполняйте работу на рабочих местах с антистатическим оснащением и носите заземляющую ленту.
- При неиспользовании предохраняйте модули от электростатического разряда, например, храните в упаковке.

Изменения или расширения в проводке системы может выполнять только персонал, ознакомленный с мерами защиты от электростатического разряда.

2.2 Остаточный риск

Модуль H1Max сам по себе не является источником опасности.

Остаточный риск исходит от:

- ошибок в проектировании
- ошибок в прикладной программе
- ошибок в электропроводке

2.3 Меры предосторожности

В месте использования соблюдать действующие положения по технике безопасности и носить предписанные средства индивидуальной защиты. Для монтажа несущего каркаса X-BASE PLATE необходимо надевать защитную обувь.

2.4 Информация для экстренных случаев

Система управления H1Max является частью технического оборудования, служащего для безопасности установки. Сбой в системе управления переведет установку в безопасное состояние.

В экстренной ситуации запрещается проникать в любое техническое оборудование, служащее для безопасности установки.

3 Описание изделия

Система HIMax представляет собой безопасную систему управления, предназначенную для непрерывной эксплуатации и максимальной готовности.

HIMax представляет собой модульную систему. Функции обработки, ввода/вывода и связи распределены по съемным модулям.

Наряду с физическими входными и выходными переменными системы HIMax переменные могут обмениваться также с другой системой через канал передачи данных.

Для обмена данными процесса предусмотрен безопасный протокол **safeethernet** и различные стандартные протоколы.

3.1 Безопасный протокол (safeethernet)

Все системы HIMax могут соединяться безопасно на уровне совокупной безопасности 3 через Ethernet. Безопасная связь осуществляется через протокол **safeethernet**.

Безопасный протокол **safeethernet** используется для безопасного обмена данными процесса между HIMax и системами управления HIMatrix в сети Ethernet.

Процессорные модули:	safeethernet реализуется на модуле ЦПУ. Максимально 4 модуля ЦПУ для одной системы управления HIMax.
Соединения:	Максимально 255 соединений safeethernet для одной системы управления HIMax.
Маршруты:	Интерфейсы Ethernet модулей ЦПУ и COM. Используемые интерфейсы Ethernet параллельно можно использовать для других протоколов.
Набор данных процесса:	Через каждое соединение safeethernet можно в каждое направление передавать максимум 1100 байтов.

i Максимальный набор данных процесса системы управления равный 512 кБ не должен превышать. В этом случае при загрузке будет отклонено параметрирование системы управления.

Описание безопасного протокола **safeethernet**, начиная с главы 0.

⚠ ПРЕДУПРЕЖДЕНИЕ



Управление безопасной передачей данных!

Ущерб, причиняемый человеку

Эксплуатирующая сторона должна позаботиться о том, чтобы используемая для **safeethernet** сеть Ethernet была достаточно защищена от несанкционированного доступа (напр., хакером).

Вид и объем мер необходимо согласовать с осуществляющим приемку отделом контроля.

3.2 Стандартные протоколы

Для оптимальной интеграции системы HiMax с полевыми устройствами и системами управления предлагаются дополнительные испытанные стандартные протоколы (NSIP). Для этого применяется как Ethernet, так и протоколы полевых шин.

Ряд протоколов связи обеспечивает лишь небезопасную передачу данных. Они могут использоваться только для небезопасных частей функции автоматизации.

Модули связи:	Стандартные протоколы реализованы на модуле COM Максимально: <ul style="list-style-type: none"> 20 модулей COM для одной системы управления HiMax. 20 стандартных протоколов для одной системы управления HiMax. 6¹⁾ Стандартные протоколы/модуль COM
Маршруты:	Интерфейсы Ethernet и интерфейсы полевых шин модулей COM.
Набор данных процесса:	При использовании всех небезопасных протоколов в одной системе управления HiMax можно передавать всего 128 кБ и принимать 128 кБ данных.
¹⁾ Клиент SNTP и сервер SNTP сюда не относятся.	

Предлагаются следующие стандартные протоколы:

Протокол	на модуль	Описание
PROFIBUS DP Master	2	Глава 0
PROFIBUS DP Slave	1	Глава 0
Modbus Master	1	Глава 0
Modbus Slave	1	Глава 0
S&R TCP	1	Глава 0

Таблица 10: Количество стандартных протоколов на модуль

i

Максимум 64 соединения TCP на модуль COM.

Максимум 1280 соединения TCP на одну систему управления HiMax с 20 модулями COM.

⚠ ПРЕДУПРЕЖДЕНИЕ



Использование ненадежных данных импорта!

Ущерб, причиняемый человеку

Ненадежные данные не должны использоваться для функций безопасности прикладной программы.

Примеры использования модуля связи:

В этих примерах на каждом из 2 модулей связи X-COM 01 конфигурировано максимальное количество протоколов и соединений.

Максимально доступно 64 сокета TCP на модуль.

Модуль COM 1

Протокол	Соединения
1 Modbus Master	44 соединения ведомого устройства TCP, 122 соединения ведомого устройства RTU
1 Modbus Slave	20 соединений ведущего устройства TCP, 1 соединение ведущего устройства RTU

Таблица 11: Протоколы на модуле COM 1

Модуль COM 2

Протокол	Соединения
1 PROFIBUS DP Master	122 соединения ведомого устройства
1 PROFIBUS DP Slave	1 соединение ведущего устройства

Таблица 12: Протоколы на модуле COM 3

3.3**Избыточность**

Система HIMax представляет собой систему с высокой степенью готовности и предлагает также избыточность для связи. Коммуникационное соединение является избыточным, если существует два физических маршрута.

Избыточность через safeethernet

Избыточность Вы можете конфигурировать в редакторе **safeethernet**, выбрав интерфейсы Ethernet для второго маршрута (см. главу 4.2).

Избыточность через стандартные протоколы

PROFIBUS DP Master PROFIBUS DP Slave TCP S&R	Избыточность стандартного протокола Вы должны конфигурировать в прикладной программе таким образом, чтобы она контролировала избыточные маршруты и присваивала переданные с избыточностью данные процессу соответствующему маршруту.
Modbus Master Modbus Slave	Не могут быть конфигурированы с избыточностью.

3.4 Интерфейсы Ethernet

Связь с внешними системами осуществляется через интерфейсы Ethernet модулей ЦПУ и COM. Каждый интерфейс Ethernet может одновременно обрабатывать несколько протоколов.

i

Через интерфейсы Ethernet модуля системной шины X-SB 01 обмен данными процесса невозможен. Интерфейсы Ethernet UP и DOWN предусмотрены исключительно для соединения несущих каркасов HIMax между собой.

3.4.1 Свойства интерфейсов Ethernet

Каждый модуль COM и ЦПУ оснащен сетевым коммутатором Ethernet с IP-адресом и четырьмя портами на соответствующей плате сопряжения.

Сетевой коммутатор Ethernet модуля ЦПУ или COM создает целенаправленное соединение между двумя участниками коммуникации для передачи данных. Это предотвращает возникновение конфликта и разгружает сеть.

Для целенаправленной передачи данных создается таблица присвоения MAC-/IP-адресов (ARP-Cache), и MAC-адреса привязываются к определенным IP-адресам. Пакеты данных передаются теперь только на IP-адреса, которые перечислены в ARP-Cache.

i

Замена модуля ЦПУ/COM с одинаковым IP-адресом

Если происходит замена модуля ЦПУ/COM, для которого настроено *ARP Aging time* = 5 минут и *MAC-Learning* = консервативно, то участник коммуникации получает новый MAC-адрес только через 5-10 минут. В течение этого времени связь через замененный модуль ЦПУ/COM невозможна.

Наряду с настроенным временем *ARP Aging time* необходимо выждать минимум неизменяемое время *MAC Aging-Time* сетевого коммутатора (прибл. 10 секунд), прежде чем снова будет возможна связь через замененный модуль ЦПУ/COM.

Программное обеспечение ARP Aging time: Возможность параметрирования

Свойство	Модуль ЦПУ	Модуль COM
Порты	4	4
Стандарт передачи	10/100/1000 Base-T, полу- и полнодуплексный режим	10/100 Base-T, полу- и полнодуплексный режим
Auto Negotiation (Автосогласование)	Да	Да
Auto-Crossover (Автопересечение)	Да	Да
Гнездо подключения	RJ 45	RJ 45
IP-адрес	Конфигурируется свободно ¹⁾	Конфигурируется свободно ¹⁾
Маска подсети	Конфигурируется свободно ¹⁾	Конфигурируется свободно ¹⁾
Поддерживаемые протоколы	safeethernet Программирующее устройство (PADT) SNTP	safeethernet Программирующее устройство (PADT) TCP-SR, SNTP, Modbus TCP/UDP

Таблица 13: Свойства интерфейсов Ethernet

¹⁾ При назначении IP-адресов и масок подсети должны соблюдаться общепринятые правила.

3.4.2 Конфигурация интерфейсов Ethernet

Конфигурация интерфейса Ethernet осуществляется в детальном виде модуля ЦПУ или COM.

Для систем управления HIMax параметры *Speed Modus* и *Flow-Control Mode* по умолчанию настроены на AutoNeg.

Откройте детальный вид модуля связи:

1. Выберите в структуре **Configuration, Resource, Hardware**.
2. Щелкнуть правой кнопкой мыши на **Hardware** и в контекстном меню выбрать Edit, чтобы открыть редактор аппаратного обеспечения.
3. Щелкнуть правой кнопкой мыши на **Communication Module** и выбрать в контекстном меню **Detail View**, чтобы открыть детальный вид.

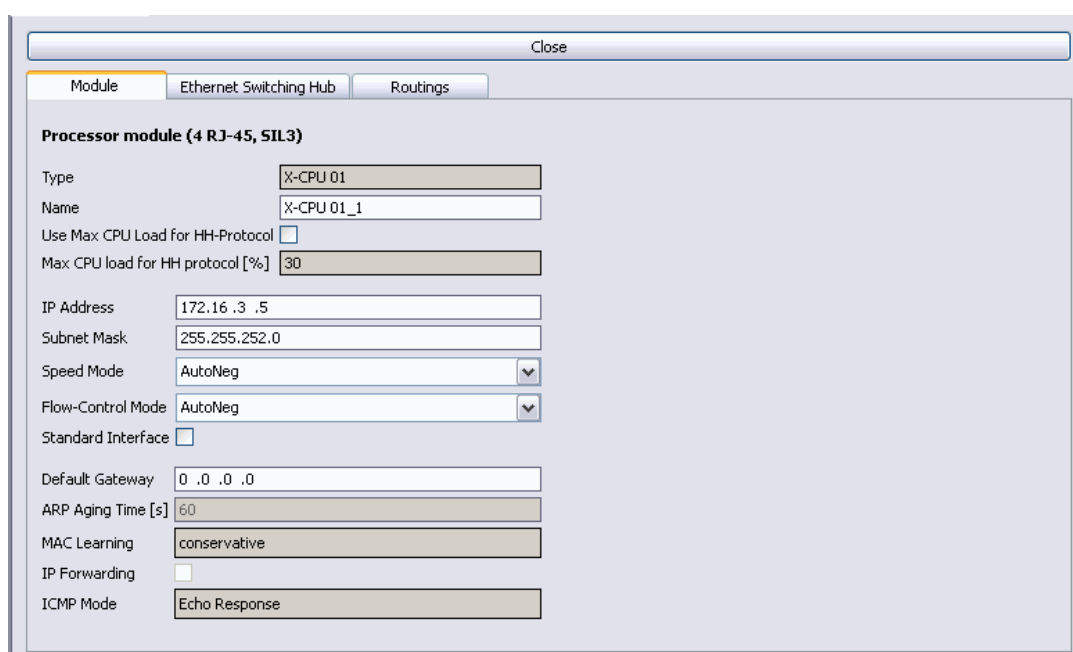


Рисунок 1: Диалоговое окно конфигурации для модуля ЦПУ и COM в SILworX

i

Записи в свойствах модулей COM и ЦПУ Вы должны компилировать заново с помощью прикладной программы и перенести их в систему управления, чтобы они стали действительными для осуществления связи с системой HIMax.

Модуль

Элемент	Описание
Name	Имя модуля связи.
Use Max CPU Load for HH-Protocol	<ul style="list-style-type: none"> Активировано: Сохранить предельное значение нагрузки ЦПУ из поля <i>Max. CPU Load [%]</i>. Деактивировано: Не использовать предельное значение нагрузки ЦПУ для safeethernet.
Max. CPU Load for HH Protocol [%]	<p>Максимальная нагрузка модуля ЦПУ, которая может создаваться при обработке протокола safeethernet.</p> <hr/> <p>i Максимальная нагрузка должна распределяться по всем используемым протоколам, которые используют данный модуль связи.</p>
IP address	IP-адрес интерфейса Ethernet
Subnet Mask	32-битовая маска адреса для классификации IP-адреса в адресе сети и хоста.
Speed Mode	<hr/> <p>i Разрешена только настройка Autoneg! При других настройках модуль перейдет в режим STOP.</p> <hr/>
Flow-Control Mode	
Standard interface	Активировано: Интерфейс используется как стандартный интерфейс для входа в систему.
Default gateway	IP-адрес шлюза по умолчанию
Activate Extended Settings	Использовать параметры <i>ARP Aging Time [s]</i> , <i>MAC Learning</i> и <i>IP Forwarding</i> .
ARP Aging Time [s]	<p>Модуль ЦПУ или COM сохраняет MAC-адреса участников коммуникации в таблице присвоения MAC-/IP-адресов (ARP-Cache).</p> <p>Если в промежуток времени от 1 до 2 <i>ARP Aging Time</i> поступают сообщения</p> <ul style="list-style-type: none"> от участника коммуникации, то MAC-адрес сохраняется в ARP-Cache. Если от участника коммуникации не поступает сообщений, то MAC-адрес удаляется из ARP-Cache. <p>Типичное значение для <i>ARP Aging Time</i> в локальной сети составляет 5 с...300 с.</p> <p>Пользователь не может считать содержание ARP-Cache.</p> <p>При использовании маршрутизаторов или шлюзов отрегулировать <i>ARP Aging Time</i> в соответствии с дополнительными задержкам для обоих направлений пути (увеличить).</p> <p>При недостаточном значении <i>ARP Aging Time</i> модуль ЦПУ/COM удаляет MAC-адрес участника коммуникации из ARP-Cache, и связь осуществляется с задержкой или прерывается.</p> <p>Диапазон значений: 1 с...3600 с Значение по умолчанию: 60 с</p>

MAC Learning	<p>С помощью MAC Learning и <i>ARP Aging Time</i> пользователь настраивает, насколько быстро должен обучаться MAC-адрес.</p> <p>Возможны следующие свойства:</p> <ul style="list-style-type: none"> ▪ Консервативно (рекомендуется): Если в ARP-Cache уже имеются MAC-адреса участников коммуникации, то эти записи блокируются минимум на 1 x <i>ARP Aging Time</i>, максимум на 2 x <i>ARP Aging Time</i>, и не могут быть заменены другими MAC-адресами. Благодаря этому гарантируется, что пакеты данных не будут умышленно или неумышленно передаваться посторонним участникам сети (ARP spoofing). ▪ Допуск: При получении сообщения IP-адрес в сообщении сравнивается с данными в ARP-Cache и сохраненный MAC-адрес в ARP-Cache сразу переписывается на MAC-адрес из сообщения. Используйте настройку Допуск, если наличие связи важнее, чем надежный доступ (authorized access) к системе управления.
IP Forwarding	<p>Позволяет модулю ЦПУ или COM работать маршрутизатором и передавать пакеты данных других узлов сети.</p> <ul style="list-style-type: none"> ▪ Активировано: Передача данных включена. ▪ Деактивировано: Передача данных отключена.
ICMP Mode	<p>Межсетевой протокол управления сообщениями (ICMP) позволяет более высокому уровню протокола распознавать состояния ошибок на сетевом уровне и оптимизировать передачу пакетов данных.</p> <p>Типы сообщений меж сетевого протокола управления сообщениями (ICMP), поддерживаемые модулем ЦПУ:</p> <ul style="list-style-type: none"> ▪ Нет ответа ICMP Все команды ICMP отключены. Благодаря этому обеспечивается высокая степень защиты от несанкционированного доступа, который может быть осуществлен через сеть. ▪ Echo Response Если включена функция Echo Response, то узел отвечает на команду Ping. Таким образом, можно определить, что узел доступен. Степень защиты все еще остается высокой. ▪ Хост не доступен Для пользователя не имеет значения. Только для испытаний, проводимых изготовителем. ▪ Все реализованные ответы ICMP Все команды ICMP включены. Благодаря этому обеспечивается более точная диагностика ошибок при возникновении сбоев в сети.

Таблица 14: Параметр конфигурации

Сетевой коммутатор Ethernet

Элемент	Описание
Port	Номер порта, как на корпусе; каждый порт может иметь только одну конфигурацию. Диапазон значений: 1...4
Speed [Mbit/s]	10 Мбит/с: Скорость передачи данных 10 Мбит/с 100 Мбит/с: Скорость передачи данных 100 Мбит/с 1000 Мбит/с: Скорость передачи данных 1000 Мбит/с (модуль ЦПУ) Autoneg (10/100/1000): автоматическая настройка скорости передачи в бодах Значение по умолчанию: Autoneg
Flow control	Полнодуплексный режим: связь в обоих направлениях одновременно Полудуплексный режим: связь в одном направлении Autoneg: автоматические управления связью Значение по умолчанию: Autoneg
Autoneg also with fixed values	«Advertising» (передача свойств скорости и управления потоком данных) выполняется также при фиксированных установленных значениях скорости и управления потоком данных. Благодаря этому другие устройства, порты которых настроены на <i>Autoneg</i> , распознают настройку порта HiMax.
Limit	Ограничение входящих групповых и широковещательных пакетов. Выкл: без ограничения Широковещательная рассылка: ограничить широковещательную рассылку (128 кбит/с) Групповая и широковещательная рассылка: ограничить групповую и широковещательную рассылку (1024 кбит/с) Значение по умолчанию: Широковещательная рассылка

Таблица 15: Параметр сетевого коммутатора Ethernet

Маршруты

Элемент	Описание
Name	Обозначение настройки маршрутизации
IP Address	Целевой IP-адрес участника коммуникации (при прямом маршруте к хосту) или сетевой адрес (при маршруте к подсети) Диапазон значений: 0.0.0.0...255.255.255.255 Значение по умолчанию: 0.0.0.0
Subnet Mask	Определяет диапазон целевого адреса для записи маршрута. 255.255.255.255 (при прямом маршруте до хоста) или маска адресованной подсети. Диапазон значений: 0.0.0.0...255.255.255.255 Значение по умолчанию: 255.255.255.255
Gateway	IP-адрес шлюза к адресованной сети. Диапазон значений: 0.0.0.0...255.255.255.255 Значение по умолчанию: 0.0.0.1

Таблица 16: Параметры маршрута

3.4.3 Используемые сетевые порты для связи Ethernet

Порты UDP/использование

8000	Программирование и управление при помощи SILworX
8001	Конфигурация удаленного устройства ввода/вывода посредством ПЭС
6010	safeethernet и OPC
123	SNTP (синхронизация по времени между программируемой электронной системой и устройством удаленного ввода/вывода, а также внешними устройствами)
8895	Modbus – ведущее устройство – UDP, если конфигурировано
502	Ведомое устройство Modbus (изменяется пользователем)

Порты TCP/использование

8895	Ведущее устройство Modbus
502	Ведомое устройство Modbus (изменяется пользователем)
Xxx	TCP-SR задается пользователем

3.5 Интерфейсы полевой шины

Связь с внешними системами может осуществляться через интерфейсы полевой шины модуля COM. Для каждого интерфейса полевой шины возможен только один протокол.

Интерфейсы полевых шин должны быть оснащены submodule полевой шины. Без submodule полевой шины невозможна связь через данный интерфейс. Submodule полевой шины определяет стандарт передачи для интерфейса.

Обозначение	Интерфейс	Протокол
FB 1	Гнездо SUB-D 9-пол.	PROFIBUS Master PROFIBUS Slave Modbus Master RS485 Modbus Slave RS485 COM USER Task (RS 232)
FB 2	Гнездо D-Sub 9-пол.	PROFIBUS Master PROFIBUS Slave Modbus Master RS485 Modbus Slave RS485 COM USER Task (RS 232)

Таблица 17: Интерфейсы полевой шины

3.5.1 Назначение штырьковых выводов гнезд D-SUB FB1 и FB2



Назначение штырьковых выводов интерфейсов полевой шины зависит от используемого submodule полевой шины и показано в таблицах ниже.

Субмодуль полевой шины для ведущего или ведомого устройства PROFIBUS DP

Штырьковый вывод	Сигнал	Функция
1	---	---
2	---	---
3	RxD/TxD-A	Принятые/переданные данные A
4	RTS	Управляющий сигнал
5	DGND	Опорный потенциал данных
6	VP	+5 В питающее напряжение
7	---	---
8	RxD/TxD-B	Принятые/переданные данные B
9	---	---

Таблица 18: Назначение штырьковых выводов подключений D-SUB FB1 и FB2 для PROFIBUS DP

Субмодуль полевой шины RS485 для ведущего или ведомого устройства Modbus

Штырьковый вывод	Сигнал	Функция
1	---	---
2	RP	5 В, разъединяется с помощью диодов
3	RxD/TxD-A	Принятые/переданные данные A
4	CNTR-A	Управляющий сигнал A
5	DGND	Опорный потенциал данных
6	VP	+5 В питающее напряжение
7	---	---
8	RxD/TxD-B	Принятые/переданные данные B
9	CNTR-B	Управляющий сигнал B

Таблица 19: Назначение штырьковых выводов подключений D-SUB FB1 и FB2 для Modbus

Субмодуль полевой шины RS232 для COM USER Task

Штырьковый вывод	Сигнал	Функция
1	---	---
2	TxD	Переданные данные
3	RxD	Принятые данные
4	---	---
5	DGND	Опорный потенциал данных
6	---	---
7	RTS	Запрос на отправку
8	---	---
9	---	---

Таблица 20: Назначение штырьковых выводов подключений D-SUB FB1 и FB2 для RS232

3.5.2 Протоколы и интерфейсы

Относящаяся к X-COM 01 плата сопряжения X-CB 001 02 оснащена следующими интерфейсами:

- 4 интерфейса Ethernet
- 2 интерфейса полевой шины

В зависимости от интерфейса Вы можете активировать протоколы, указанные в таблице, см. таблицу 600 **ху**

х: Опция для .

[1]. **safeethernet** встроено во всех системах управления.

[2]. Для ведущего устройства PROFIBUS и ведомого устройства PROFIBUS активация осуществляется путем установки субмодуля полевой шины.

[3]. Для субмодуля RS485 (Modbus RS485) и субмодуля RS232 (COM User Task) Вы должны дополнительно получить код активации программы для выбранного протокола полевой шины.

Вы можете испытывать данные протоколы полевой шины без лицензии в течение 5000 часов работы.

[4]. Для данных протоколов Ethernet Вы должны получить код активации программы. Вы можете испытывать данные протоколы Ethernet без лицензии в течение 5000 часов работы.

ПРЕДУПРЕЖДЕНИЕ



Прикладная программа не может больше запускаться без лицензии!

По истечении 5000 часов работы связь не прерывается, пока система управления не будет остановлена. После этого прикладная программа не запускается без действительной лицензии для проектных протоколов (недействительная конфигурация).

Запросите своевременно код активации программы!

Протокол	Интерфейсы	Активация
HIMA safeethernet	Ethernet	[1]
Modbus TCP Master	Ethernet	[4]
Modbus TCP Slave	Ethernet	[4]
TCP Send/Receive	Ethernet	[4]
SNTP Server/Client	Ethernet	[4]
PROFIBUS DP Master	FB1 и FB2	[2]
PROFIBUS DP Slave	FB1 и FB2	[2]
Modbus Master RS485	FB1 и FB2	[3]
Modbus Slave RS485	FB1 и FB2	[3]

Таблица 21: Протоколы связи

3.5.3 Структура номера модуля COM

Номер модуля COM изменяется в результате оснащения субмодулей полевой шины следующим образом:

98 50600 **xy**

x: Опция для субмодулей полевой шины 1 (FB1)

y: Опция для субмодулей полевой шины 2 (FB2)

Опциональные значения для x и y:

0: Без субмодулей полевой шины

1: RS485 (ведущее устройство Modbus, ведомое устройство Modbus определяется по коду лицензии)

2: Ведущее устройство PROFIBUS DP

3: Ведомое устройство PROFIBUS DP

i

HIMA рекомендует эксплуатировать PROFIBUS DP через интерфейс полевой шины FB1 (скорость передачи макс. 12 Мбит/с). Через интерфейс полевой шины FB2 максимально допустимая скорость передачи составляет 1,5 Мбит/с.

Примеры:

Номер изделия	Субмодули полевой шины 1 (FB1)	Субмодули полевой шины 2 (FB2)
98 50600 21	Ведущее устройство PROFIBUS	RS485
98 50600 23	Ведущее устройство PROFIBUS	Ведомое устройство PROFIBUS (макс. 1,5 Мбит/с)
98 50600 11	RS485	RS485
98 50600 00	-	-

Таблица 22: Примеры для номеров изделия модуля COM

⚠ ОСТОРОЖНО



Ненадлежащее открывание модуля COM

Повреждение модуля COM

Оснащение модулей связи субмодулями полевой шины осуществляется исключительно компанией HIMA.

4 safeethernet

Все системы HIMax подходят для **safeethernet**. Они могут соединяться безопасно на уровне совокупной безопасности 3 через Ethernet (до 1 Гбит/с).

Соответствующие интерфейсы Ethernet систем управления HIMax параллельно можно использовать для других протоколов.

Связь **safeethernet** между системами управления осуществляется через различные топологии сети Ethernet. Настройте параметры протокола **safeethernet** в соответствии с используемой сетью Ethernet, чтобы повысить скорость и эффективность передачи данных.

Данные параметры можно настроить с помощью так называемых сетевых профилей. Заводская настройка параметров обеспечивает надежную связь, поэтому пользователю не придется сначала вникать в подробности конфигурации сети.



Протокол **safeethernet** представляет собой безопасный протокол и имеет сертификат TÜV до уровня совокупной безопасности 3 согласно МЭК 61508.

Необходимое оснащение и требования к системе:

Система управления HIMAX	HIMax с модулем ЦПУ
Активация	Данная функция по умолчанию разблокирована во всех системах HIMax.

safeethernet (свойства):

Процессорные модули	safeethernet реализуется на модуле ЦПУ. Максимально 4 модуля ЦПУ для одной системы управления HIMax.
Соединения	Максимально 255 соединений safeethernet для одной системы управления HIMax.
Маршрут	Интерфейсы Ethernet модулей ЦПУ и COM Используемые интерфейсы Ethernet параллельно можно использовать для других протоколов.
Избыточные маршруты	2-х канальная эксплуатация Избыточные маршруты safeethernet между HIMax и HIMax настраиваются в редакторе safeethernet .
Набор данных процесса	Через каждое соединение safeethernet можно в каждое направление передавать максимум 1100 байтов. <div> <div>i</div> <div>Максимальный набор данных процесса системы управления равный 512 кБ не должен превышать. В этом случае при загрузке будет отклонено параметрирование системы управления.</div> </div>
Межпроектная связь	Соединения safeethernet с ресурсом в другом проекте можно конфигурировать в SILworX, см. главу 4.9.

4.1 Что представляет собой safeethernet

В сфере автоматизации производства и процессов основными являются такие понятия как детерминизм, надежность, взаимозаменяемость, способность к расширению и, прежде всего, безопасность.

safeethernet представляет собой протокол для безопасной передачи данных до уровня совокупной безопасности 3 на основе технологии Ethernet.

safeethernet включает в себя механизмы, которые распознают следующие ошибки и реагируют на них безопасно:

- искажение переданных данных (удвоенные, потерянные или измененные биты)
- неверная адресация сообщений (отправитель, получать)
- неверная последовательность данных (повторение, потеря, обмен)
- неверная временная характеристика (задержка, эхо)

В основе **safeethernet** лежит стандарт IEEE 802.3.

При передаче релевантных для безопасности данных используется стандартный протокол Ethernet.

safeethernet использует «ненадежные каналы передачи данных» (Ethernet) по принципу «черного канала» и контролирует их у отправителя и получателя с помощью безопасных механизмов протокола. Благодаря этому в безопасной сети можно использовать такие компоненты сети Ethernet как сетевые концентраторы, сетевые коммутаторы, маршрутизаторы.

Существенным отличием от стандартной связи Ethernet является безопасность и возможность обработки в режиме реального времени **safeethernet**. Специальный механизм протокола гарантирует детерминированное поведение, в том числе при сбое или вступлении новых участников коммуникации. В этом случае система автоматически вносит новые компоненты в работающую систему. Во время работы может осуществляться обмен любыми компонентами. С помощью сетевого коммутатора можно точно определить время передачи данных. Таким образом, Ethernet может выполнять обработку в режиме реального времени.

Возможная скорость передачи, достигающая 1 Гбит/с для безопасных данных, выходит за рамки стандартной. В качестве среды передачи Вы можете использовать, например, медные провода и световоды.

С помощью **safeethernet** возможны соединения с внутрифирменным интранетом, а также соединения с интернетом. Таким образом, для безопасной и небезопасной передачи данных требуется только одна сеть.

i

Сеть может использоваться также другими участниками, если это позволяет пропускная способность канала.

safeethernet делает возможными гибкие системные структуры для децентрализованной автоматизации с определенным временем реакции. В зависимости от запроса Вы можете на выбор распределять данные централизованно или децентрализованно по участникам внутри сети.

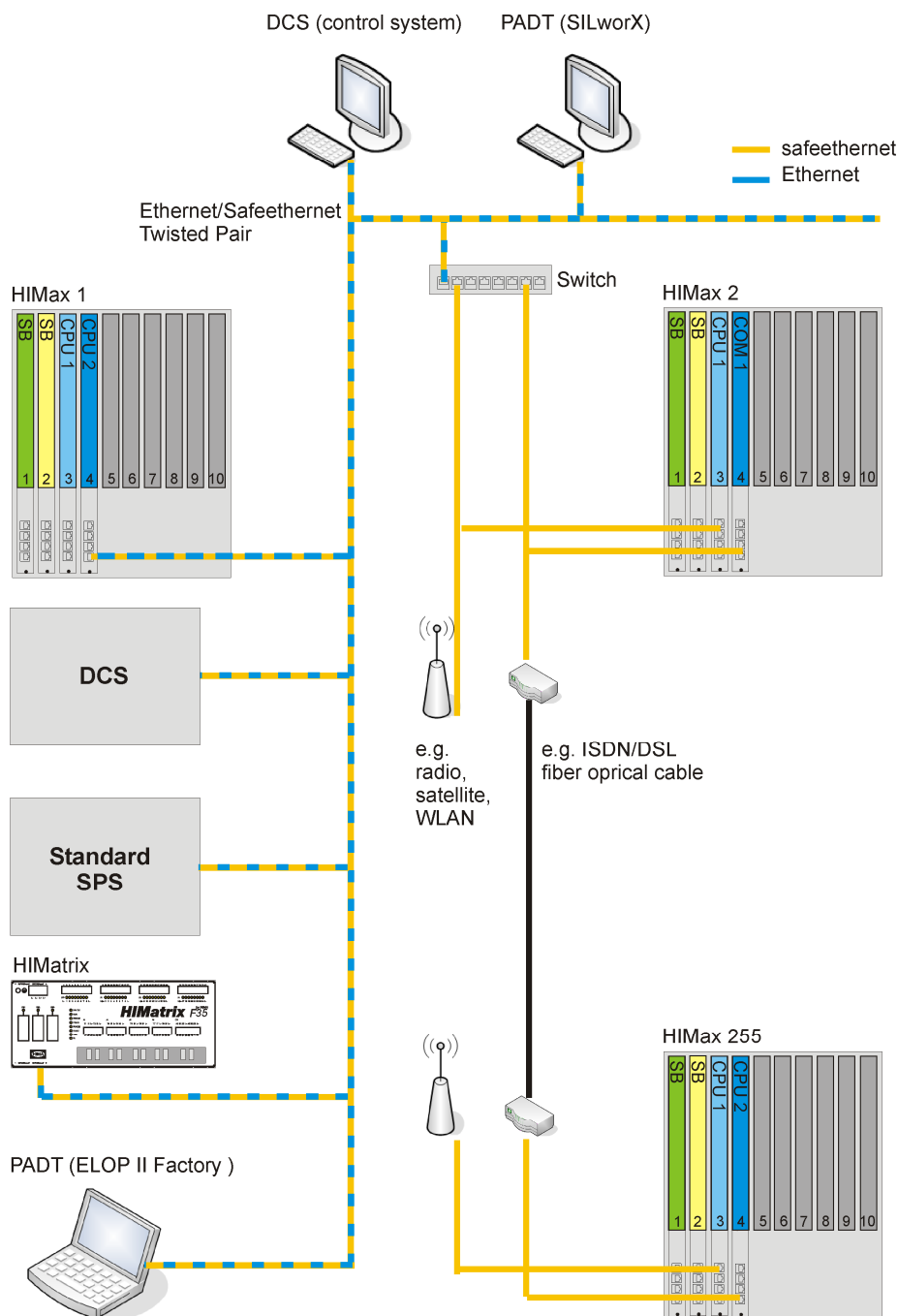


Рисунок 2: Системные структуры

i

Возможен непреднамеренный переход в безопасное состояние!

При внутреннем соединении следить за тем, чтобы не образовывались сетевые петли. Пакеты данных должны попадать в систему управления по одному пути.

4.2 Конфигурация избыточного соединения safeethernet

На этом примере Вы можете конфигурировать избыточное соединение HIMA/HIMA safeethernet.

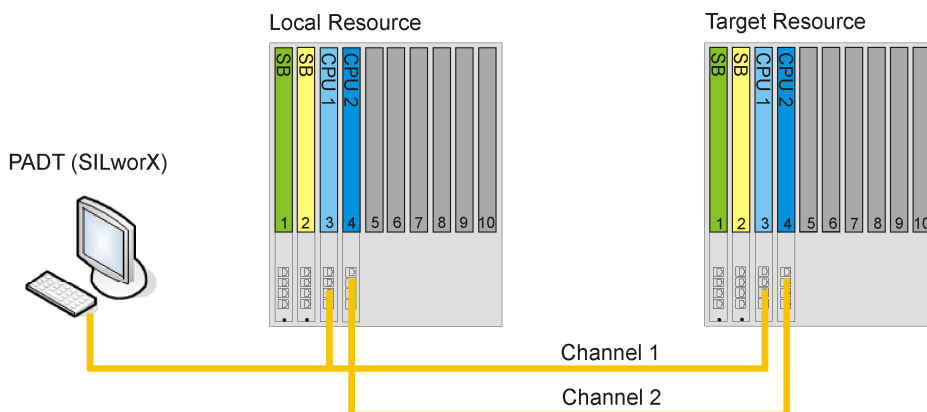


Рисунок 3: Создание конфигурации избыточного соединения

i

HIMA рекомендует при избыточном соединении safeethernet провести оба маршрута (канал 1 и канал 2) через отдельные модули связи. При этом пропускная способность и задержка на соответствующих маршрутах должны быть примерно одинаковыми.

Создание соединения safeethernet

Создайте в редакторе safeethernet соединение safeethernet между локальным и целевым ресурсом.

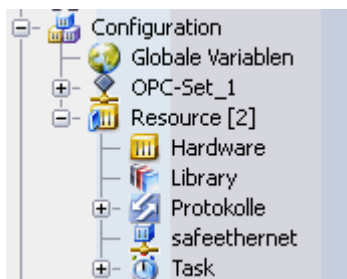


Рисунок 4: Структура ресурса

Чтобы открыть редактор safeethernet локального ресурса, необходимо:

1. Открыть в структуре **Configuration, Resource**.
2. Щелкнув правой кнопкой мыши на **safeethernet**, выберите контекстное меню **Edit**.
☒ В выборе объекта находятся целевые ресурсы.

Чтобы создать соединение safeethernet с target resource, необходимо:

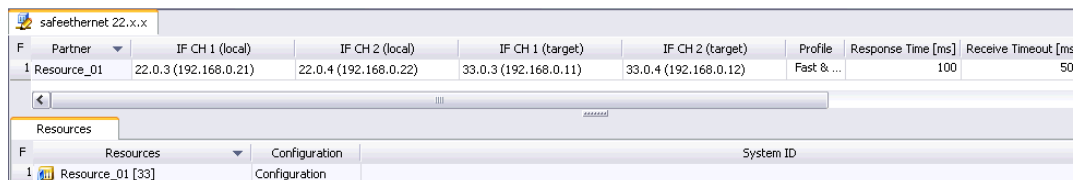
1. В выборе объекта щелкнуть на **Target Resource** и с помощью Drag&Drop перетащить на свободное место в рабочей области редактора safeethernet.

i

Обратный путь связи в редакторе safeethernet целевого ресурса добавляется автоматически.

Чтобы конфигурировать соединение **safeethernet**, необходимо:

1. Выбрать **Ethernet interfaces Channel 1** локального и целевого ресурса.
2. Выбрать **Ethernet interfaces Channel 2** локального и целевого ресурса.
3. Выбрать **Network Profile** (напр., Fast&Noisy) соединения **safeethernet**.
4. Рассчитать и внести **Receive Timeout** и **Response Time** (см. главу 4.6.1).



F	Partner	IF CH 1 (local)	IF CH 2 (local)	IF CH 1 (target)	IF CH 2 (target)	Profile	Response Time [ms]	Receive Timeout [ms]
1	Resource_01	22.0.3 (192.168.0.21)	22.0.4 (192.168.0.22)	33.0.3 (192.168.0.11)	33.0.4 (192.168.0.12)	Fast & ...	100	500

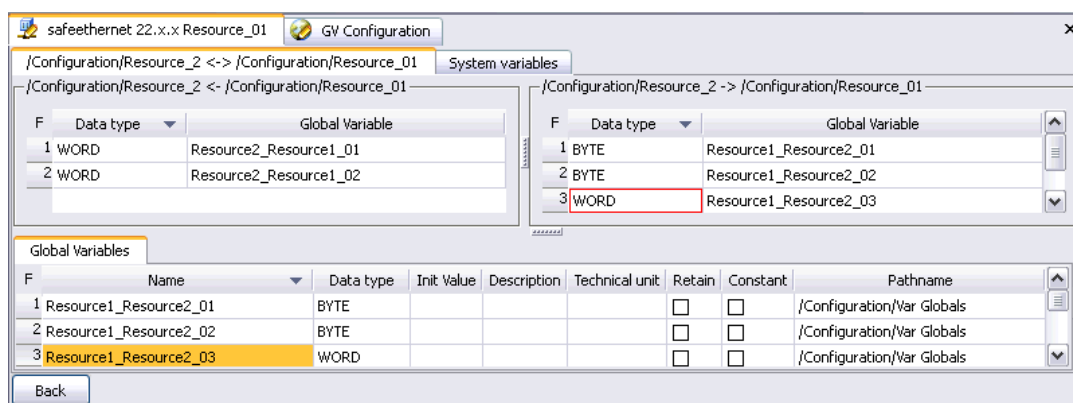
Рисунок 5: Значения параметров избыточного соединения **safeethernet**

Соединение переменных процесса

Чтобы открыть детальный вид соединения **safeethernet**, необходимо:

Условие: Редактор **safeethernet** локального ресурса открыт.

1. Открыть контекстное меню **Target Resource** с помощью правой кнопки мыши.
2. Выбрать **Detail View**.
3. Выбрать вкладку **Resource (target)<->Resource (local)**.



F	Data type	Global Variable
1	WORD	Resource2_Resource1_01
2	WORD	Resource2_Resource1_02
3	WORD	Resource1_Resource2_03

Рисунок 6: Детальный вид редактора **safeethernet**

i

Можно использовать только глобальные переменные из контекста конфигурации, а не из контекста ресурса!

Чтобы добавить переменные передачи **safeethernet**, необходимо:

Переменные передачи передаются с локального ресурса на целевой ресурс

1. Выбрать область **Resource (local)->Resource (target)**.
2. В выборе объекта выбрать **Global Variable** и с помощью Drag&Drop перетащить в столбец **Resource (target)->Resource (local)**.
3. Повторить данное действие для других переменных передачи **safeethernet**.

Чтобы добавить переменные приема safeethernet, необходимо:

Переменные приема принимаются с локального ресурса.

1. Выбрать область **Resource (target)<-Resource (local)**.
2. В выборе объекта выбрать **Global Variable** и с помощью Drag&Drop перетащить в столбец **Resource (target)<-Resource (local)**.
3. Повторить данное действие для других переменных приема safeethernet.

Подтверждение соединения safeethernet:

1. Открыть в структуре **Configuration, Resource, safeethernet**.
2. Щелкнуть на кнопку **Verification** в строке меню и подтвердить с помощью **OK**.
3. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.

i

Конфигурация соединения safeethernet должна быть скомпилирована заново с помощью прикладной программы локального и целевого ресурса и перенесена в системы управления, прежде чем она будет активна для связи HIMax.

Проверка связи safeethernet

В панели управления установите на нуль индикации *Bad Messages* и *Resends*.

1. Эксплуатируйте систему HIMax с полной нагрузкой:
 - Все протоколы связи работают (safeethernet и стандартные протоколы).
 - Вынуть процессорные модули и снова вставить как описано в главе 0.
 - Загрузка прикладной программы через перезагрузку.

i

Для проверки создания избыточного соединения safeethernet Вы должны отсоединить одно избыточное соединение и снова добавить, а затем другое. При этом в связи safeethernet не должно возникнуть ошибки.

2. Проверьте на панели управления обеих систем управления индикации *Bad Messages* и *Resends*.
 Если показания *Bad Messages* и *Resends*
 = 0, то настройки safeethernet в порядке.
 ≥ 0, настройки safeethernet необходимо проверить еще раз.
 - Рассчитайте заново с помощью максимального времени цикла значение *Receive Timeout*, см. главу 4.6.1 и 4.6.2.
 - Изменяйте *Response Time*, как описано в главе 4.6.3.

i

Другие причины для *Bad Messages* и *Resends*!

Проверьте правильность создания сети (напр., кабели, сетевые коммутаторы, компьютеры).
 Если сеть Ethernet используется неэксклюзивно для safeethernet, то необходимо проверить нагрузку сети (вероятность конфликтов данных).

4.3 Редактор safeethernet

В редакторе safeethernet Вы можете создавать и конфигурировать соединения safeethernet с участниками коммуникации (ресурсами).

Чтобы открыть редактор safeethernet local resource, необходимо:

1. Открыть в структуре **Configuration, Resource**.
2. Щелкнув правой кнопкой мыши на **safeethernet**, выберите контекстное меню **Edit**.
☒ Редактор safeethernet содержит рабочую область и выбор объекта.

В редакторе safeethernet Вы можете создавать и конфигурировать соединения safeethernet с участниками коммуникации (ресурсами). Для этого перетащите ресурсы из выбора объекта в рабочую область.

Для конфигурации соединения safeethernet Вы должны настроить следующие параметры протокола safeethernet:

Параметр	Описание
Partner	Имя ресурса участника
IF CH...	Доступные интерфейсы Ethernet на локальном и целевом ресурсе, см. также главу 3.4.
Profile	Комбинирование подходящих параметров safeethernet, см. также главу 4.8.
Sync/Async	Обмен данными можно настроить синхронно или асинхронно по отношению к циклу ЦПУ, см. также главу 4.6.4.
Response Time [ms]	Время до подтверждения принятия сообщения у отправителя, см. также главу 4.6.3.
Receive Timeout [ms]	Время контроля на ПЭС 1, в течение которого должен быть принят правильный ответ от ПЭС 2, см. также главу 4.6.2.
Resend Timeout [ms]	Время контроля на ПЭС 1, в течение которого ПЭС 2 должна подтвердить прием пакета данных, см. также главу 4.6.2.
Acknowledge Timeout [ms]	Максимальное время, после которого принятый пакет данных должен быть подтвержден ЦПУ, см. также главу 4.6.6.
Prod.-Rate	Интервал передачи пакетов: Минимальный интервал времени между двумя пакетами данных, см. также главу 4.6.7.
Queue	Количество пакетов данных, отправленных без подтверждения приема, см. также главу 4.6.8.


Freeze values on lost connection [ms]	<p>Поведение входных переменных данного соединения safeethernet при разрыве соединения.</p> <p>Use default values Для входных переменных используются исходные данные.</p> <p>Unlimited Входные переменные остаются на текущем значении и используются до восстановления соединения.</p> <p>Limited Ввод: Двойной щелчок на выпадающее поле и ввод времени.</p> <p>Входные переменные остаются на текущем значении и используются до параметризованного времени ожидания. Затем используются исходные данные.</p> <p>Запуск времени ожидания может быть задержан на один цикл ЦПУ.</p>
	<p>⚠ ОСТОРОЖНО</p> <p> Для безопасных функций, которые реализуются через safeethernet, разрешается использовать только настройку Use Initial Data.</p>
Fragments per cycle	Настройка для соединения safeethernet с сервером X-OPC. Фиксированная настройка: вид (1 фрагмент = 1100 байтов) передается за один цикл HIMax на сервер X-OPC.
Command Inbudget	Настройка для соединения safeethernet с сервером X-OPC.
Command Outbudget	<p>Настройка, как часто принимается данная команда (событие) по отношению к другим видам.</p> <p>Для определения приоритета передачи данных определенного вида через соединение safeethernet каждому виду присваивается бюджет. Виды с бюджетом n и виды с бюджетом m отправляются в соотношении n к m. Аналогичное действует для отправления команд.</p>
Ignored warning count	Функция отсутствует!
Warning period [ms]	Функция отсутствует!
Enable SER	Значение по умолчанию: активировано.

Таблица 23: Параметр Протокол **safeethernet**

Выбор объекта

Выбор объекта предоставляет все ресурсы внутри данного проекта, с которыми данный ресурс может быть соединен через **safeethernet**.

i

Для соединений **safeethernet** с ресурсами вне проекта или с (спроектированной в ELOP II Factory) системой управления HIMatrix имеется функция экспорта (см. главу 4.9).

4.4 Детальный вид редактора safeethernet

Детальный вид всегда имеет ссылку на локальный ресурс, для которого Вы запустили редактор safeethernet.

Чтобы открыть Детальный вид соединения safeethernet, необходимо:

1. Щелкнуть правой кнопкой мыши на соединение safeethernet и открыть контекстное меню.
2. Щелкнуть на **Detail View**.
 - ☒ **Детальный вид** содержит вкладку **System Variables** и **Resource (local) <->Resource (target)**.

Вкладка: Системная переменная

Вы можете регулировать соединение safeethernet в прикладной программе с помощью системных переменных и анализировать его состояние.

Системная переменная	Описание										
Ack. Frame No.	Счетчик приема (актуальн.).										
Number of bad messages	Количество всех поврежденных сообщений на канал (неверный CRC, неверный заголовок, прочие ошибки).										
Number of bad messages for the red. channel											
Number of successful connections	Количество успешных соединений с момента сброса статистики.										
Number of lost messages	Количество неудачных сообщений на одном из двух маршрутов с момента сброса статистики. Счетчик ведется только до полного сбоя в канале.										
Number of lost messages for the red. channel											
Early Queue Usage	Количество сообщений, которые были созданы в первой очереди с момента сброса статистики, см. также главу 4.6.8.										
Bad messages	Количество отклоненных сообщений с момента сброса статистики.										
Frame no.	Счетчик передачи (актуальный).										
Channel state	<p>Текущее состояние канала 1. Состояние канала – это текущее состояние канала 1 в момент (поряд. № X-1) принятия сообщения с поряд. № X.</p> <table> <tr> <th>Состояние</th><th>Описание</th></tr> <tr> <td>0</td><td>Информация по состоянию канала 1 отсутствует.</td></tr> <tr> <td>1</td><td>Канал 1 в порядке.</td></tr> <tr> <td>2</td><td>Последнее сообщение содержало ошибку, текущее в порядке.</td></tr> <tr> <td>3</td><td>Ошибка в канале 1.</td></tr> </table>	Состояние	Описание	0	Информация по состоянию канала 1 отсутствует.	1	Канал 1 в порядке.	2	Последнее сообщение содержало ошибку, текущее в порядке.	3	Ошибка в канале 1.
Состояние	Описание										
0	Информация по состоянию канала 1 отсутствует.										
1	Канал 1 в порядке.										
2	Последнее сообщение содержало ошибку, текущее в порядке.										
3	Ошибка в канале 1.										

Системная переменная	Описание																		
Layout Version	Сигнатура используемого в связи формата данных.																		
Last channel latency	Латентность канала указывает время задержки между обоими избыточными маршрутами по отношению ко времени приема сообщений с идентичным поряд. №. Для этого ведется статистика со средней, минимальной и последней латентностью. Если минимальное значение > максимального значения, то статистические значения не действительны. Латентность последнего канала и латентность среднего канала равняется в этом случае 0.																		
Last latency of the red. channel																			
Max. channel latency																			
Max. latency of the red. channel																			
Min. channel latency																			
Min. latency of the red. channel																			
Average channel latency																			
Average latency of the red. channel																			
Monotony	Счетчик передачи полезной нагрузки (актуальн.).																		
New Layout Version	Сигнатура нового формата данных.																		
Quality of Channel 1	<div>Состояние главного маршрута.<table><tr><th>Бит №</th><th>Бит = 0</th><th>Бит = 1</th></tr><tr><td>0</td><td>Маршрут не доступен</td><td>Маршрут доступен</td></tr><tr><td>1</td><td>Маршрут не используется</td><td>Маршрут активно используется</td></tr><tr><td>2</td><td>Маршрут не соединен</td><td>Маршрут соединен</td></tr><tr><td>3</td><td>-</td><td>Маршрут сначала доставляет сообщение</td></tr><tr><td>4–7</td><td>Зарезервировано</td><td>Зарезервировано</td></tr></table></div>	Бит №	Бит = 0	Бит = 1	0	Маршрут не доступен	Маршрут доступен	1	Маршрут не используется	Маршрут активно используется	2	Маршрут не соединен	Маршрут соединен	3	-	Маршрут сначала доставляет сообщение	4–7	Зарезервировано	Зарезервировано
Бит №	Бит = 0	Бит = 1																	
0	Маршрут не доступен	Маршрут доступен																	
1	Маршрут не используется	Маршрут активно используется																	
2	Маршрут не соединен	Маршрут соединен																	
3	-	Маршрут сначала доставляет сообщение																	
4–7	Зарезервировано	Зарезервировано																	
Quality of Channel 2	Состояние избыточного маршрута, см. состояние канала 1 (главный маршрут).																		
Receive Timeout	Время в миллисекундах (мс) на ПЭС 1, в течение которого должен быть принят действительный ответ от ПЭС 2, см. также главу 4.6.2.																		
Response Time	Время в миллисекундах (мс) до подтверждения приема сообщения у отправителя, см. также главу 4.6.3																		
Reset safeethernet statistic	Статистические значения для соединения связи в прикладной программе Сброс. Для этого установить параметр <i>Reset Statistic</i> с FALSE на TRUE.																		
Transport Control Ch1	<div>Управление маршрутом канала 1<table><tr><th>Бит 0</th><th>Функция</th></tr><tr><td>FALSE</td><td>Маршрут доступен</td></tr><tr><td>TRUE</td><td>Маршрут заблокирован</td></tr></table> <table><tr><th>Бит 1</th><th>Функция</th></tr><tr><td>FALSE</td><td>Маршрут доступен для тестирования</td></tr><tr><td>TRUE</td><td>Маршрут заблокирован</td></tr></table> Бит 2...7 зарезервирован.</div>	Бит 0	Функция	FALSE	Маршрут доступен	TRUE	Маршрут заблокирован	Бит 1	Функция	FALSE	Маршрут доступен для тестирования	TRUE	Маршрут заблокирован						
Бит 0	Функция																		
FALSE	Маршрут доступен																		
TRUE	Маршрут заблокирован																		
Бит 1	Функция																		
FALSE	Маршрут доступен для тестирования																		
TRUE	Маршрут заблокирован																		
Transport Control Ch2	Управление маршрутом канала 2, см. управление маршрутом канала 1.																		

Системная переменная	Описание								
Connection Control	<p>С помощью данной системной переменной можно осуществлять управление соединением safeethernet в прикладной программе.</p> <table> <tr> <th>Команда</th><th>Описание</th></tr> <tr> <td>Autoconnect (0x0000)</td><td>Значение по умолчанию: После потери связи safeethernet система управления пытается в следующем цикле ЦПУ восстановить соединение.</td></tr> <tr> <td>Toggle Mode 0 (0x0100) Toggle Mode 1 (0x0101)</td><td> <p>После потери связи можно восстановить соединение с помощью смены режима переключения, управляемой программой.</p> <ul style="list-style-type: none"> Установлен TOGGLE MODE_0 (0x100): Установить на TOGGLE MODE 1 (0x101), чтобы восстановить соединение. Установлен TOGGLE MODE 1 (0x101): Установить на TOGGLE_MODE_0 (0x100), чтобы восстановить соединение. </td></tr> <tr> <td>Disabled (0x8000)</td><td>Связь safeethernet отключена.</td></tr> </table>	Команда	Описание	Autoconnect (0x0000)	Значение по умолчанию: После потери связи safeethernet система управления пытается в следующем цикле ЦПУ восстановить соединение.	Toggle Mode 0 (0x0100) Toggle Mode 1 (0x0101)	<p>После потери связи можно восстановить соединение с помощью смены режима переключения, управляемой программой.</p> <ul style="list-style-type: none"> Установлен TOGGLE MODE_0 (0x100): Установить на TOGGLE MODE 1 (0x101), чтобы восстановить соединение. Установлен TOGGLE MODE 1 (0x101): Установить на TOGGLE_MODE_0 (0x100), чтобы восстановить соединение. 	Disabled (0x8000)	Связь safeethernet отключена.
Команда	Описание								
Autoconnect (0x0000)	Значение по умолчанию: После потери связи safeethernet система управления пытается в следующем цикле ЦПУ восстановить соединение.								
Toggle Mode 0 (0x0100) Toggle Mode 1 (0x0101)	<p>После потери связи можно восстановить соединение с помощью смены режима переключения, управляемой программой.</p> <ul style="list-style-type: none"> Установлен TOGGLE MODE_0 (0x100): Установить на TOGGLE MODE 1 (0x101), чтобы восстановить соединение. Установлен TOGGLE MODE 1 (0x101): Установить на TOGGLE_MODE_0 (0x100), чтобы восстановить соединение. 								
Disabled (0x8000)	Связь safeethernet отключена.								
Connection State	<p>Состояние соединения анализирует состояние связи между двумя системами управления в прикладной программе.</p> <table> <tr> <th>Состояние/значение</th><th>Описание</th></tr> <tr> <td>Closed (0)</td><td>Соединение закрыто, и попытка его открыть не производится.</td></tr> <tr> <td>Try_open (1)</td><td>Производится попытка открыть соединение, но оно еще не открыто. Данное состояние действует одинаково как для активной, так и для пассивной стороны.</td></tr> <tr> <td>Connected (2)</td><td>Соединение установлено и эксплуатируется (активный контроль времени и обмен данными).</td></tr> </table>	Состояние/значение	Описание	Closed (0)	Соединение закрыто, и попытка его открыть не производится.	Try_open (1)	Производится попытка открыть соединение, но оно еще не открыто. Данное состояние действует одинаково как для активной, так и для пассивной стороны.	Connected (2)	Соединение установлено и эксплуатируется (активный контроль времени и обмен данными).
Состояние/значение	Описание								
Closed (0)	Соединение закрыто, и попытка его открыть не производится.								
Try_open (1)	Производится попытка открыть соединение, но оно еще не открыто. Данное состояние действует одинаково как для активной, так и для пассивной стороны.								
Connected (2)	Соединение установлено и эксплуатируется (активный контроль времени и обмен данными).								
Repeats	Количество повторов с момента сброса статистики.								
Timestamp for the last fault on the red. channel [ms]	Доля метки времени в миллисекундах (актуальное время системы).								
Timestamp for the last fault on the red. channel [s]	Доля метки времени в секундах (актуальное время системы).								
Timestamp for the last fault [ms]	Доля метки времени в миллисекундах (актуальное время системы).								
Timestamp for the last fault [s]	Доля метки времени в секундах (актуальное время системы).								

Системная переменная	Описание										
State of the red. channel	<p>Текущее состояние канала 2. Состояние канала – это текущее состояние канала 2 в момент (поряд. № X-1) принятия сообщения с поряд. № X.</p> <table> <tr> <th>Состояние</th><th>Описание</th></tr> <tr> <td>0</td><td>Информация по состоянию канала 2 отсутствует.</td></tr> <tr> <td>1</td><td>Канал 2 в порядке.</td></tr> <tr> <td>2</td><td>Последнее сообщение содержало ошибку, текущее в порядке.</td></tr> <tr> <td>3</td><td>Ошибка в канале 2.</td></tr> </table>	Состояние	Описание	0	Информация по состоянию канала 2 отсутствует.	1	Канал 2 в порядке.	2	Последнее сообщение содержало ошибку, текущее в порядке.	3	Ошибка в канале 2.
Состояние	Описание										
0	Информация по состоянию канала 2 отсутствует.										
1	Канал 2 в порядке.										
2	Последнее сообщение содержало ошибку, текущее в порядке.										
3	Ошибка в канале 2.										

Таблица 24: Вкладка Системные переменные в редакторе safeethernet

4.5 Возможные соединения safeethernet

Вы можете создавать соединение **safeethernet** между двумя системами управления HIMax, одиночное или избыточное.

Доступные интерфейсы Ethernet для соединения **safeethernet** всегда отображаются со ссылкой на ресурс (local), для которого был запущен редактор **safeethernet**.

Все доступные интерфейсы Ethernet системы управления отображаются в меню Dropdown параметров **IF CH...**

Элемент	Описание
IF CH1 (local)	Интерфейс Ethernet (канал 1) ресурса
IF Ch2 (local)	Интерфейс Ethernet (канал 2) ресурса
IF Ch1 (target)	Интерфейс Ethernet (канал 1) участника
IF Ch2 (target)	Интерфейс Ethernet (канал 2) участника

Таблица 25: Доступные интерфейсы Ethernet

4.5.1 Одиночное соединение safeethernet (канал 1)

Для создания одиночного соединения Вы должны разместить интерфейсы Ethernet *IF CH1 (local)* и *IF Ch1 (target)* в локальном ресурсе.

4.5.2 Избыточное соединение safeethernet (канал 1 и канал 2)

Возможны избыточные маршруты **safeethernet** между двумя системами управления HIMax.

i

Пропускная способность и задержка на обоих маршрутах должны быть примерно одинаковыми.

Для избыточного соединения можно использовать следующие интерфейсы Ethernet:

- Интерфейсы Ethernet *IF Ch1 (local)* и *IF Ch1 (target)* для канала 1.
- Интерфейсы Ethernet *IF Ch2 (local)* и *IF Ch2 (target)* для канала 2.
- Для избыточного соединения через канал 1 и канал 2 только через один интерфейс Ethernet выберите в редакторе **safeethernet** тот же самый интерфейс Ethernet в *IF Ch1 (local)* канал 1 и *IF Ch2 (local)* канал 2.

i

Обратный путь связи в редакторе **safeethernet** целевого ресурса добавляется автоматически.

4.5.3 Разрешенные комбинации

В следующей таблице приведены возможные комбинации для избыточного соединения **safeethernet**.

Канал 1 IF Ch1 (local)/IF Ch1 (target)	Канал 2 IF Ch2 (local)/IF Ch2 (target)
CPU1/CPU1	CPU2/CPU2
CPU1/CPU1	CPU1/CPU1
COM1/COM1	COM2/COM2
CPU1/COM1	CPU2/COM2
CPU1/COM2	CPU2/COM1
CPU1/CPU1	COM1/COM1

Таблица 26: Комбинации для соединений **safeethernet**

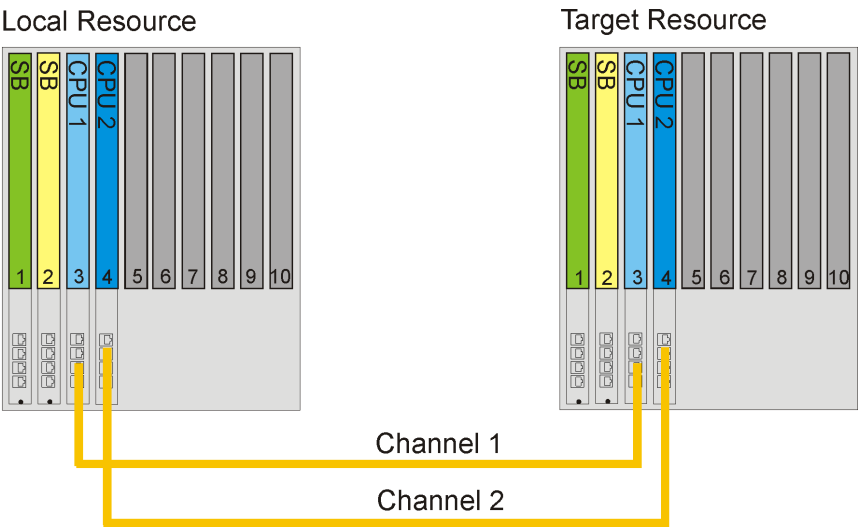


Рисунок 7: Избыточное соединение между двумя системами управления HIMax

Чтобы обеспечить неизменную скорость обработки данных, несмотря на потерянные пакеты данных (например, в результате воздействия ЭМС), рекомендуется создать следующее, сконфигурированное как избыточное, соединение (через провод).

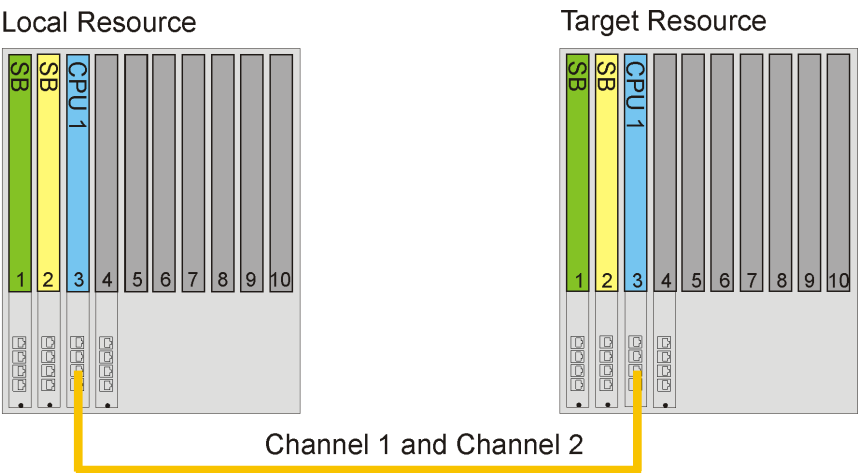


Рисунок 8: Конфигурированное как избыточное соединение двух систем управления HIMax через провод

4.6 Параметры safeethernet

Безопасная связь создается в редакторе **safeethernet**. Для этого Вам необходимо задать описанные в данной главе параметры.

4.6.1 Максимальное время цикла системы управления HIMax

Для определения максимального времени цикла для системы управления HIMax компания HIMA рекомендует следующий порядок действий полностью для всей системы, у которой установлены все процессорные модули.

1. Настроить на высокую отметку время сторожевого устройства для испытания.
2. Вынуть и снова вставить первый процессорный модуль. Подождать, пока процессорный модуль синхронизируется. Выполнить эти действия для **каждого** последующего процессорного модуля системы управления HIMax.

i Процессорный модуль, который Вы добавляете к работающей системе HIMax, автоматически синхронизируется с конфигурацией имеющихся процессорных модулей. При этом достигается максимальное время цикла системы управления. Подробное описание установки и снятия модуля ЦПУ Вы найдете в руководстве по X-CPU HI 801 064 R.

3. Считать и записать время синхронизации процессорных модулей с n на n+1 в истории диагностики.
 4. Повторить данные действия для второго равноправного участника (второй системы управления HIMax).
-

i Для последующего расчета используйте максимальное считанное в истории диагностики время синхронизации обеих систем управления. Для максимального времени цикла, которое может достигнуть система управления, действует следующее:
Время синхронизации + резерв (15 мс).

4.6.2 Время ожидания приема (Receive Timeout)

ReceiveTMO – время контроля, в течение которого от участника коммуникации должен быть получен корректный ответ.

Если в течение *ReceiveTMO* (времени ожидания приема) от участника коммуникации не поступает корректного ответа, то безопасная связь закрывается. Входные переменные данного соединения **safeethernet** ведут себя согласно настроенному параметру *Freeze Data on Lost Connection [ms]*.

Поскольку *ReceiveTMO* (время ожидания приема) является релевантным для безопасности параметром и составной частью Worst Case Reaction Time T_R (времени реакции в худшем случае, см. главу 4.7.1 и далее), то *ReceiveTMO* должно рассчитываться, как показано далее, и заноситься в редактор **safeethernet**.

$ReceiveTMO \geq 4 \cdot Delay + 5 \cdot max. cycle time$ (действительно для профиля «Fast & Noisy»)

Delay: Задержка на линии передачи, например, в виду сетевого коммутатора, спутника и т.д.)

Max. cycle time Максимальное время цикла обеих систем управления

i Необходимая отказоустойчивость связи может быть достигнута путем увеличения *ReceiveTMO* (времени ожидания приема), если это допустимо по времени для процесса использования, см. главу 4.7.

4.6.3 Время ответа (ResponseTime)

ResponseTime – это время в миллисекундах (мс), которое проходит, пока отправитель сообщения не получит подтверждение приема от получателя.

Для параметрирования с применением профиля **safeethernet** должно быть задано *ResponseTime* (время ответа), ожидаемое ввиду физических данных маршрута.

Заданное *ResponseTime* (время ответа) воздействует на конфигурацию всех параметров соединения **safeethernet**, это время ответа рассчитывается следующим образом:

$$\text{ResponseTime} \leq \text{ReceiveTMO}/n$$

$$n = 2, 3, 4, 5, 6, 7, 8 \dots$$

Соотношение времени ожидания приема (*ReceiveTMO*) и времени ответа (*ResponseTime*) воздействует на способность к отказоустойчивости, например, при потере пакета (повтор потерянных пакетов данных) или задержках на тракте передачи.

В сети, в которой может произойти потеря пакета, должно быть выполнено следующее условие:

$$\text{min. Response Time} \leq \text{ReceiveTMO}/2 \geq 2 * \text{Delay} + 2,5 * \text{max. Cycle Time}$$

(действительно для профиля «Fast & Noisy»)

Если это условие выполнено, то можно предотвратить потерю минимум одного пакета данных, и при этом не будет прервана связь **safeethernet**.

i

Если это условие не выполнено, то доступность соединения **safeethernet** гарантирована только при бесконфликтной и бесперебойной работе сети. Однако, это не влияет на безопасность процессорного модуля!

4.6.4 Настройка Sync/Async

Sync Настройка *Sync* означает, что обмен данными осуществляется синхронно с циклом ЦПУ, т.е. выполняется ожидание ответа от участника коммуникации.

- Во входной фазе цикла ЦПУ передачу и прием осуществляет активный участник коммуникации.
- В выходной фазе цикла ЦПУ активный участник коммуникации еще раз осуществляет передачу и прием.

Такой принцип действий желателен, например, для связи системы управления с ее удаленными устройствами ввода/вывода, и является настройкой по умолчанию.

Для связи между двумя системами управления синхронный обмен данными подходит меньше, поскольку пассивная система управления, прежде чем ответить, сначала выполняет вашу прикладную программу. Поскольку активная система управления ожидает ответа пассивной системы управления, цикл ЦПУ активной системы управления может сильно увеличиться.

Это означает, что цикл ЦПУ пассивной системы управления должен быть достаточно маленьким, чтобы пассивная система управления могла принимать и отвечать в каждой входной/выходной фазе активной системы управления.

Async Является настройкой по умолчанию.

При настройке *Async* экземпляр протокола **safeethernet** во входной фазе ЦПУ и отправляет согласно правилам отправителя в выходной фазе ЦПУ.

4.6.5 Время ожидания повторной передачи (ResendTMO)

Время ожидания повторной передачи (ResendTMO) не может быть введено вручную, а рассчитывается из профиля и времени ответа (Response Time).

Время контроля в миллисекундах (мс) на ПЭС 1, в течение которого ПЭС 2 должна подтвердить прием пакета данных, в противном случае происходит повторное отправление пакета данных.

Правило: $ResendTMO \leq Receive-Timeout$

При различной конфигурации времени ожидания повторной передачи (*ResendTMO*) у участников коммуникации активный участник протокола (меньший SRS) определяет фактическое значение времени ожидания повторной передачи соединения протокола.

4.6.6 Время ожидания подтверждения (Acknowledge Timeout)

Время ожидания подтверждения (AckTMO) не может быть введено вручную, а рассчитывается из профиля и времени ответа (Response Time).

AckTMO – это максимальное время, после которого принятый пакет данных должен быть подтвержден ЦПУ.

Для сети с высокой скоростью передачи данных время ожидания подтверждения (*AckTMO*) должно равняться нулю, т.е. прием пакета данных подтверждается немедленно. Для сети с небольшой скоростью передачи данных (напр., линия телефон-модем) время ожидания подтверждения (*AckTMO*) должно быть больше нуля. В этом случае осуществляется попытка передать сообщение о подтверждении вместе с данными, чтобы снизить нагрузку на сеть, избегая блоков адресации и защиты.

Правила:

- $AckTMO \leq Receive-Timeout$
- $AckTMO \leq Resend-Timeout$, если $Prod-Rate > Resend-Timeout$.

4.6.7 Интервал передачи пакетов (Prod-Rate)

Интервал передачи пакетов (ProdRate) не может быть введен вручную, а рассчитывается из профиля и времени ответа (Response-Time).

Минимальный интервал времени в миллисекундах (мс) между двумя пакетами данных.

Целью параметра *Prod-Rate* (интервал передачи пакетов) является ограничение количества пакетов данных до такой степени, чтобы не перегружался (медленный) канал связи. Благодаря этому достигается равномерная нагрузка среды передачи данных, и не осуществляется прием устаревших данных со стороны получателя.

Правила:

- $ProdRate \leq Receive-Timeout$
- $ProdRate \leq Resend-Timeout$, если $Acknowledge-Timeout > Resend-Timeout$

i

Интервал передачи пакетов равный нулю означает, что в каждом цикле прикладной программы осуществляется передача пакетов данных.

4.6.8 Память

Память не может быть введена вручную, а рассчитывается из профиля и времени ответа (Response-Time).

Память (Queue-Tiefe) – это количество пакетов данных, которые могут быть отправлены без ожидания подтверждения о приеме.

Значение зависит от пропускной способности сети и возможных задержек из-за времени прохождения в сети.

Все соединения **safeethernet** используют доступную память для сообщений в ЦПУ.

4.7 Максимальное время реакции

На следующих примерах формулы для расчета максимальной времени реакции в случае соединения с системами управления HIMatrix действительны только в том случае, если настроено безопасное время = 2 * время сторожевого устройства. Для систем управления HIMax эти формулы действительны всегда.

i

Максимальное время реакции зависит от процесса. Согласовать с осуществляющим приемку отделом контроля.

Понятия:

ReceiveTMO:	Время контроля в ПЭС 1, в течение которого должен приниматься действительный ответ от ПЭС 2. По истечении времени безопасная связь завершается.
ProdRate:	Минимальный интервал между двумя передачами данных.
Watchdog Time:	Максимально разрешенная продолжительность цикла RUN в системе управления. Продолжительность цикла RUN зависит от степени сложности прикладной программы и количества соединений safeethernet . Внести время сторожевого устройства (WDZ) в свойства ресурса.
Worst Case Reaction Time	Максимальное время реакции для передачи изменения сигнала физического входа (In) ПЭС 1 до изменения физического выхода (Out) ПЭС 2.
Delay:	Задержка на линии передачи, например, при модемном или спутниковом соединении. При прямом соединении сначала может быть принята задержка равная 2 мс. Фактическая задержка на линии передачи может быть измерена ответственным сетевым администратором.

4.7.1 Расчет максимального времени реакции двух ПЭС HIMax

Рассчитать максимальное время реакции T_R («Worst Case») от смены датчика ПЭС 1 (In) до реакции выхода (Out) ПЭС 2, как показано далее:

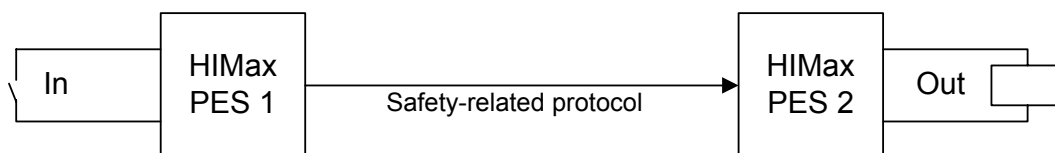


Рисунок 9: Соединение **safeethernet** двух систем управления HIMax

Время реакции при соединении двух систем управления HIMax

$$T_R = t_1 + t_2 + t_3 + t_4$$

T_R	Worst Case Reaction Time (Время реакции в худшем случае)
t_1	Безопасное время ПЭС 1
t_2	0 мс, если <i>ProdRate</i> = 0 (профиль <i>Fast</i>), в противном случае <i>ReceiveTMO</i> + время сторожевого устройства ПЭС 1
t_3	<i>ReceiveTMO</i>
t_4	Безопасное время ПЭС 2

4.7.2 Расчет макс. времени реакции в соединении с ПЭС HIMatrix

Рассчитать максимальное время реакции T_R («Worst Case») от смены датчика (In) ПЭС 1 HIMax до реакции выхода (Out) ПЭС 2 HIMatrix, как показано далее:

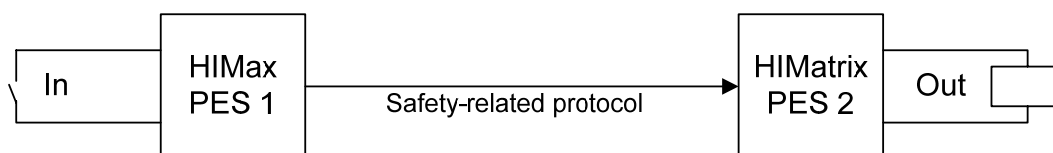


Рисунок 10: Соединение safeethernet системы HIMax с системой управления HIMatrix

Время реакции при соединении системы HIMax с системой управления HIMatrix

$$T_R = t_1 + t_2 + t_3 + t_4$$

T_R	Время реакции в худшем случае (Worst Case Reaction Time)
t_1	Безопасное время ПЭС 1 HIMax
t_2	0 мс, если <i>ProdRate</i> = 0 (профиль <i>Fast</i>), в другом случае <i>время ожидания приема (ReceiveTMO)</i> + время сторожевого устройства ПЭС 1
t_3	<i>ReceiveTMO</i>
t_4	Безопасное время ПЭС 2 HIMatrix

4.7.3 Расчет макс. времени реакции с двумя системами управления HIMatrix или удаленными устройствами ввода/вывода

Рассчитать максимальное время реакции T_R («Worst Case») от смены датчика (In) в первой ПЭС HIMatrix или в удаленном устройстве ввода/вывода (напр., F3 DIO 20/8 01) до реакции вывода во второй ПЭС HIMatrix или в удаленном устройстве ввода/вывода, как показано далее:

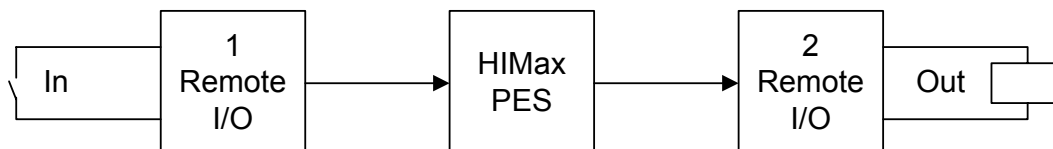


Рисунок 11: Соединение safeethernet с удаленными устройствами ввода/вывода

Время реакции с удаленным устройством ввода/вывода

T_R	$t_1 + t_2 + t_3 + t_4 +$ (путь ввода)
	$+ t_5 + t_6 + t_7$ (путь вывода)
T_R	Worst Case Reaction Time (Время реакции в худшем случае)
t_1	Безопасное время 1 удаленного устройства ввода/вывода
t_2	0 мс, если <i>ProdRate</i> = 0 (профиль <i>Fast</i>), в другом случае <i>ReceiveTMO1</i> + время сторожевого устройства 1-го удаленного устройства ввода/вывода
t_3	<i>ReceiveTMO1</i>
t_4	2 * время сторожевого устройства ПЭС HIMax
t_5	<i>ReceiveTMO2</i>
t_6	0 мс, если <i>ProdRate</i> = 0 (профиль <i>Fast</i>), в другом случае <i>ReceiveTMO2</i> + время сторожевого устройства ПЭС
t_7	Безопасное время 2-го удаленного устройства ввода/вывода

i

Оба удаленных устройства ввода/вывода 1 и 2 могут быть идентичными. Значения времени действует также в том случае, если вместо удаленного устройства ввода/вывода используется ПЭС HIMatrix.

4.7.4 Расчет макс. времени реакции с двумя системами HIMax и одной ПЭС HIMatrix

Рассчитать максимальное время реакции T_R («Worst Case») от смены датчика (In) в первой ПЭС HIMax до реакции вывода (Out) во второй ПЭС HIMax, как показано далее:

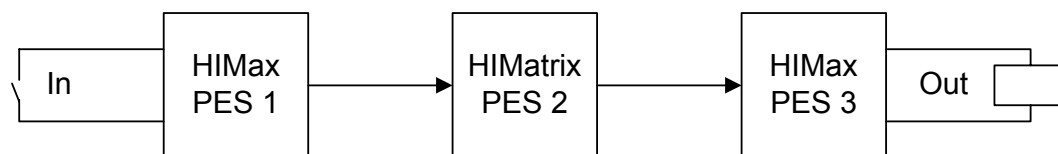


Рисунок 12: Соединение safeethernet двух систем HIMax с одной системой управления HIMatrix

Время реакции с двумя системами управления HIMax и одной системой управления HIMatrix

T_R	$t_1 + t_2 + t_3 + t_4 +$ (путь ввода)
	$+ t_5 + t_6 + t_7$ (путь вывода)
T_R	Worst Case Reaction Time (Время реакции в худшем случае)
t_1	Безопасное время ПЭС 1 HIMax
t_2	0 мс, если <i>ProdRate</i> = 0 (профиль <i>Fast</i>), в другом случае <i>ReceiveTMO1</i> + время сторожевого устройства ПЭС 1 HIMax
t_3	<i>ReceiveTMO1</i>
t_4	Безопасное время ПЭС 2 HIMatrix
t_5	<i>ReceiveTMO2</i>
t_6	0 мс, если <i>ProdRate</i> = 0 (профиль <i>Fast</i>), в другом случае « <i>ReceiveTMO2</i> » + время сторожевого устройства ПЭС 2 HIMatrix
t_7	Безопасное время ПЭС 3 HIMax

i

Обе ПЭС 1 и 3 HIMax могут быть идентичными.

4.8 Профили Safeethernet

Профили **safeethernet** представляют собой комбинацию подходящих параметров, которые настраиваются автоматически, если Вы выбираете один из профилей **safeethernet**. Для параметрирования Вы должны будете только конфигурировать по отдельности время ожидания приема и ожидаемое время ответа.

Цель профиля **safeethernet** состоит в том, чтобы оптимизировать скорость обработки данных в сети с учетом физических условий.

і

Условием для действительной оптимизации является то, что *временной интервал связи* должен быть достаточно большим для того, чтобы в **одном** цикле ЦПУ обработать все соединения **safeethernet**. *Временной интервал связи* настраивается в свойствах ресурса.

Предоставляется шесть профилей **safeethernet**, из которых Вы можете выбрать подходящий для линии передачи профиль **safeethernet**.

Сетевой профиль	Глава
Fast & Cleanroom	0
Fast & Noisy	<input type="checkbox"/>
Medium & Cleanroom	0
Medium & Noisy	<input type="checkbox"/>
Slow & Cleanroom	0
Slow & Noisy	0

Таблица 27: Сетевые профили **safeethernet**

4.8.1 Профиль I (Fast & Cleanroom)

і

Профиль *Fast & Noisy* в обычном случае должен предпочитаться профилю *Fast & Cleanroom*.

Использование

Профиль *Fast & Cleanroom* подходит для применений в идеальных условиях, например, в лаборатории!

- Для обработки данных с максимальной скоростью
- Для применений, которые требуют быстрой передачи данных
- Для применений, для которых требуется минимальное время реакции в худшем случае

Требования к сети

- Fast: технология 100 Мбит (100 Base TX), технология 1 Гбит
- Clean: Бесперебойная работа сети.
Следует избегать потерь данных в результате перегрузки сети, влияния извне или манипуляций с сетью.
- Требуется сетевой коммутатор LAN!

Характеристики пути связи

- Минимальные задержки
- Ожидаемое время ответа (ResponseTime) ≤ время ожидания приема (ReceiveTMO)
(в противном случае ERROR при программировании)

4.8.2 Профиль II (Fast & Noisy)

Использование

Профиль *Fast & Noisy* является стандартным профилем SILworX для связи через **safeethernet**.

- Для обработки данных с большой скоростью
- Для применений, которые требуют быстрой передачи данных
- Для применений, которые требуют минимального времени реакции в худшем случае

Требования к сети

- Fast: технология 100 Мбит (100 Base TX), технология 1 Гбит
- Noisy: Сеть не является бесперебойной.
Незначительная вероятность потери пакетов данных
время для ≥ 1 повтора
- Требуется сетевой коммутатор LAN!

Характеристики пути связи

- Минимальные задержки
- Ожидаемое время ответа (ResponseTime) \leq время ожидания приема/2 (ReceiveTMO) (в противном случае ERROR при параметрировании)

4.8.3 Профиль III (Medium & Cleanroom)

i

Профиль *Medium & Noisy* в обычном случае должен предпочитаться профилю *Medium & Cleanroom*.

Использование

Профиль *Medium & Cleanroom* используется для применений в бесперебойной сети, которые требуют среднескоростной передачи данных.

- Для среднескоростной обработки данных.
- Подходит для виртуальных частных сетей (VPN), в которых обмен данными происходит медленно через промежуточные устройства безопасности (сетевое устройство защиты, кодирование), но без сбоев.
- Подходит для применений, в которых максимальное время реакции не является критическим фактором.

Требования к сети

- Medium: технология 10 Мбит (10 Base T), 100 Мбит (100 Base TX), 1 Гбит
- Требуется сетевой коммутатор LAN!
- Clean: Бесперебойная работа сети.
Следует избегать потерь данных в результате перегрузки сети, влияния извне или манипуляций с сетью.
Время для ≥ 0 повторов.

Характеристики пути связи

- Небольшие задержки
- Ожидаемое время ответа (ResponseTime) \leq время ожидания приема (ReceiveTMO) (в противном случае ERROR при параметрировании)

4.8.4 Профиль IV (Medium & Noisy)

Использование

Профиль *Medium & Noisy* используется для применений, которые требуют среднескоростной передачи данных.

- Для среднескоростной обработки данных
- Для применений, которые требуют среднескоростной передачи данных
- Подходит для применений, в которых максимальное время реакции не является критическим фактором

Требования к сети

- Medium: технология 10 Мбит (10 Base T), 100 Мбит (100 Base TX), 1 Гбит
- Требуется сетевой коммутатор LAN!
- Noisy: Сеть не является бесперебойной.
Незначительная вероятность потери пакетов данных,
время для ≥ 1 повтора

Характеристики пути связи

- Небольшие задержки
- Ожидаемое время ответа (ResponseTime) \leq время ожидания приема/2 (ReceiveTMO) (в противном случае ERROR при параметрировании)

4.8.5 Профиль V (Slow & Cleanroom)

i

Профиль *Slow & Noisy* в обычном случае должен предпочитаться профилю *Slow & Cleanroom*.

Использование

Профиль *Slow & Cleanroom* предназначен для применений в бесперебойной сети, которые требуют передачи данных с медленной скоростью.

- Для обработки данных с медленной скоростью.
- Для применений, которые требуют передачи данных с медленной скоростью на (возможно, сильно удаленные) системы управления, и в случаях, если условия линии связи непредсказуемы.

Требования к сети

- Slow: Передача данных через ISDN, постоянное соединение или радиорелейная связь.
- Clean: Бесперебойная работа сети.
Следует избегать потерь данных в результате перегрузки сети, влияния извне или манипуляций с сетью.
Время для ≥ 0 повторов.

Характеристики пути связи

- Небольшие задержки
- Ожидаемое время ответа (ResponseTime) = время ожидания приема (ReceiveTMO) (в противном случае ERROR при программировании)

4.8.6 Профиль VI (Slow & Noisy)

Использование

Профиль *Slow & Noisy* предназначен для применений, которые требуют передачи данных с медленной скоростью на (возможно, сильно удаленные) системы управления.

- Для обработки данных с медленной скоростью
- Для применений, в основном, для передачи данных через плохие телефонные линии или поврежденные радиорелейные линии.

Требования к сети

- *Slow*: Передача данных через телефонную, спутниковую, радиосвязь и т.д.
- *Noisy*: Сеть не является бесперебойной.
Незначительная вероятность потери пакетов данных
время для ≥ 1 повтора

Характеристики пути связи

- От небольших до длительных задержек
- Ожидаемое время ответа (*ResponseTime*) \leq время ожидания приема/2 (*ReceiveTMO*) (в противном случае ERROR при параметрировании)

4.9 Межпроектная связь

Межпроектная связь требуется для того, чтобы ресурсы могли обмениваться переменными процессов из различных проектов.

Связь между проектами осуществляется через **safeethernet** и конфигурируется в редакторе **safeethernet**.

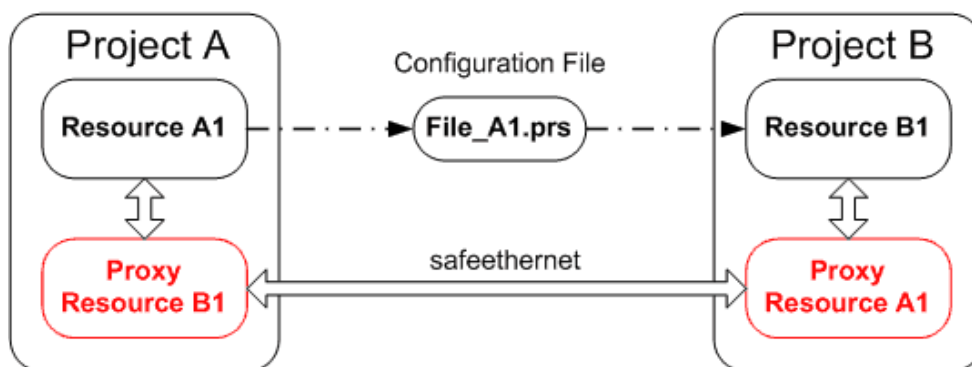


Рисунок 13: Соединение **safeethernet** между ресурсом A1 в проекте A и ресурсом B1 в проекте B

Локальным проектом называется проект, в котором Вы выполняете конфигурацию соединения **safeethernet** и создаете файл конфигурации.

Целевым проектом называется проект, в который Вы импортируете файл конфигурации.

При обмене данными локальный и целевой проект являются равноправными участниками коммуникации.

Соответствующий ресурс прокси (Proxy resource) служит в качестве заполнителя для соответствующего ресурса из внешнего проекта и используется для импорта и экспорта соединений **safeethernet**.

Ресурс прокси B1 в проекте является заполнителем *ресурса B1* из *проекта B*.

Ресурс прокси A1 в проекте B является заполнителем *ресурса A1* из *проекта A*.

В локальном проекте (здесь *проект A*) Вы должны создать и конфигурировать ресурс прокси (здесь *ресурс прокси B1*) вручную. После конфигурации импортировать файл конфигурации (здесь *File_A1.prs*) в целевой проект (здесь *из ресурса B1*).

Файл конфигурации *File_A1.prs* содержит полное описание *ресурса A1* для соединения **safeethernet** с *ресурсом B1*. После импорта файла конфигурации *File_A1.prs* в *Resource B1 Proxy Resource A1* автоматически создается в *Project B*.

4.9.1 Варианты межпроектной связи

Далее описываются варианты связи проектов А и В через **safeethernet**.

При этом в первом варианте проект А является локальным проектом, а во втором варианте локальным проектом является проект В. Принципиально пользователь может выбирать сам, в каком проекте он будет создавать конфигурацию.

По сложности оба варианта конфигурации являются приблизительно одинаковыми и ведут к одинаковому результату.

Локальный проект А

В локальном проекте А Вы конфигурируете связь с целевым проектом В и создаете файлы конфигурации. Преимуществом является то, что Вы должны создать вручную только ресурс прокси В1 в локальном проекте.

Локальный проект

Целевой проект

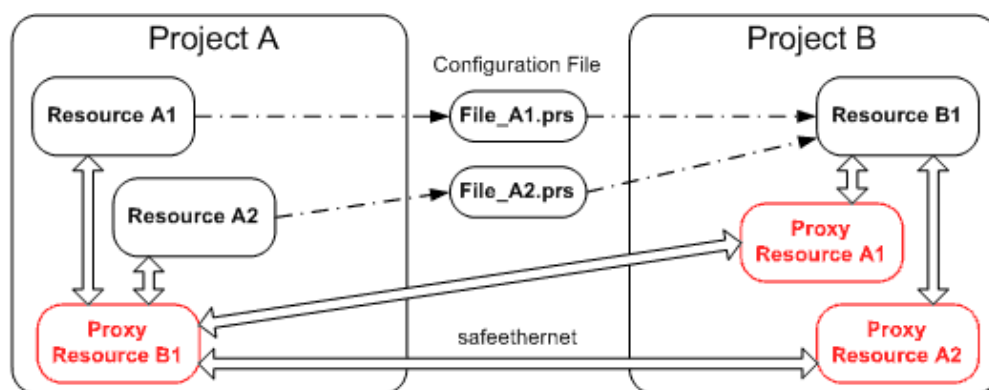


Рисунок 14: Вариант с проектом А в качестве локального проекта

Локальный проект В

В локальном проекте В Вы конфигурируете связь с целевым проектом А и создаете файл конфигурации. Недостатком является то, что Вы должны вручную создать два ресурса прокси (А1 и А2) в локальном проекте В.

Целевой проект

Локальный проект

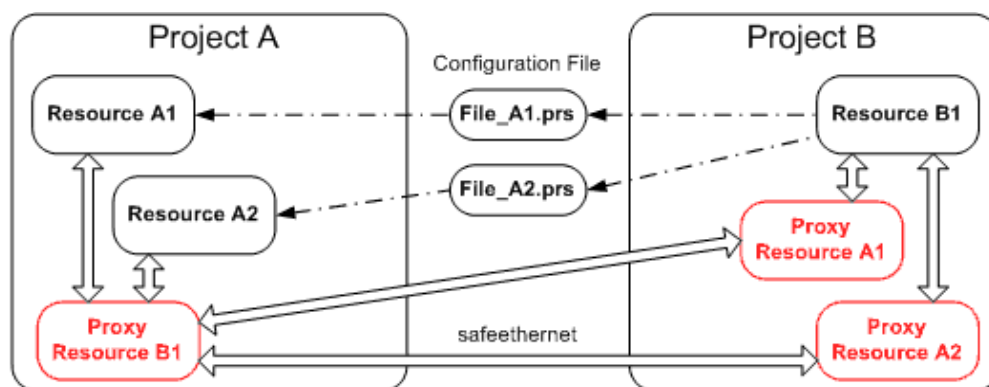


Рисунок 15: Вариант с проектом В в качестве локального проекта

4.10 Межпроектная связь между SILworX и ELOP II Factory

На этом примере конфигурируется соединение **safeethernet** между HIMax и HIMatrix.

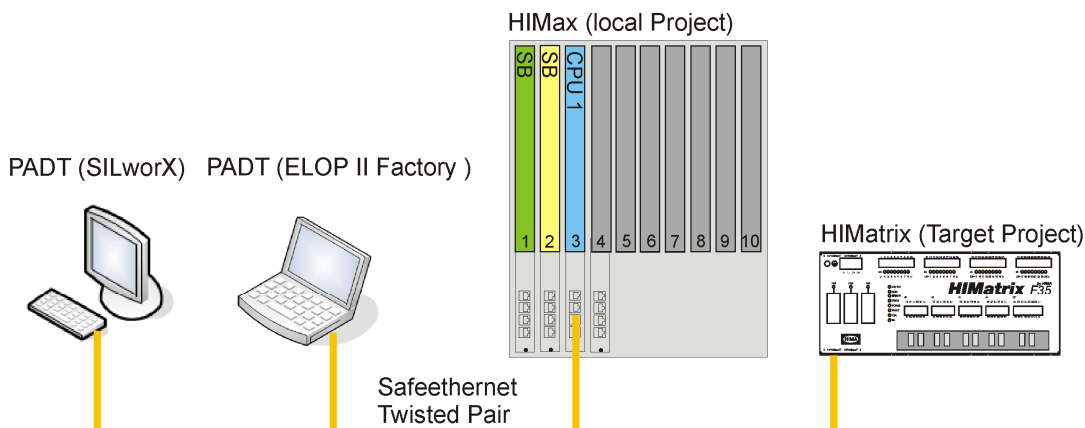


Рисунок 16: Установка связи между SILworX и ELOP II Factory

Открыть ресурс целевого проекта (HIMatrix), который должен служить ресурсом прокси в локальном проекте (HIMax).

Определить следующие параметры для данного целевого ресурса:

- System ID
- Safety time [ms]
- Watchdog time [ms]
- IP address

i

Данные свойства ресурса являются релевантными для безопасности и подлежат ограничениям. Подробные указания Вы найдете в руководстве по функциональной безопасности HI 801 061 R.

4.10.1 Конфигурация HIMax в проекте SILworX

Создание ресурса прокси (Proxy Resource)

Ресурс прокси служит в качестве заполнителя для ресурса из внешнего проекта и используется для импорта и экспорта соединений **safeethernet**.

Чтобы создать в локальном проекте ресурс прокси, необходимо:

1. Открыть локальный проект, в котором должен быть создан ресурс прокси.
2. В структуре открыть **Configuration**.
3. Щелкнуть правой кнопкой мыши на **Configuration** и выбрать **New, Proxy Resource ELOP II Factory**.
 - ☒ Добавлен новый ресурс прокси.

Чтобы конфигурировать ресурс прокси в локальном проекте, необходимо:

1. В контекстном меню ресурса прокси выбрать **Properties**.
2. В поле **Name** ввести однозначное имя.
Для ресурса прокси в локальном проекте используйте имя ресурса в целевом проекте.
3. Ввести определенные прежде в целевом проекте **System ID**, **Safety Time [ms]** и **Watchdog Time [ms]** для данного ресурса прокси.
4. Щелкнуть на **OK**. Другие параметры могут содержать значения по умолчанию.

Чтобы открыть структуру ресурса прокси:

1. Щелкнуть правой кнопкой мыши на **Hardware** и выбрать **Edit, HIMatrix Proxy**.
2. Подтвердить с помощью **OK**, чтобы открыть редактор аппаратного обеспечения ресурса прокси.



Рисунок 17: Ресурс прокси HIMatrix

3. Щелкнуть дважды на **COM Module** и ввести полученный **IP Address** ресурса прокси.
4. Щелкнуть на **Save** и затем на **Close**.
5. Данные действия повторить для каждого последующего ресурса прокси в локальном проекте.

Соединение локального ресурса с ресурсом прокси

Создайте в редакторе **safeethernet** соединение **safeethernet** между локальным ресурсом и ресурсом прокси.

Чтобы открыть редактор safeethernet локального ресурса, необходимо:

1. Открыть в структуре **Configuration, Resource**.
2. Щелкнув правой кнопкой мыши на **safeethernet**, выберите в контекстном меню **Edit**. В выборе объекта находится созданный ресурс прокси.

Чтобы создать соединение safeethernet с ресурсом прокси, необходимо:

1. В выборе объекта щелкнуть на **Proxy Resource** и с помощью Drag&Drop перетащить на свободное место в рабочей области редактора **safeethernet**.
2. Выбрать интерфейсы **Ethernet** локального ресурса и ресурса прокси.
Следующие параметры определяют скорость обработки данных, а также отказоустойчивость и устойчивость к конфликтам для соединения **safeethernet**:
5. Выбрать **Network Profile** (напр., **Fast&Noisy**) соединения **safeethernet**.
6. Рассчитать и внести время ожидания приема (**Receive Timeout**) и время ответа (**Response Time**).

Пример для значений параметров соединения **safeethernet** для ресурса прокси:

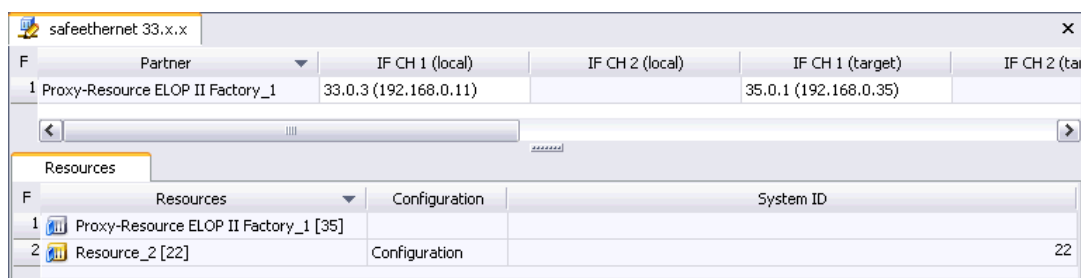


Рисунок 18: Параметры соединения **safeethernet** для ресурса прокси

Соединение переменных процесса

Соедините переменные процесса в детальном виде соединения **safeethernet**.

Чтобы открыть детальный вид соединения **safeethernet**, необходимо:

Условие: Редактор **safeethernet** локального ресурса должен быть открыт.

1. Правой кнопкой мыши щелкнуть на **Proxy Resource** и открыть ресурс прокси.
2. Выбрать в контекстном меню **Detail View**, чтобы открыть детальный вид соединения **safeethernet**.
3. Выбрать вкладку **Resource<->Proxy Resource**.

Чтобы добавить переменные передачи **safeethernet**, необходимо:

Переменные передачи передаются с локального ресурса на ресурс прокси.

1. Выбрать область **Resource->Proxy Resource**.
2. В выборе объекта выбрать **Global Variable** и с помощью Drag&Drop перетащить в столбец **Resource (target)->Resource (local)**.
3. Повторить данное действие для других переменных передачи **safeethernet**.

Чтобы добавить переменные приема **safeethernet**, необходимо:

Переменные приема принимаются с локального ресурса.

1. Выбрать область **Resource<-Proxy Resource**.
2. В выборе объекта выбрать **Global Variable** и с помощью Drag&Drop перетащить в столбец **Resource (target)<-Resource (local)**.
3. Повторить данное действие для других переменных приема **safeethernet**.

Экспорт файла конфигурации из SILworX

Конфигурированное в SILworX соединение **safeethernet** необходимо экспортировать как файл конфигурации с расширением *.prs. Данный файл конфигурации можно затем импортировать в ELOP II Factory, чтобы сохранить соединение **safeethernet** для системы управления HIMatrix.

Чтобы экспортировать соединение **safeethernet**, необходимо:

1. В редакторе **safeethernet** щелкнуть на **Proxy Resource** и открыть контекстное меню.
2. В контекстном меню выбрать **Export Connection with Proxy Resource**: открывается стандартное диалоговое окно для сохранения файла.
3. В диалоговом окне «File name» ввести имя файла конфигурации и сохранить файл с расширением *.prs.
4. Закрыть локальный проект.

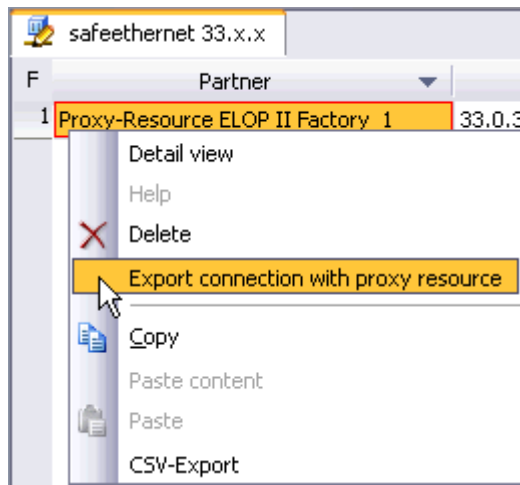


Рисунок 19: Экспорт соединения Safeethernet

Подтверждение соединения **safeethernet**:

1. Открыть в структуре **Configuration, Resource, safeethernet**.
2. Щелкнуть на кнопку **Verification** в строке меню и подтвердить с помощью **OK**.
3. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.



Конфигурацию соединения **safeethernet** и прикладную программу ресурса HIMax необходимо компилировать заново и загрузить в систему управления, чтобы они были активны для связи HIMax.

4.10.2 Конфигурация HIMatrix в ELOP II Factory

Чтобы импортировать файл конфигурации в целевой проект (HIMatrix), необходимо:

1. Запустить ELOP II Factory.
2. Открыть целевой проект HIMatrix, в который должен быть импортирован файл конфигурации.
3. Выбрать в структуре целевой ресурс и открыть в контекстном меню.
4. Выбрать в контекстном меню **Import Connections**:
откроется диалоговое окно для загрузки файла с расширением *.prs.
5. Выбрать в диалоговом окне файл конфигурации, который Вы создали в локальном проекте HIMax, и щелкнуть на **OK**.
☒ После импорта файла конфигурации автоматически создается локальный ресурс HIMax в качестве ресурса прокси в целевом проекте HIMatrix.

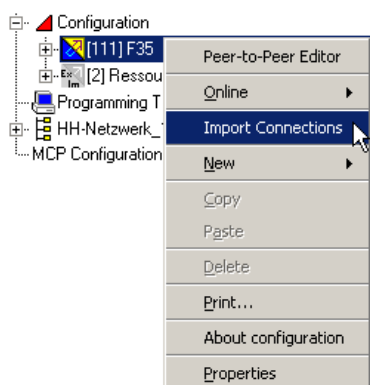


Рисунок 20: Импорт соединений в ELOP II Factory

Присвоение сигналов процесса ELOP II Factory

Соедините сигналы процесса в целевом ресурсе (HIMatrix).

Открыть редактор сигналов через строку меню **Signals, Editor**.

Чтобы открыть редактор P2P для целевого ресурса в ELOP II Factory, необходимо:

1. Открыть в структуре **Configuration, Resource, P2P Editor**.
2. В **P2P Editor** ввести **HH Network** для данного соединения.
3. В **P2P Editor** щелкнуть на **Connect Process Signals**.

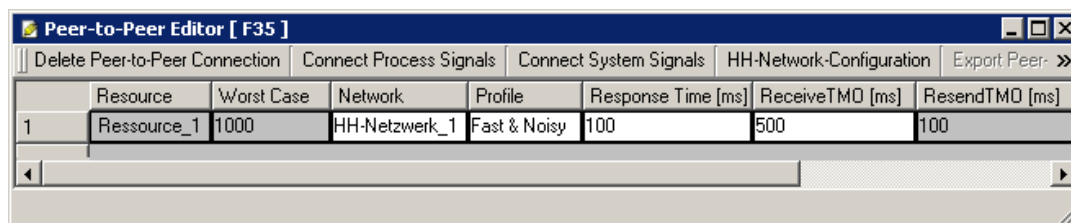


Рисунок 21: Редактор P2P в ELOP II Factory

i

Следите за тем, чтобы оба участника коммуникации использовали одинаковый профиль и одинаковые настройки (автоматически сохраняется при импорте файла конфигурации).

Сигналы передачи в одноранговой сети присваиваются следующим образом:

Сигналы передачи в одноранговой сети передаются от ресурса HIMatrix к ресурсу HIMax.

1. Выбрать вкладку **HIMatrix Resource-> HIMax Proxy Resource**.
Вкладка содержит импортированные сигналы передачи одноранговой сети.
2. В редакторе сигналов щелкнуть на **Process Signal** и с помощью Drag&Drop перетащить на сигнал передачи в диалоговом окне для сигналов процесса одноранговой сети, который Вы хотите объединить.
3. Повторить данное действие для других сигналов передачи одноранговой сети.

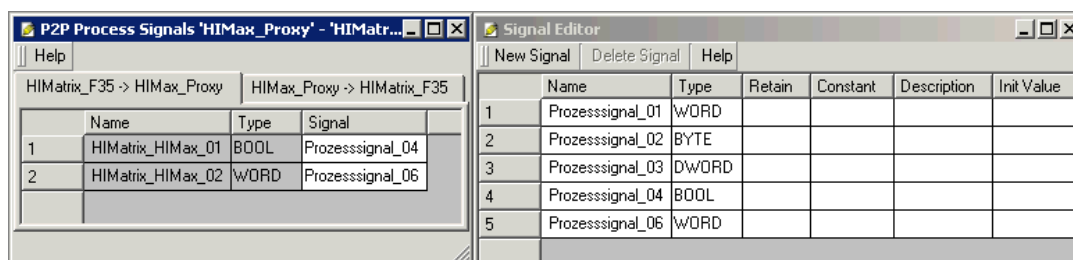


Рисунок 22: Присвоение сигналов передачи в ELOP II Factory

Сигналы приема одноранговой сети присваиваются следующим образом:

Сигналы приема одноранговой сети принимаются ресурсом HIMatrix.

1. Выбрать вкладку **Resource<-Proxy Resource**.
Вкладка содержит импортированные сигналы приема одноранговой сети.
2. В редакторе сигналов щелкнуть на **Process Signal** и с помощью Drag&Drop перетащить сигнал приема в диалоговое окно для сигналов процесса одноранговой сети, который Вы хотите объединить.
3. Повторить данное действие для других сигналов приема одноранговой сети.

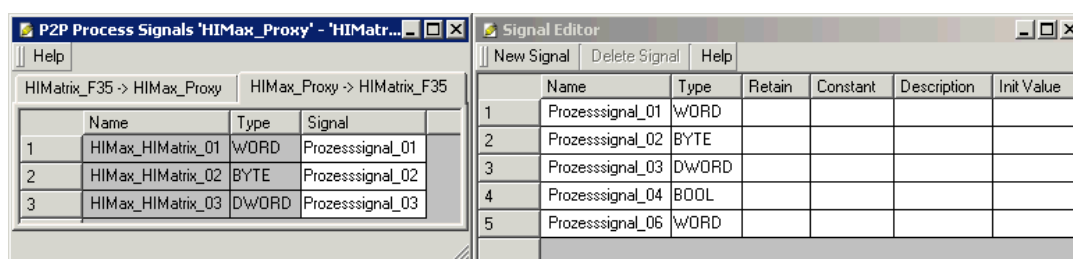


Рисунок 23: Присвоение сигналов приема в ELOP II Factory



Более подробная информация по объединению сигналов процесса в ELOP II Factory содержится в онлайн-справке ELOP II Factory.



Конфигурацию однорангового соединения и прикладную программу ресурса HIMatrix необходимо компилировать заново и загрузить в систему управления. Только после этого одноранговое соединение становится активным для HIMatrix.

4.11 Панель управления (safeethernet)

В панели управления пользователь может проверять и регулировать настройки соединения **safeethernet**. Для этого отображается актуальная информация по состоянию (напр., время цикла, состояние шины и т.д.) соединения **safeethernet**.

Чтобы открыть панель управления для контроля соединения **safeethernet, необходимо:**

1. В структуре выбрать **Resource**.
2. Выбрать в контекстном меню ресурса **Online**.
3. В **System Log-in** ввести данные доступа, чтобы открыть панель управления ресурса.
4. В структуре панели управления выбрать **safeethernet**.

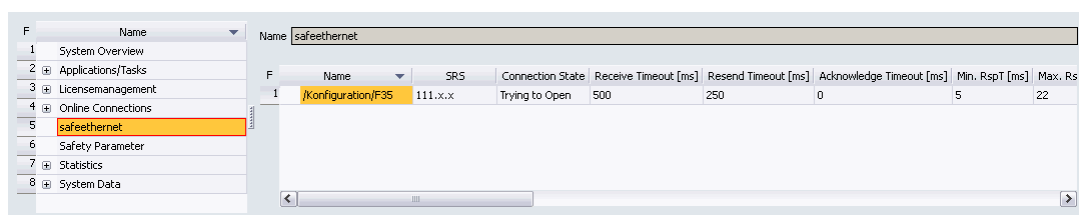


Рисунок 24: Панель управления для контроля соединения

Сброс статистических значений:

С помощью функции контекстного меню Вы можете сбросить на нуль статистические данные (время цикла мин., макс. и т.д.).

Чтобы сбросить статистические данные соединения **safeethernet, необходимо:**

1. В структуре выбрать соединение **safeethernet**.
2. В контекстном меню соединения **safeethernet** выбрать **safeethernet Reset Statistic**.

4.11.1 Поле индикации (соединение **safeethernet**)

В поле индикации отображаются следующие значения выбранного соединения **safeethernet**:

Элемент	Описание
Name	Имя ресурса участника коммуникации
SRS	System.Rack.Slot
Connection State	Состояние соединения safeethernet (см. также главу 4.4)
Receive timeout [ms]	(см. также главу 4.6.2)
Resend timeout [ms]	(см. также главу 4.6.5)
Acknowledge timeout [ms]	(см. также главу 4.6.6)
Min. RspT [ms]	Фактическое время ответа как минимальное, максимальное, последнее и среднее значение. См. также главу 4.6.3.
Max. RspT [ms]	
Last RspT [ms]	
Av. RspT [ms]	
Bad Messages	Количество отклоненных сообщений с момента сброса статистики.
Repeats	Количество повторов с момента сброса статистики.
Number of Successful Connections	Количество успешных соединений с момента сброса статистики.
Early Queue Usage	Количество сообщений, которые были созданы в первой очереди с момента сброса статистики. См. также главу 4.6.8.
Frame No.	Актуальный счетчик передачи
Ack.Frame No.	Актуальный счетчик приема
Monotony	Актуальный счетчик передачи полезной нагрузки
Layout Version	Сигнатура точки передачи текущей коммуникации
New Layout Version	Сигнатура точки передачи новой коммуникации
Connection control	Состояние системы управления соединением. См. также главу 4.4.
Transport Control Ch1	Деблокировка маршрута, канал 1. См. также главу 4.4.
Transport Control Ch2	Деблокировка маршрута, канал 2. См. также главу 4.4.
Quality of Channel 1	Состояние маршрута, канал 1. См. также главу 4.4.
Quality of Channel 2	Состояние маршрута, канал 2. См. также главу 4.4.
Late Received Redundant Messages	При избыточных маршрутах. Количество принятых с опозданием сообщений после сброса статистики.
Lost Redundant Messages	При избыточных маршрутах. Количество принятых только на одном из двух маршрутов сообщений с момента сброса статистики.
Protocol Version	0 и 1 = старая версия протокола для HIMatrix < V7 2 = новая версия протокола для HIMax

Таблица 28: Поле индикации соединения **safeethernet**

5 PROFIBUS DP

PROFIBUS DP является международным открытым стандартом для полевой шины и используется во всех случаях, где требуется быстрое время реакции при преимущественно небольшом объеме данных.

Ведущее устройство HIMA PROFIBUS DP и ведомое устройство HIMA PROFIBUS DP отвечают критериями европейского стандарта EN 50170 [7] и международного стандарта МЭК 61158 для PROFIBUS DP.

Ведущее устройство HIMA PROFIBUS DP может производить циклический и ациклический обмен данными с ведомым устройством PROFIBUS DP.

Для ациклического обмена данными в SiLworX имеются различные функциональные блоки. С помощью данных функциональных блоков Вы можете оптимально отрегулировать ведущее устройство HIMA PROFIBUS DP и ведомое устройство PROFIBUS DP в соответствии с требованиями вашего проекта.

Избыточное соединение PROFIBUS DP может быть достигнуто только с помощью конфигурации второго ведущего/ведомого устройства PROFIBUS DP и согласования в прикладной программе.

- Ведущее устройство PROFIBUS DP (см. главу 5.1)
- Ведомое устройство PROFIBUS DP (см. главу 5.13)

5.1 Ведущее устройство HIMA PROFIBUS DP

В данной главе описываются свойства ведущего устройства HIMA PROFIBUS DP, а также функции меню и диалоговые окна в SILworX, которые потребуются Вам для конфигурирования ведущего устройства HIMA PROFIBUS DP.

Необходимое оснащение и требования к системе:

Элемент	Описание
HIMA controller	HIMax с модулем COM
COM module	Модуль COM HIMax на используемом последовательном интерфейсе полевой шины (FB1 или FB2) должен быть оснащен опциональным submodule ведущего устройства HIMA PROFIBUS DP, см. главу 3.5.
Activation	Активация посредством вставного модуля, см. главу 3.5.2.

Таблица 29: Необходимое оснащение и требования к системе

Свойства ведущего устройства PROFIBUS DP:

Элемент	Описание
Type of HIMA PROFIBUS DP master	Ведущее устройство DP-V1 класс 1 с дополнительными функциями DP-V2
Transfer rate	9,6 кбит/с...12 Мбит/с
Bus address	0...125
Max. number of PROFIBUS DP masters	Для каждого модуля COM Вы можете конфигурировать два ведущих устройства PROFIBUS DP.
Max. number of PROFIBUS DP slaves	Вы можете конфигурировать до 122 ведомых устройств на ресурс (во всех экземплярах протокола ведущего устройства). При этом имеется ограничение, что к одному сегменту шины без промежуточного усилителя может быть подключено максимум 31 ведомое устройство.
Maximum process data length To a slave	Вывод DP =: макс. 244 байта Ввод DP =: макс. 244 байта

Таблица 30: Свойства ведущего устройства PROFIBUS DP

Согласно норме допускается всего три промежуточных усилителя, таким образом, возможно максимум 122 участника шины на один последовательный интерфейс ведущего устройства.

5.1.1 Создание ведущего устройства HIMA PROFIBUS DP

Чтобы создать новое ведущее устройство HIMA PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols**.
2. В контекстном меню протокола выбрать **New, PROFIBUS DP Master**, чтобы добавить новое ведущее устройство PROFIBUS DP.
3. Выбрать в контекстном меню ведущего устройства PROFIBUS DP **Properties, General**.
4. Выбрать **Modul** и **Interfaces**.

5.2 Пример PROFIBUS DP

На данном примере ведущее устройство HIMA PROFIBUS DP осуществляет обмен переменными с ведомым устройством HIMA PROFIBUS DP.

Здесь содержится информация о том, как создать и параметризовать ведущее устройство HIMA PROFIBUS DP и ведомое устройство HIMA PROFIBUS DP.

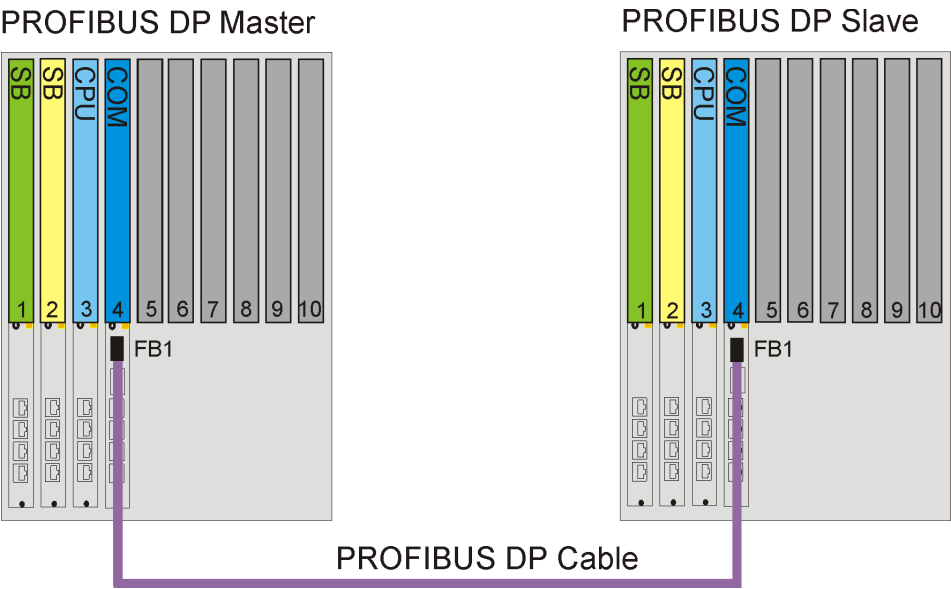


Рисунок 25: Связь через PROFIBUS DP

Модули COM обеих систем управления HIMA на интерфейсе полевой шины 1 должны быть оснащены соответствующим submodule PROFIBUS DP, см. главу 3.5.

Для данного примера в SILworX необходимо создать следующие глобальные переменные:

Глобальная переменная	Тип
PB_Slave_Master1	UINT
PB_Slave_Master2	DWORD
PB_Slave_Master3	DWORD
PB_Slave_Master4	BYTE
PB_Master_Slave1	DWORD
PB_Master_Slave2	BYTE

5.2.1 Конфигурирование ведомого устройства PROFIBUS DP

Конфигурация ведомого устройства PROFIBUS DP.

Чтобы создать новое ведомое устройство HIMA PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols**.
2. В контекстном меню протокола выбрать **New, PROFIBUS DP Slave**, чтобы добавить новое ведомое устройство PROFIBUS DP.
3. В контекстном меню ведомого устройства PROFIBUS DP выбрать **Edit**.
4. В вкладке **Properties** выбрать **COM-Module** и **Interfaces** (напр., FB1).

Присвоение переменных в ведомом устройстве HIMA PROFIBUS DP:

1. В контекстном меню ведомого устройства PROFIBUS DP выбрать **Edit**.
2. В диалоговом окне **Edit** выбрать вкладку **Process Variables**.



Начальный адрес входных и выходных переменных ведомого устройства HIMA PROFIBUS DP всегда начинается с 0. Если ведущее устройство PROFIBUS DP (другого изготовителя) ожидает начальный адрес более высокого уровня, то перед исходными переменными вводятся фиктивные переменные.

Выходы в ведомом устройстве HIMA PROFIBUS DP

Название	Тип	Смещение	Глобальная переменная
PB_Slave_Master1	UINT	0	PB_Slave_Master1
PB_Slave_Master2	DWORD	2	PB_Slave_Master2
PB_Slave_Master3	DWORD	6	PB_Slave_Master3
PB_Slave_Master4	BYTE	10	PB_Slave_Master4

Таблица 31: Выходы ведомого устройства HIMA PROFIBUS DP

1. В выборе окна перетащить глобальные переменные для передачи с помощью Drag&Drop в область **Output Variables**.



Выходные переменные ведомого устройства HIMA PROFIBUS DP в данном примере состоят из **четырёх переменных** общим размером **11 байтов**. Выходная переменная с минимальным смещением имеет начальный адрес **0**.

2. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Output Variables**.
3. Выбрать в контекстном меню **New Offsets**, чтобы заново генерировать смещения переменных.

Входы в ведомом устройстве HIMA PROFIBUS DP

Название	Тип	Смещение	Глобальная переменная
PB_Master_Slave1	DWORD	14	PB_Master_Slave1
PB_Master_Slave2	BYTE	18	PB_Master_Slave2

Таблица 32: Входы ведомого устройства HIMA PROFIBUS DP

1. В выборе окна перетащить глобальные переменные для приема с помощью Drag&Drop в область **Input Variables**.



Входные переменные ведомого устройства HIMA PROFIBUS DP состоят в данном примере из **двух переменных** общим размером **3 байта**. Входная переменная с минимальным смещением имеет начальный адрес **0**.

2. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Input Variables**.
3. Выбрать в контекстном меню **New Offsets**, чтобы заново генерировать смещения переменных.

Подтверждение конфигурации ведомого устройства PROFIBUS DP:

1. В структуре открыть **Configuration, Resource, Protocols, PROFIBUS DP Slave**.
2. Щелкнуть на кнопку **Verification** в строке меню и подтвердить с помощью **OK**.
3. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.



Конфигурацию ведомого устройства PROFIBUS DP необходимо компилировать заново вместе с прикладной программой ресурса ведомого устройства PROFIBUS DP и загрузить в системы управления, чтобы они были активны для связи PROFIBUS DP.

5.2.2 Конфигурирование ведущего устройства PROFIBUS DP

Чтобы создать новое ведущее устройство HIMA PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols**.
2. В контекстном меню протокола выбрать **New, PROFIBUS DP Master**, чтобы добавить новое ведущее устройство PROFIBUS DP.
3. В контекстном меню ведущего устройства PROFIBUS DP выбрать **Properties, General**.
4. Во вкладке **General** выбрать **модуль COM** и **Interfaces** (напр., FB1).



Следующие действия служат для конфигурирования ведомого устройства HIMA PROFIBUS DP в ведущем устройстве HIMA PROFIBUS DP.

Чтобы создать ведомое устройство HIMA PROFIBUS DP в ведущем устройстве PROFIBUS DP, необходимо:

1. В контекстном меню ведущего устройства PROFIBUS DP выбрать **New, PROFIBUS DP Slave**.

Чтобы создать файл GSD для нового ведомого устройства PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS Slave**.
2. В контекстном меню ведущего устройства PROFIBUS DP выбрать **Read GSD File** и выбрать относящийся к ведомому устройству PROFIBUS файл GSD (напр., hax100ea.gsd).



Файлы GSD для систем управления HIMA Вы найдете на интернет-сайте компании HIMA www.hima.com.

Создание модулей HIMax PROFIBUS DP

В ведущем устройстве PROFIBUS DP Вы должны конфигурировать число фактически передаваемых байтов. Это можно сделать посредством выбора модулей, пока не будет достигнута физическая конфигурация ведомого устройства.

i

Число используемых модулей не имеет значения для получения необходимого числа байтов, если максимальное количество модулей равно 32 не превышено.

Чтобы не усложнять излишне конфигурацию ведущего устройства PROFIBUS DP, количество выбранных модулей должно быть по возможности минимальным.

Чтобы создать необходимые модули PROFIBUS DP, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS Slave**.
2. В строке меню выбрать **PROFIBUS DP Master, Add modules**.
3. Для данного примера выбрать следующие подходящие модули, чтобы принять от ведомого устройства PROFIBUS DP **11 байтов** и передать **3 байта**.

Чтобы пронумеровать модули PROFIBUS DP, необходимо:

1. Щелкнуть правой кнопкой мыши на первый **PROFIBUS DP Modul** и выбрать в контекстном меню **Properties**.
2. Введите **0** в поле **Slot**.
3. Повторить данное действие для других **PROFIBUS DP Modul** и пронумеровать по порядку.

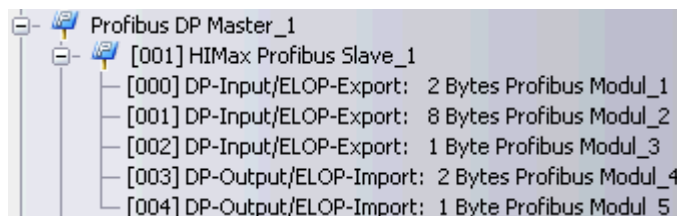


Рисунок 26: Ведомое устройство HIMax PROFIBUS DP с модулями

i

Нумеруйте модули HIMax PROFIBUS DP, начиная с **0**, в порядке возрастания без пробелов.

При этом последовательность модулей PROFIBUS DP для работы не имеет значения. Для более удобного обзора необходимо создавать модули ввода DP и модули вывода DP упорядоченно.

Конфигурирование модулей ввода/вывода

i

Сумма переменных в байтах должна совпадать с размером соответствующего модуля в байтах.

Чтобы конфигурировать модуль ввода [000] DP Input/ELOP Export: 2 Bytes:

1. В ведомом устройстве PROFIBUS DP выбрать модуль ввода **[000] DP Input/ELOP Export: 2 Bytes**
2. Щелкнуть правой кнопкой мыши на модуль ввода и выбрать в контекстном меню **Edit**.
3. В диалоговом окне **Edit** выбрать вкладку **Process Variables**.
4. В выборе объекта выбрать подходящую переменную и с помощью Drag&Drop перетащить в область **Input Signals** модуля ввода **[000] DP Input/ELOP Export: 2 Bytes**.

Название	Тип	Смещение	Глобальная переменная
PB_Slave_Master1	UINT	0	PB_Slave_Master1

Таблица 33: Переменные модуля ввода [000] DP Input/ELOP Export: 2 Bytes

5. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Input Signals**.
6. Выбрать в контекстном меню **New Offsets**, чтобы заново генерировать смещения переменных.

Чтобы конфигурировать модуль ввода [001] DP Input/ELOP Export: 8 Bytes:

1. В ведомом устройстве PROFIBUS DP выбрать модуль ввода **[001] DP Input/ELOP Export: 8 Bytes**.
2. Щелкнуть правой кнопкой мыши на модуль ввода и выбрать в контекстном меню **Edit**.
3. В диалоговом окне **Edit** выбрать вкладку **Process Variables**.
4. В выборе объекта выбрать подходящую переменную и с помощью Drag&Drop перетащить в область **Input Signals** модуля ввода **[001] DP Input/ELOP Export: 8 Bytes**.

Название	Тип	Смещение	Глобальная переменная
PB_Slave_Master2	DWORD	0	PB_Slave_Master2
PB_Slave_Master3	DWORD	4	PB_Slave_Master3

Таблица 34: Переменные модуля ввода [001] DP Input/ELOP Export: 8 Bytes

5. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Input Signals**.
6. Выбрать в контекстном меню **New Offsets**, чтобы заново генерировать смещения переменных.

Чтобы конфигурировать модуль ввода [002] DP Input/ELOP Export: 1 Byte:

1. В ведомом устройстве PROFIBUS DP выбрать модуль ввода **[002] DP Input/ELOP Export: 1 Byte**.
2. Щелкнуть правой кнопкой мыши на модуль ввода и выбрать в контекстном меню **Edit**.
3. В диалоговом окне **Edit** выбрать вкладку **Process Variables**.
4. В выборе объекта выбрать подходящую переменную и с помощью Drag&Drop перетащить в область **Input Signals** модуля ввода **[002] DP Input/ELOP Export: 1 Byte**.

Название	Тип	Смещение	Глобальная переменная
PB_Slave_Master4	BYTE	0	PB_Slave_Master4

Таблица 35: Переменные модуля ввода [002] DP Input/ELOP Export: 1 Byte

5. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Input Signals**.
6. Выбрать в контекстном меню **New Offsets**, чтобы заново генерировать смещения переменных.

Чтобы конфигурировать модуль вывода [003] DP Output/ELOP Import 2 Bytes:

1. Выберите в ведомом устройстве PROFIBUS DP модуль вывода **[003] DP Output/ELOP Import 2 Bytes**.
2. Щелкнуть правой кнопкой мыши на модуль вывода и выбрать в контекстном меню **Edit**.
3. В диалоговом окне **Edit** выбрать вкладку **Process Variables**.
4. В выборе объекта выбрать подходящую переменную и с помощью Drag&Drop перетащить в область **Output Signals** модуля вывода **[003] DP Output/ELOP Import 2 Bytes**.

Название	Тип	Смещение	Глобальная переменная
PB_Master_Slave1	UINT	0	PB_Master_Slave1

Таблица 36: Переменные модуля вывода [003] DP Output/ELOP Import: 2 Bytes

5. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Output Signals**.
6. Выбрать в контекстном меню **New Offsets**, чтобы заново генерировать смещения переменных.

Чтобы конфигурировать модуль вывода [004] DP Output/ELOP Import 1 Byte:

1. Выберите в ведомом устройстве PROFIBUS DP модуль вывода **[004] DP Output/ELOP Import 1 Byte**.
2. Щелкнуть правой кнопкой мыши на модуль вывода и выбрать в контекстном меню **Edit**.
3. В диалоговом окне **Edit** выбрать вкладку **Process Variables**.
4. В выборе объекта выбрать подходящую переменную и с помощью Drag&Drop перетащить в область **Output Signals** модуля вывода **[004] DP Output/ELOP Import 1 Byte**.

Название	Тип	Смещение	Глобальная переменная
PB_Master_Slave2	BYTE	0	PB_Master_Slave2

Таблица 37: Переменные модуля вывода [004] DP Output/ELOP Import: 1 Byte

5. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Output Signals**.
6. Выбрать в контекстном меню **New Offsets**, чтобы заново генерировать смещения переменных.

Создание данных пользователя в ведущем устройстве PROFIBUS DP

Чтобы создать данные пользователя в ведущем устройстве PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master**.
2. Щелкнуть правой кнопкой мыши на **PROFIBUS Master** и выбрать в контекстном меню **Properties**.

3. Выбрать вкладку **Data** и щелкнуть кнопки ... рядом с данными пользователя.

В поле данных пользователя размером 32 байта определяются начальный адрес и число переменных блока, см. также главу 5.8.

4. Для данного примера необходимо ввести следующие данные пользователя:
 - 4, чтобы были приняты **четыре переменных** от ведущего устройства PROFIBUS DP.
 - 2, чтобы были переданы **две переменных** от ведущего устройства PROFIBUS DP.
 Начальный адрес блока ввода и вывода начинается с **0**.

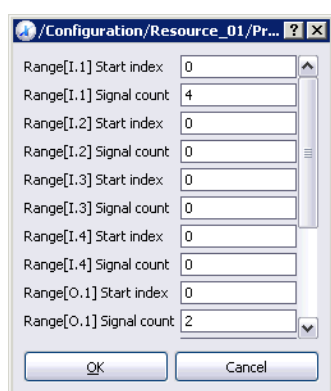


Рисунок 27: Поле данных пользователя

Подтверждение конфигурации ведомого устройства PROFIBUS DP:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master**.
2. Щелкнуть на кнопку **Verification** в строке меню и подтвердить с помощью **ОК**.
3. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.



Рисунок 28: Диалоговое окно Verification



Конфигурацию ведущего устройства PROFIBUS DP необходимо компилировать заново вместе с прикладной программой ресурса ведущего устройства PROFIBUS DP и загрузить в системы управления, чтобы они были активны для связи PROFIBUS DP.

Оптимизация параметров PROFIBUS DP

При использовании значений по умолчанию для параметров PROFIBUS подразумевается бесперебойная связь PROFIBUS. С помощью дальнейшей оптимизации настроек Вы можете ускорить обмен данными и улучшить результат распознавания ошибок.

Чтобы определить фактическое контрольное время обращения (Target Rotation Time TTR [ms]), необходимо:

1. Открыть панель управления ведущего устройства HIMax PROFIBUS DP.
2. В структуре панели управления выбрать **PROFIBUS DP Master** и считать фактическое **Target Rotation Time TTR [ms]** (контрольное время обращения). Запомните данное значение.

Чтобы определить необходимые параметры ведомого устройства PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS DP Slave**.
2. Щелкнуть правой кнопкой мыши на **HIMax PROFIBUS Slave** и выбрать **Properties**.
3. Выбрать вкладку **Features** и считать параметр **Min. Slave Interval MSI [ms]** для данного ведомого устройства PROFIBUS DP. Запомните данное значение.
4. Выбрать вкладку **Baud rates** и считать параметр **Max. Tsdr** для используемой скорости передачи. Запомните данное значение.

Чтобы ввести полученные параметры, необходимо:

1. Щелкнуть правой кнопкой мыши на **PROFIBUS Master** и в контекстном меню **Properties**.
2. Выбрать вкладку **Timings**.

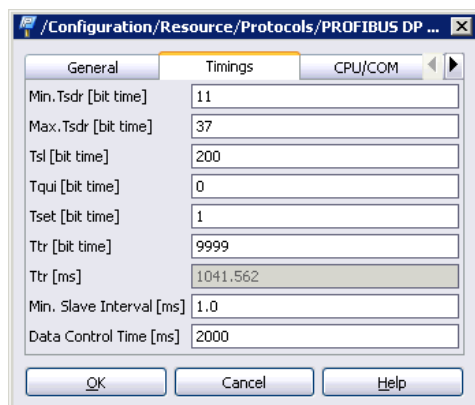


Рисунок 29: Свойства ведущего устройства PROFIBUS DP

3. Считанные **Max. Tsdr** преобразовать в **bit Time** и ввести значения.
4. Считанное значение **Target Rotation Time TTR [ms]** преобразовать в **bit Time** и, прибавив 1/3 коэффициент безопасности, ввести в поле **Target Rotation Time TTR [ms]**.
5. Ввести считанное значение **Min. Slave Interval MSI [ms]**.

i

Если конфигурировано несколько ведомых устройств, то должно использоваться максимальное значение MaxTsdr [bit time] и максимальное значение Min. Slave Intervall [ms].

6. Контрольное время полезной нагрузки [мс] необходимо настроить $\geq 6 \cdot Ttr$ [ms].

Ввод времени сторожевого устройства для ведомого устройства PROFIBUS-DP

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS DP Slave**.
2. Щелкнуть правой кнопкой мыши на **HIMax PROFIBUS Slave** и выбрать **Properties**.

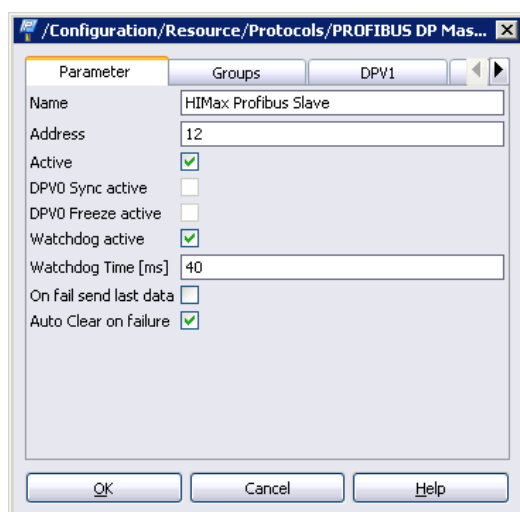


Рисунок 30: Свойства ведомого устройства PROFIBUS DP

3. Выбрать вкладку **Parameters** и поставить галочку в кнопке-флажке **Watchdog Active**.
4. Внести время сторожевого устройства $[ms] \geq 6 \cdot T_{tr} [ms]$ в поле **Watchdog Time [ms]**.

i

Конфигурацию ведущего и ведомого устройства PROFIBUS DP необходимо компилировать заново вместе с прикладной программой ресурсов ведущего и ведомого устройства PROFIBUS DP и загрузить в системы управления, чтобы они были активны для связи PROFIBUS DP.

5.3 Функции меню ведущего устройства PROFIBUS DP

5.3.1 Редактирование (Edit)

Функция меню **Edit** из контекстного меню ведущего устройства PROFIBUS DP открывает диалоговое окно **Edit**.

Вкладка **System Variables** содержит следующие системные переменные, которые позволяют анализировать состояние ведущего устройства PROFIBUS DP в прикладной программе.

Элемент	Описание																
Error count [DWORD]	Количество ошибок с момента сброса статистики.																
Baud rate [DWORD]	Скорость передачи в бодах (бит/с), с которой эксплуатируется шина.																
Bus error [BYTE]	<p>Если возникает ошибка шины, то в системной переменной <i>Bus Error</i> появляется код ошибки. Код ошибки отображается до тех пор, пока ошибка шины не будет устранена.</p> <table border="1"> <thead> <tr> <th>Код</th><th>Значение</th></tr> </thead> <tbody> <tr> <td>0</td><td>ОК, нет ошибки шины</td></tr> <tr> <td>1</td><td>Ошибка в адресе: Адрес ведущего устройства на шине уже существует.</td></tr> <tr> <td>2</td><td>Неисправность шины На шине зарегистрирована неисправность, (напр., неверно подключена оконечная нагрузка шины, несколько участников осуществляют передачу одновременно).</td></tr> <tr> <td>3</td><td>Ошибка протокола Принят пакет с неправильной кодировкой.</td></tr> <tr> <td>4</td><td>Ошибка аппаратного обеспечения Получено сообщение об ошибке аппаратного обеспечения, напр., при слишком коротком настроенном времени.</td></tr> <tr> <td>5</td><td>Неизвестная ошибка Ведущее устройство изменило состояние по неизвестной причине.</td></tr> <tr> <td>6</td><td>Сброс значений контроллера При серьезных неисправностях шины происходит сброс микросхемы контроллера.</td></tr> </tbody> </table> <p>Чтобы проанализировать переменную состояния <i>Bus Error</i> в прикладной программе, должно иметься соединение с переменной.</p>	Код	Значение	0	ОК, нет ошибки шины	1	Ошибка в адресе: Адрес ведущего устройства на шине уже существует.	2	Неисправность шины На шине зарегистрирована неисправность, (напр., неверно подключена оконечная нагрузка шины, несколько участников осуществляют передачу одновременно).	3	Ошибка протокола Принят пакет с неправильной кодировкой.	4	Ошибка аппаратного обеспечения Получено сообщение об ошибке аппаратного обеспечения, напр., при слишком коротком настроенном времени.	5	Неизвестная ошибка Ведущее устройство изменило состояние по неизвестной причине.	6	Сброс значений контроллера При серьезных неисправностях шины происходит сброс микросхемы контроллера.
Код	Значение																
0	ОК, нет ошибки шины																
1	Ошибка в адресе: Адрес ведущего устройства на шине уже существует.																
2	Неисправность шины На шине зарегистрирована неисправность, (напр., неверно подключена оконечная нагрузка шины, несколько участников осуществляют передачу одновременно).																
3	Ошибка протокола Принят пакет с неправильной кодировкой.																
4	Ошибка аппаратного обеспечения Получено сообщение об ошибке аппаратного обеспечения, напр., при слишком коротком настроенном времени.																
5	Неизвестная ошибка Ведущее устройство изменило состояние по неизвестной причине.																
6	Сброс значений контроллера При серьезных неисправностях шины происходит сброс микросхемы контроллера.																
Average cycle time [DWORD]	Измеренное среднее время цикла шины в миллисекундах.																
Last cycle time [DWORD]	Измеренное время цикла шины в миллисекундах.																
Master State [BYTE]	<p>Отображает текущее состояние протокола.</p> <p>0: OFFLINE 1: STOP 2: CLEAR 3: OPERATE</p> <p>Чтобы проанализировать переменную состояния ведущего устройства в прикладной программе, должно быть установлено соединение с переменной.</p>																

Элемент	Описание
Maximum cycle time [DWORD]	Измеренное максимальное время цикла шины в миллисекундах.
Min. Slave Interval [DWORD]	Измеренный минимальный интервал ведомого устройства, присвоенного данному ведущему устройству.
Minimum cycle time [DWORD]	Измеренное минимальное время цикла шины в миллисекундах.
Target rotation time [DWORD]	Контрольное время обращения маркера.

Таблица 38: Системные переменные ведущего устройства PROFIBUS DP

5.3.2 Функция меню Свойства

Функция меню **Properties** из контекстного меню ведущего устройства PROFIBUS DP открывает диалоговое окно **Properties**.

Диалоговое окно содержит следующие вкладки:

Вкладка Общее

Элемент	Описание																																												
Type	Ведущее устройство PROFIBUS DP																																												
Name	Любое однозначное имя для ведущего устройства PROFIBUS DP																																												
Module	Выбор модуля COM, на котором обрабатывается данный протокол.																																												
Use Max CPU Load	Активировано: Сохранить предельное значение нагрузки ЦПУ из поля <i>Max. CPU Load [%]</i> . Деактивировано: Не использовать предельное значение нагрузки ЦПУ для данного протокола.																																												
Max. CPU Load [%]	Максимальная нагрузка модуля ЦПУ, которая может создаваться при обработке протокола. Диапазон значений: 1...100% Значение по умолчанию: 30%																																												
Address	Адрес станции ведущего устройства. На шине может быть только один адрес станции ведущего устройства. Диапазон значений: 0...125 Значение по умолчанию: 0																																												
Interface	Интерфейс COM, который должен использоваться для ведущего устройства. Диапазон значений: FB1, FB2																																												
Baud rate	Скорость передачи в бодах (бит/с), с которой эксплуатируется шина. Возможные значения: <table><tr><th>Значение</th><th>Скорость передачи в бодах</th><th>FB1</th><th>FB2</th></tr><tr><td>9600</td><td>9,6 кбит/с</td><td>X</td><td>X</td></tr><tr><td>19200</td><td>19,2 кбит/с</td><td>X</td><td>X</td></tr><tr><td>45450</td><td>45,45 кбит/с</td><td>X</td><td>X</td></tr><tr><td>93750</td><td>93,75 кбит/с</td><td>X</td><td>X</td></tr><tr><td>187500</td><td>187,5 кбит/с</td><td>X</td><td>X</td></tr><tr><td>500000</td><td>500 кбит/с</td><td>X</td><td>X</td></tr><tr><td>1500000</td><td>1,5 Мбит/с</td><td>X</td><td>X</td></tr><tr><td>3000000</td><td>3 Мбит/с</td><td>X</td><td>-</td></tr><tr><td>6000000</td><td>6 Мбит/с</td><td>X</td><td>-</td></tr><tr><td>12000000</td><td>12 Мбит/с</td><td>X</td><td>-</td></tr></table>	Значение	Скорость передачи в бодах	FB1	FB2	9600	9,6 кбит/с	X	X	19200	19,2 кбит/с	X	X	45450	45,45 кбит/с	X	X	93750	93,75 кбит/с	X	X	187500	187,5 кбит/с	X	X	500000	500 кбит/с	X	X	1500000	1,5 Мбит/с	X	X	3000000	3 Мбит/с	X	-	6000000	6 Мбит/с	X	-	12000000	12 Мбит/с	X	-
Значение	Скорость передачи в бодах	FB1	FB2																																										
9600	9,6 кбит/с	X	X																																										
19200	19,2 кбит/с	X	X																																										
45450	45,45 кбит/с	X	X																																										
93750	93,75 кбит/с	X	X																																										
187500	187,5 кбит/с	X	X																																										
500000	500 кбит/с	X	X																																										
1500000	1,5 Мбит/с	X	X																																										
3000000	3 Мбит/с	X	-																																										
6000000	6 Мбит/с	X	-																																										
12000000	12 Мбит/с	X	-																																										

Таблица 39: Общие свойства ведущего устройства PROFIBUS DP

Вкладка Время

Элемент	Описание
MinTsdrr [bit time]	Min. Station Delay Time: Минимальное время, которое должно пройти, прежде чем ведомое устройство PROFIBUS DP может ответить. Диапазон значений: 11...1023 Значение по умолчанию: 11
MaxTsdrr [bit time]	Max. Station Delay Time: Максимальное время, которое требуется ведомому устройству PROFIBUS DP, чтобы ответить. Max Tsdrr ≥ Tsdrr (подключенного ведомого устройства с максимальным Tsdrr) Значения MaxTsdrr ведомого устройства считываются из файлов GSD и находятся в диалоговом окне Properties ведомого устройства во вкладке Baud rates . Диапазон значений: 37...65525 Значение по умолчанию: 37
Tsl [bit time]	Slot Time Максимальный промежуток времени, на протяжении которого ведущее устройство ожидает ответа ведомого устройства. $Tsl > MaxTsdrr + 2 * Tset + Tqui + 13$ Диапазон значений: 37...16383 Значение по умолчанию: 37
Tqui [bit time]	Quiet Time for Modulator (время переключения модулятора) Время, необходимое участнику для переключения с передачи на прием. Диапазон значений: 0...493 Значение по умолчанию: 0
Tset [bit time]	Setup Time Время реакции на событие Диапазон значений: 1...494 Значение по умолчанию: 1
Ttr [bit time]	Контрольное время обращения маркера Максимально доступное время для обращения маркера. Нижнюю оценку Ttr получают с помощью расчета, см. главу 5.4.4. Диапазон значений: 256...16777215 Значение по умолчанию: 999
Ttr [ms]	Фактическое контрольное время обращения маркера в мс
Min. Slave Interval [ms]	Минимальное время, которое должно пройти между двумя циклическими запросами ведомого устройства. Оно соблюдается ведущим устройством и ни в коем случае не должно изменяться путем уменьшения значения. Цикл PROFIBUS DP может быть продлен, если изохронный режим не активен, и число ациклических телеграмм в одном цикле возрастает. Значение для <i>Min. Slave Interval</i> ведомого устройства считывается из файла GSD и находится в диалоговом окне Properties ведомого устройства во вкладке Features . В изохронном режиме <i>Min. Slave Interval</i> задает время изохронного цикла.

Элемент	Описание
Min. Slave Interval [ms] (continuation)	Изохронный режим активируется, если активированы опции изохронного режима Sync или Freeze. См. также время обновления между ЦПУ и COM (вкладка ЦПУ/COM). Диапазон значений: 0...6553.5 Значение по умолчанию: 30%
Data Control Time [ms]	Промежуток времени, за который ведущее устройство должно сообщать свое актуальное состояние на шину. Ориентировочное значение: Контрольное время полезной нагрузки = время сторожевого устройства ведомого устройства Диапазон значений: 0...65535 [10 мс] Значение по умолчанию: 2000

Таблица 40: Вкладка Время в диалоговом окне свойств ведущего устройства PROFIBUS DP

Вкладка ЦПУ/COM

Заданные значения для параметров обеспечивают высокоскоростной обмен данными PROFIBUS DP между модулем COM и модулем ЦПУ в системе управления HiMax. Данные параметры необходимо изменять только в том случае, если снижение нагрузки COM и/или ЦПУ требуется для применения, и это допустимо для процесса.

- i** Изменение параметров рекомендуется доверять только опытному программисту. Увеличение времени обновления COM и ЦПУ означает также, что увеличится фактическое время обновления данных PROFIBUS DP. Проверить временные требования устройства.

Соблюдайте также параметр *Min. Slave Interval [ms]*, который определяет время обновления данных PROFIBUS DP, передаваемых на ведомое устройство PROFIBUS DP и получаемых от него. Его можно увеличивать в соответствии со временем обновления COM/ЦПУ.

Элемент	Описание
Refresh Rate [ms]	Время обновления в миллисекундах, когда происходит обмен данными протокола между COM и ЦПУ. Если значение <i>Refresh Rate</i> равняется нулю или меньше, чем время цикла системы управления, то обмен данными осуществляется так быстро, насколько это возможно. Диапазон значений: 0...(2 ³¹ -1) Значение по умолчанию: 0
Within one cycle	Активировано: Обмен всеми данными протокола с ЦПУ на COM в течение одного цикла ЦПУ. Деактивировано: Обмен всеми данными протокола с ЦПУ на COM, распределенный на несколько циклов ЦПУ по 900 байтов на направление. Таким образом, при необходимости можно снизить время цикла системы управления. Значение по умолчанию: Активировано

Таблица 41: Вкладка ЦПУ/COM в диалоговом окне свойств ведущего устройства PROFIBUS DP

Вкладка Прочее

Элемент	Описание
Max. Retry Limit	Максимальное количество повторов передачи с ведущего устройства, если ведомое устройство не отвечает. Диапазон значений: 0...7 Значение по умолчанию: 1
Highest Active Address	Highest Station Address (HSA) Максимальный ожидаемый адрес станции ведущего устройства. Ведущие устройства с адресами станций, выходящими за рамки HSA, не входят в маркерное кольцо. Диапазон значений: 0...125 Значение по умолчанию: 125
Isochron Mode Sync	Изохронный режим Sync позволяет осуществлять регулирование с тактовой синхронизацией в ведущем и ведомом устройстве и одновременную активацию физических выходов нескольких ведомых устройств. Если изохронный режим Sync активен, то ведущее устройство передает команду управления «Sync» в виде широковещательной телеграммы на все ведомые устройства. Как только ведомые устройства, поддерживающие изохронный режим, получают команду управления «Sync», они синхронно переключают данные из прикладной программы на физические выходы. Значения физических выходов остаются неизменными до следующей команды Sync. Время цикла задается в «Min. Slave Intervall». Условие: $T_{tr} < \text{Min. Slave Intervall}$ Значение по умолчанию: Деактивировано
Isochron Mode Freeze	Изохронный режим Freeze позволяет осуществлять синхронный прием входных данных нескольких ведомых устройств. Если изохронный режим Freeze активен, то ведущее устройство передает команду управления «Freeze» в виде широковещательной телеграммы на все ведомые устройства. Как только ведомые устройства, поддерживающие изохронный режим, получают команду «Freeze», переменные физических входов фиксируются на текущем значении. Значения можно в этом случае считать с ведущего устройства. Входные данные обновляются только после следующей команды управления «Freeze». Время цикла задается в «Min. Slave Intervall». Условие: $T_{tr} < \text{Min. Slave Intervall}$ Значение по умолчанию: Деактивировано
Auto clear on error	Ведущее устройство переходит в состояние CLEAR, если ведомое устройство, для которого установлен параметр «Auto clear on error», выходит из строя. Значение по умолчанию: Деактивировано
Time Master	Ведущее устройство является также ведущим устройством времени и периодически передает через шину время системы. Значение по умолчанию: Деактивировано
Clock Sync Interval [ms]	Интервал синхронизации времени. Промежуток времени, в течение которого ведущее устройство времени передает на шину время системы. Диапазон значений: 0...65535 Значение по умолчанию: 0

Таблица 42: Прочие свойства ведущего устройства PROFIBUS DP

5.4 Метод доступа к шине PROFIBUS-DP

Метод доступа к шине предоставляет каждому участнику шины определенное окно времени, в течение которого участник шины может выполнять свою задачу по коммуникации.

5.4.1 Протокол ведущего/ведомого устройства

Распределение шины между ведущим и ведомым устройством PROFIBUS-DP выполняется по методу ведущего/ведомого устройства.

Активное ведущее устройство PROFIBUS-DP осуществляет связь с пассивным ведомым устройством PROFIBUS-DP.

Ведущее устройство PROFIBUS-DP, имеющее маркер, может осуществлять передачу и коммуникацию с присвоенным ему ведомым устройством PROFIBUS-DP. Ведущее устройство назначает шину ведомому устройству на определенное время, в течение которого ведомое устройство должно ответить.

5.4.2 Протокол маркера

Распределение шины между устройствами автоматизации (ведущее устройство, класс 1) и/или программирующими устройствами (ведущее устройство, класс 2) осуществляется посредством маркерной передачи данных.

Все ведущие устройства PROFIBUS-DP, подключенные к одной шине, образуют маркерное кольцо. Активное ведущее устройство PROFIBUS-DP, имеющее маркер, выполняет в это время функцию ведущего устройства на шине.

Ведущие устройства PROFIBUS-DP расположены в маркерном кольце по возрастающим адресам станций, и в этой последовательности маркер передается к ведущему устройству PROFIBUS-DP с максимальным адресом станции.

Оно передает маркер на ведущее устройство с минимальным адресом станции, чтобы замкнуть маркерное кольцо.

Контрольное время обращения маркера соответствует однократному обращению маркера через все ведущие устройства PROFIBUS-DP. Контрольное время обращения T_{tr} – это максимально допустимое время обращения маркера.

5.4.3 Контрольное время обращения маркера (T_{tr})

Ориентировочные значения для различной скорости передачи.

При конфигурировании ведущего устройства PROFIBUS DP учитывать то, что часть параметров во вкладке **Timings** зависит от настроенной во вкладке **General** скорости передачи в бодах. Для первой (начальной) конфигурации используйте указанные в таблице ориентировочные значения. Далее эти значения будут оптимизированы.

	9,6k	19,2k	45,45k	93,75k	187,5k	500k	1,5M	3M	6M	12M
MinTsdr	11	11	11	11	11	11	11	11	11	11
MaxTsdr	60	60	400	60	60	100	150	250	450	800
Tsl bit time	100	100	640	100	100	200	300	400	600	1000
Tqui bit time	0	0	0	0	0	0	0	3	6	9
Tset bit time	1	1	95	1	1	1	1	4	8	16

Таблица 43: Ориентировочные значения $HIMax$ для контрольного времени обращения маркера при различной скорости передачи

Все временные данные указаны в терабитах (1 Тбит = 1/[бит/с]).

MinTsdr имеет размер минимум 11 Тбит, поскольку каждый символ состоит из 11 бит (1 стартовый бит, 1 стоповый бит, 1 бит контроля четности, 8 бит данных).

Время передачи для символа

Скорость передачи в бодах	Тбит бит = 1/Скорость передачи в бодах	Время
9600 бит/с	1/9600 = 104,166 мкс	11*104,166 мкс = 114,583 мс
6 Мбит/с	1/6*1066 = 166,667 нс	11*166,667 нс = 1,833 мкс

Таблица 44: Время передачи для символа при различной скорости передачи

5.4.4 Расчет контрольного времени обращения маркера T_{tr}

Минимальное контрольное время обращения маркера T_{tr} рассчитывается следующим образом:

$$T_{tr_{min}} = n * (198 + T_1 + T_2) + b * 11 + 242 + T_1 + T_2 + T_{sl}$$

Элемент	Значение
n	Количество активных ведомых устройств
b	Число байтов входных/выходных данных активных ведомых устройств (вход плюс выход)
T ₀	35 + 2 * T _{set} + T _{qui}
T ₁	Если T ₀ < MinT _{sdr} : T ₁ = MinT _{sdr} Если T ₀ > MinT _{sdr} : T ₁ = T ₀
T ₂	Если T ₀ < MaxT _{sdr} : T ₂ = MaxT _{sdr} Если T ₀ > MaxT _{sdr} : T ₂ = T ₀
T _{sl}	Slot Time Максимальный промежуток времени, в течение которого ведущее устройство ожидает ответа ведомого устройства
198	Два раза заголовок телеграммы различной длины (для запроса и ответа)
242	Global_Control, FDL_Status_Req и передача маркера

Таблица 45: Элементы для расчета контрольного времени обращения маркера

i

Оценка контрольного времени обращения маркера T_{tr} действительна только в том случае, если на шине эксплуатируется только одно ведущее устройство, не требуется повтор передачи и не происходит передача ациклических данных.

Ни в коем случае не настраивайте T_{tr} на значение, которое меньше значения, рассчитанного с помощью указанной формулы значение, поскольку в этом случае не гарантируется исправная работа. Компания HIMA рекомендует настраивать значение, которое в два или три раза больше рассчитанного значения.

Пример расчета контрольного времени обращения маркера T_{tr}

Задана следующая конфигурация:

5 активных ведомых устройств

(n = 5)

20 бит входных/выходных данных на ведомое устройство

(b = 100)

Следующие постоянные времени для скорости передачи 6 Мбит/с взяты из таблицы. см. таблица

i

Оценка контрольного времени обращения маркера T_{tr} действительна только в том случае, если на шине эксплуатируется только одно ведущее устройство, не требуется повтор передачи и не происходит передача ациклических данных.

Ни в коем случае не настраивайте T_{tr} на значение, которое меньше значения, рассчитанного с помощью указанной формулы значение, поскольку в этом случае не гарантируется исправная работа. Компания HIMA рекомендует настраивать значение, которое в два или три раза больше рассчитанного значения.

:

- $MinTsdr = 11 T_{bit}$
- $MaxTsdr = 450 T_{bit}$
- $T_{sl} \text{ Bit Time} = 600 T_{bit}$
- $T_{qui} \text{ Bit Time} = 6 T_{bit}$
- $T_{set} \text{ bit time} = 8 T_{bit}$

$$T_0 = 35 + 2 * T_{set} + T_{qui}$$

$$T_0 = 35 + 2 * 8 + 6$$

$$T_0 = 57 T_{bit}$$

Так как $T_0 > MinTsdr$: **$T_1 = T_0 = 57 T_{bit}$**

Так как $T_0 < MaxTsdr$: **$T_2 = MaxTsdr = 450 T_{bit}$**

Полученные значения подставить в формулу для минимального контрольного времени обращения маркера:

$$T_{tr_{min}} = n * (198 + T_1 + T_2) + b * 11 + 242 + T_1 + T_2 + T_{sl}$$

$$T_{tr_{min}} = 5 (198 + 57 + 450) + 100 * 11 + 242 + 57 + 450 + 600$$

$$T_{tr_{min}} [T_{бит}] = 5974 T_{бит}$$

Результат:

$$T_{tr_{min}} [\text{мкс}] = 5974 T_{бит} * 166,67 \text{ нс} = 995,68 \text{ мкс}$$

i

T_{tr} проверяется при вводе в диалоговом окне.

Если введенное значение T_{tr} меньше, чем рассчитанное в SILworX значение, то появляется сообщение об ошибке в строке индикации статуса. Дополнительно предлагается минимальное значение для T_{tr} .

Если выбран изохронный режим *Sync* или *Freeze*, то время цикла задается параметром *MinSlaveInterval*. Значение T_{tr} должно быть в любом случае меньше, чем *Min. Slave Interval*.

Несоблюдение данного условия в изохронном режиме ведет к появлению сообщения по ошибке.

5.5 Изохронный цикл PROFIBUS DP (начиная с DP-V2)

Цикл PROFIBUS DP состоит здесь из фиксированной, циклической и обусловленной событием, ациклической части телеграммы.

Ациклическая часть телеграммы в цикле PROFIBUS DP может его продлить соответствующим образом, что в определенных применениях, например, в силовых установках, нежелательно.

Чтобы достичь постоянного времени цикла (t_{const}), в ведущем устройстве активируется изохронный режим, при котором параметр *Min. Slave Interval [ms]* задает (t_{const}) постоянное время цикла. Параметрированный таким образом изохронный цикл PROFIBUS DP имеет точность такта с отклонением < 10 мс.

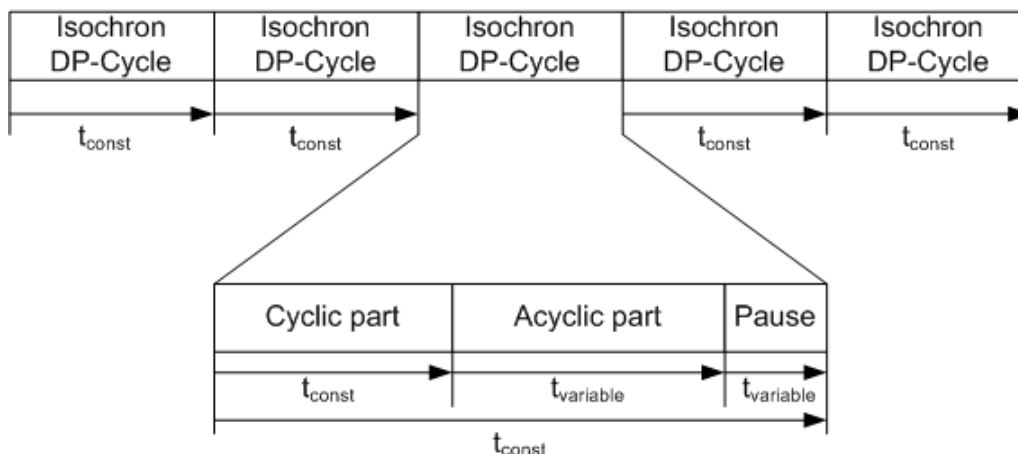


Рисунок 31: Изохронный цикл PROFIBUS DP

Чтобы определить циклическую долю, пользователь должен рассчитать минимальное контрольное время обращения маркера.

Дополнительно для ациклической доли должен быть зарезервирован достаточный промежуток времени (обычно двух- или трехкратное минимальное контрольное время обращения маркера T_{tr}). Если зарезервированное время не требуется, то перед следующим циклом делается пауза, чтобы поддерживать время цикла постоянным. (см. также главу 5.4.3, Контрольное время обращения маркера T_{tr}).

i

Ведущее устройство параметрируется с помощью *Min. Slave Interval [ms]* с использованием определенного пользователем времени цикла DP.

Для того чтобы изохронный режим был активен, должен быть активирован минимум один из двух параметров *Isochron Mode Sync* или *Isochron Mode Freeze* в ведущем устройстве.

На шине в изохронном режиме может эксплуатироваться только одно ведущее устройство. Другие ведущие устройства не допускаются.

5.5.1 Изохронный режим (начиная с DP-V2)

Данная функция позволяет осуществлять регулирование с тактовой синхронизацией в ведущем и ведомом устройстве, независимо от нагрузки шины. Цикл шины синхронизируется с тактовым отклонением < 10 мс. Благодаря этому могут быть реализованы высокоточные процессы позиционирования.

i

Преимущества *изохронного режима* в ограниченном виде могут использоваться ведомыми устройствами (ведомые устройства DP-V0), не поддерживающими *изохронный режим*. Для этого у ведомых устройств активируют *Slaves Sync* и/или *Freeze* и присваивают их группе 8.

Обычно используют режим *Sync* и *Freeze* одновременно.

5.5.2 Изохронный режим Sync (начиная с DP-V2)

Режим *Isochron Mode Sync* позволяет осуществлять регулирование с тактовой синхронизацией в ведущем и ведомом устройстве и одновременную активацию выходов нескольких ведомых устройств.

5.5.3 Изохронный режим Freeze (начиная с DP-V2)

Режим *Isochron Mode Freeze* позволяет осуществлять синхронный прием входных данных нескольких ведомых устройств.

5.6 Функции меню ведомого устройства PROFIBUS DP (в ведущем устройстве)

5.6.1 Создание ведомого устройства PROFIBUS DP (в ведущем устройстве)

Чтобы создать в ведущем устройстве PROFIBUS DP от HIMA ведомое устройство PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, Profibus DP Master**.
2. В контекстном меню ведущего устройства Profibus DP выбрать **New, PROFIBUS Slave**, чтобы добавить новое ведомое устройство PROFIBUS.

5.6.2 Редактирование (Edit)

Функция меню **Edit** из контекстного меню ведомого устройства PROFIBUS DP открывает диалоговое окно **System Variables**.

Вкладка **System Variables** содержит следующие системные переменные, которые позволяют анализировать состояние ведомого устройства PROFIBUS DP в прикладной программе.

Элемент	Описание	
Activation Control [BYTE]	Смена с 0 на 1 деактивирует ведомое устройство. Смена с 1 на 0 активирует деактивированное ранее ведомое устройство. Активировано = 0 Деактивировано = 1	
PNO Ident Nummer [WORD]	Присвоенный PROFIBUS DP Nutzerorganisation e.V. 16-битный номер, который однозначно обозначает продукт (полевое устройство).	
Standard Diagnostic [DWORD]	Посредством стандартной диагностики ведомое устройство сообщает ведущему устройству свое текущее состояние. Данная переменная всегда содержит последнюю принятую стандартную диагностику. Параметры соответствуют диагностической телеграмме согл. МЭК 61158.	
Connection Count [DWORD]	Увеличивается с каждым новым соединением. Считает с момента сброса счетчика.	
Connection State [BYTE]	Значение	Описание
	0	Деактивировано: загружаются наборы параметров для данных ведомых устройств, но ведомые устройства игнорируются. Входные данные устанавливаются на предустановленные значения по умолчанию, на шине не наблюдается никакой активности в отношении данных ведомых устройств.
	1	Неактивно (соединение отсутствует): Если ведомое устройство (больше) не доступно, входные данные устанавливаются на предустановленные значения по умолчанию. Для каждого ведомого устройства могут быть выбраны следующие опции: <ul style="list-style-type: none"> ведущее устройство передает выходные данные на ведомое устройство или ведущее устройство выполняет попытку параметризовать ведомое устройство заново.
	2	Активно (установлено соединение): Ведомые устройства выполняют обмен входными/выходными данными с ЦПУ.

Элемент	Описание
Counter Slave Alarm [DWORD]	Количество отправленных аварийных сигналов. Считает с момента сброса счетчика.
Counter Standard Diagnostic [DWORD]	Количество отправленных диагностических сообщений. Считает с момента сброса счетчика.

Таблица 46: Системные переменные ведомого устройства PROFIBUS DP

5.6.3 Свойства

Функция меню **Properties** из контекстного меню ведомого устройства PROFIBUS DP открывает диалоговое окно **Properties**. Диалоговое окно содержит следующие вкладки:

Вкладка Параметры

Элемент	Описание
Name	Имя ведомого устройства
Address	Адрес ведомого устройства Диапазон значений: 0...125 Значение по умолчанию: 0
Active	Состояние ведомого устройства Только активное ведомое устройство может соединяться с ведущим устройством PROFIBUS DP. Значение по умолчанию: Activated
DPV0 Sync active	Режим <i>Sync Mode</i> позволяет осуществлять синхронную активацию выходов нескольких ведомых устройств DP-V0. Внимание: У ведомых устройств DP-V2, работающих в режиме <i>Isochron Mode Sync</i> , это поле должно быть деактивировано. Значение по умолчанию: Deactivated
DPV0 Freeze active	Режим <i>Freeze Mode</i> позволяет осуществлять синхронный прием входных данных нескольких ведомых устройств DP-V0. Внимание: У ведомых устройств DP-V2, работающих в режиме <i>Isochron Mode Freeze</i> , это поле должно быть деактивировано. Значение по умолчанию: Deactivated
Watchdog Active	Если выбрано, то ведомое устройство распознает сбой ведущего устройства и переходит в безопасное состояние. Значение по умолчанию: Deactivated
Watchdog time [ms]	Кнопка-флажок «Сторожевое устройство активно» должна быть активирована. Если в течение данного промежутка времени не приходит обмен данными между ведущим и ведомым устройством, то ведомое устройство отключается и сбрасывает все выходные данные DP на предустановленное значение по умолчанию. 0 = деактивировано Ориентировочное значение: Время сторожевого устройства ведомого устройства $> 6 * T_{tr}$ Диапазон значений: 0...65535 Значение по умолчанию: 0

Элемент	Описание
On failure send last data	FALSE: В случае ошибки соединение прерывается и устанавливается снова. TRUE: В случае ошибки передает данные дальше даже без подтверждения ведомого устройства. Значение по умолчанию: Deactivated
Auto clear on failure	Если <i>Auto clear on failure</i> в ведущем устройстве и у этого ведомого устройства установлен на TRUE, то ведущее устройство при сбое работы данного ведомого устройства переключает PROFIBUS DP полностью в безопасное состояние. Значение по умолчанию: Activated

Таблица 47: Вкладка Параметры ведомого устройства PROFIBUS DP

Вкладка Группы

В данной вкладке ведомые устройства могут быть организованы в различных группах. Команды глобального управления *Sync* и *Freeze* могут в этом случае целенаправленно направляться в одну или несколько групп.

Элемент	Описание	
Member of Group 1	Элемент в группе 1	Значение по умолчанию: Deactivated
Member of Group 2	Элемент в группе 2	
Member of Group 3	Элемент в группе 3	
Member of Group 4	Элемент в группе 4	
Member of Group 5	Элемент в группе 5	
Member of Group 6	Элемент в группе 6	
Member of Group 7	Элемент в группе 7	
Member of Group 8	Элемент в группе 8	

Таблица 48: Вкладка Группы в диалоговом окне свойств ведомого устройства PROFIBUS DP

Вкладка DP-V1

В данной вкладке находятся параметры, которые определены только, начиная с DP-V1. У ведомых устройств DP-V0 здесь не может быть ничего выбрано. Параметры, поддерживаемые ведомым устройством, отображаются в столбце «Поддерживается».

Элемент	Описание
DP V1	Если режим DP-V1 не активирован, то другие свойства DP-V1 тоже не могут использоваться. Ведомое устройство ведет себя в этом случае, как ведомое устройство DP-V0. При необходимости следует изменить также конфигурационные данные (см. руководство по ведомому устройству). Значение по умолчанию: Deactivated
Failsafe	Если данный режим активирован, то ведущее устройство в состоянии CLEAR не передает нули в качестве выходных данных на ведомое устройство, а передает пустой пакет данных (Failsafe packet). Ведомое устройство распознает в этом случае, что оно должно отправить на выходы безопасные выходные данные (которые не обязательно являются нулями). Значение по умолчанию: Deactivated

Элемент	Описание
Isochron Mode	Данная функция позволяет осуществлять регулирование с тактовой синхронизацией в ведущем и ведомом устройстве, независимо от нагрузки шины. Цикл шины синхронизируется с тактовым отклонением < 1 мс. Благодаря этому могут быть реализованы высокоточные процессы позиционирования. Значение по умолчанию: Deactivated
Publisher Active	Данная функция требуется для сторонней связи ведомого устройства. Это позволяет осуществлять прямую и ускоренную связь между ведомыми устройствами посредством широковещательной рассылки, минуя ведущее устройство. Значение по умолчанию: Deactivated
Prm Block Struct. Supp.	Ведомое устройство поддерживает структурированные данные параметрирования (только чтение). Значение по умолчанию: Deactivated
Check Cfg Mode	Сокращенный контроль конфигурации, если активирован режим проверки конфигурации, то ведомое устройство может эксплуатироваться без полной конфигурации. Для ввода в эксплуатацию данное поле необходимо деактивировать. Значение по умолчанию: Deactivated

Таблица 49: Вкладка DP-V1 в диалоговом окне свойств ведомого устройства PROFIBUS DP

Вкладка Аварийные сигналы

На данной странице можно активировать аварийные сигналы. Это возможно только у ведомых устройств DP-V1, если активировано DP-V1 и ведомое устройство поддерживает аварийные сигналы. Какие аварийные сигналы поддерживаются, можно распознать по галочке, установленной в столбце **Supp.** Если аварийный сигнал предписан, это отображено в столбце **Required.**

Элемент	Описание	
Update Alarm	Аварийный сигнал, если параметр модуля изменен.	Значение по умолчанию: Deactivated
Status Alarm	Аварийный сигнал, если состояние модуля изменено.	
Vendor Alarm	Аварийный сигнал изготовителя.	
Diagnostic Alarm	Аварийный сигнал при определенных событиях, например, коротком замыкании, перегреве и т.д., на модуле.	
Process Alarm	Аварийный сигнал при важных событиях в процессе.	
Pull & Plug Alarm	Аварийный сигнал при снятии и установке модуля.	

Таблица 50: Вкладка Аварийные сигналы в диалоговом окне свойств ведомого устройства PROFIBUS DP

Вкладка Данные

В данной вкладке находится информация о поддерживаемой длине данных, а также о данных пользователя (расширенные конфигурационные данные).

Элемент	Описание
Max. Input Len [Byte]	Максимальная длина входных данных.
Max. Output Len [Byte]	Максимальная длина выходных данных.
Max. Data Len [Byte]	Максимальная общая длина входных и выходных данных.
User Prmdata Len [Byte]	Длина данных пользователя.
User Prmdata	Конфигурационные данные. Не рекомендуется выполнять здесь редактирование. Вместо этого используйте диалоговое окно <i>User Parameter</i> , см. главу 0.
Max. Diag. Data Len [Byte]	Максимальная длина диагностических данных, передаваемых ведомым устройством.

Таблица 51: Вкладка Данные в диалоговом окне свойств ведомого устройства PROFIBUS DP

Вкладка Модель

На этой странице находится различная информация:

Элемент	Описание
Model	Обозначение изготовителя ведомого устройства PROFIBUS DP
Manufacturer	Изготовитель полевого устройства
Ident Number	Маркировка ведомого устройства «PROFIBUS Nutzerorganisation» (PNO)
Revision	Версия ведомого устройства PROFIBUS DP
Hardware release	Версия аппаратного обеспечения ведомого устройства PROFIBUS DP
Software release	Версия программного обеспечения ведомого устройства PROFIBUS DP
GSD file name	Имя файла GSD
Info Text	Дополнительная информация по ведомому устройству PROFIBUS DP

Таблица 52: Вкладка Модель в диалоговом окне свойств ведомого устройства PROFIBUS DP

Вкладка Свойства

Элемент	Описание
Modular Station	TRUE: Модульная станция FALSE: Компактная станция
First slot number	Нумерация модулей (слотов) должна начинаться с данного номера и осуществляться по возрастанию.
Max Modules	Максимальное количество модулей, которые можно разместить на модульной станции.
Support for 'Set Slave Add'	Ведомое устройство поддерживает динамическое присвоение адреса.
Min. Slave Interval [ms]	Минимальное время, которое должно пройти между двумя циклическими обращениями ведомого устройства.
Diag. Update	Число циклов опроса до того момента, пока диагностика ведомого устройства не отобразит текущее состояние.
Support for WDBase1ms	Ведомое устройство поддерживает 1 мс как временную базу для сторожевого устройства.
Support for DPV0 Sync	Ведомое устройство поддерживает DP-V0 Sync.
Support for DPV0 Freeze	Ведомое устройство поддерживает DP-V0 Freeze.
DPV1 Data Types	Ведомое устройство поддерживает типы данных DP-V1.
Extra Alarm SAP	Ведомое устройство поддерживает SAP 50 для подтверждения аварийного сигнала.
Alarm Seq. Mode Count	Отображает, сколько активных аварийных сигналов ведомое устройство может обработать одновременно. Нуль обозначает аварийный сигнал любого типа.

Таблица 53: Вкладка Свойства в диалоговом окне свойств ведомого устройства PROFIBUS DP

Вкладка Скорость передачи

В данной вкладке находятся значения скорости передачи, которые поддерживает ведомое устройство, а также соответствующее значение *MaxTsdr*.

MaxTsdr – это время, в течение которого ведомое устройство должно ответить на запрос ведущего устройства. Диапазон значений зависит от ведомого устройства и скорости передачи и лежит между 15 и 800 Тбит.

Элемент	Описание
9,6k	MaxTsdr = 60
19,2k	MaxTsdr = 60
31,25k	Не поддерживается
45,45k	MaxTsdr = 60
93,75k	MaxTsdr = 60
187,5k	MaxTsdr = 60
500k	MaxTsdr = 70
1,5M	MaxTsdr = 75
3M	MaxTsdr = 90
6M	MaxTsdr = 100
12M	MaxTsdr = 120

Таблица 54: Вкладка Скорость передачи в диалоговом окне свойств ведомого устройства PROFIBUS DP

Вкладка Ациклическая передача

В данной вкладке находятся некоторые параметры для ациклической передачи данных:

Элемент	Описание
Support for C1 Read/Write	Ведомое устройство поддерживает ациклическую передачу данных.
C1 Read/Write required	Ведомое устройство требует ациклической передачи данных.
C1 Max Data Len[Byte]	Максимальная длина ациклического пакета данных.
C1 Response Timeout [ms]	Время ожидания для ациклической передачи данных.

Таблица 55: Вкладка Ациклическая передача в диалоговом окне свойств ведомого устройства PROFIBUS DP

5.7 Считывание файла GSD

Файл GSD содержит данные для параметрирования ведомого устройства PROFIBUS DP.

Чтобы создать файл GSD для нового ведомого устройства PROFIBUS DP, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, PROFIBUS Slave**.
2. В контекстном меню ведущего устройства PROFIBUS DP выбрать **Read GSD File** и выбрать относящийся к ведомому устройству PROFIBUS файл GSD (напр., *hax100ea.gsd*).

i

Файлы GSD для систем управления HIMax Вы найдете на интернет-сайте компании HIMA www.hima.com.

Ответственность за правильность файла GSD несет изготовитель соответствующего полевого устройства.

Файл GSD *hax100ea.gsd* ведомого устройства HIMax PROFIBUS DP содержит следующие модули:

Модули ввода ведущего устройства PROFIBUS DP	Тип	Количество
DP Input/ELOP Export	Byte	1
DP Input/ELOP Export	Bytes	2
DP Input/ELOP Export	Bytes	4
DP Input/ELOP Export	Bytes	8
DP Input/ELOP Export	Bytes	16
DP Input/ELOP Export	Word	1
DP Input/ELOP Export	Words	2
DP Input/ELOP Export	Words	4
DP Input/ELOP Export	Words	8
DP Input/ELOP Export	Words	16
Модули вывода ведущего устройства PROFIBUS DP	Тип	Количество
DP Output/ELOP Import	Byte	1
DP Output/ELOP Import	Bytes	2
DP Output/ELOP Import	Bytes	4
DP Output/ELOP Import	Bytes	8
DP Output/ELOP Import	Bytes	16
DP Output/ELOP Import	Word	1
DP Output/ELOP Import	Words	2
DP Output/ELOP Import	Words	4
DP Output/ELOP Import	Words	8
DP Output/ELOP Import	Words	16

Таблица 56: Файл GSD ведомого устройства PROFIBUS DP от HIMax

5.8 Обработка параметров пользователя

В поле данных пользователя определяется начальный адрес и число переменных для блоков.

Кроме того, для ведущего устройства PROFIBUS DP должно быть конфигурировано число фактически передаваемых байтов. Это можно осуществить путем выбора модулей PROFIBUS DP из файла GSD ведомого устройства PROFIBUS DP (см. также главу 5.2.2).

Чтобы открыть диалоговое окно для обработки параметров пользователя, необходимо:

1. Открыть в структуре **Configuration/Resource/Protocols PROFIBUS DP Master**.
2. Щелкнуть правой кнопкой мыши на **PROFIBUS Slave** и выбрать **Properties**.
3. Щелкнуть на вкладку **Data** и выбрать кнопки ... рядом с данными пользователя.

i

Структура диалогового окна **Edit User Parameters** определена в файле GSD ведомого устройства.

Принципиальная структура поля данных пользователя размером 32 байта:

Поле данных пользователя размером 32 байта создается следующим образом:

32 байта сгруппированы в восьми блоках, по четыре байта в каждом.

Блоки 1...4 определяют, какие и сколько переменных ведущее устройство PROFIBUS DP принимает от ведомого устройства PROFIBUS DP.

Блоки 5...8 определяют, какие и сколько переменных ведущее устройство PROFIBUS DP передает на ведомое устройство PROFIBUS DP.

Первые два байта каждого блока указывают начальный адрес для первой считываемой или записываемой переменной блока.

Последние два байта каждого блока указывают число переменных, которые должны быть приняты или переданы.

Конфигурирование данных пользователя в различных блоках:

В обычных случаях не требуется распределять переменные (данные пользователя) по различным блокам. Будет достаточно определить только первый блок входных и выходных переменных, считать и записать данные *en bloc*.

В случаях применения, в которых все же требуется считывать и записывать только выбранные переменные, можно определить по четыре блока переменных для входных и выходных переменных.

Пример

Ведущее устройство PROFIBUS DP передает и принимает следующие переменные от ведомого устройства PROFIBUS DP:

- 1 блок: **4** входных переменных, начиная с начального адреса **0**.
- 2 блок: **6** входных переменных, начиная с начального адреса **50**.
- 4 блок: **9** входных переменных, начиная с начального адреса **100**.
- 5 блок: **2** выходных переменных, начиная с начального адреса **10**.

Конфигурирование данных пользователя в ведущем устройстве PROFIBUS DP:

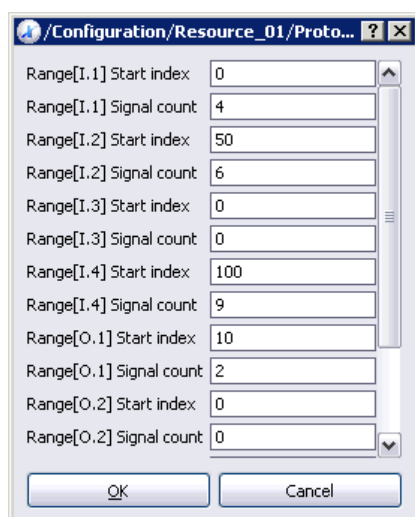
Ведущ. уст-во Импорт/Ведом. уст-во Экспорт	Начальный адрес	Число переменных
1 блок (байт 0...3)	0,0	0,4
2 блок (байт 4...7)	0,50	0,6
3 блок (байт 8...11)	0,0	0,0
4 блок (байт 12...15)	0,100	0,9

Таблица 57: Пример: Блоки 1...4 поля данных пользователя

Ведущ. уст-во Экспорт/Ведом. устр-во Импорт	Начальный адрес	Число переменных
5 блок (байт 16...19)	0,10	0,2
6 блок (байт 20...23)	0,0	0,0
7 блок (байт 24...27)	0,0	0,0
8 блок (байт 28...31)	0,0	0,0

Таблица 58: Пример: Блоки 1...4 поля данных пользователя

Диалоговое окно *Edit User Parameters* ведомого устройства HIMatrix или HIMax PROFIBUS DP.

Рисунок 32: Диалоговое окно *Edit User Parameters*

5.9 Функциональные блоки PROFIBUS

С помощью данных функциональных блоков Вы можете оптимально отрегулировать ведущее устройство HIMA PROFIBUS DP и присвоенное ему ведомое устройство PROFIBUS DP в соответствии с требованиями вашего проекта.

Функциональные блоки параметрируйте в прикладной программе, чтобы функции ведущего и ведомого устройства (аварийные сигналы, диагностические данные, состояния) можно было устанавливать и считывать в прикладной программе.

i

Функциональные блоки требуются только для специальных приложений. Для обычного циклического обмена данными между ведущим и ведомым устройством данные функциональные блоки не требуются!

Принципиальная конфигурация функциональных блоков PROFIBUS DP, см. главу 10.

Доступны следующие функциональные блоки:

Функциональный блок	Описание функции	Подходит, начиная со степени нагрузки
MSTAT 0	Контроль состояния ведущего устройства с помощью прикладной программы	DP-V0
RALRM 0	Считывание аварийных сообщений ведомого устройства	DP-V1
RDIAG 0	Считывание диагностических сообщений ведомого устройства	DP-V0
RDREC 0	Считывание ациклических наборов данных ведомого устройства	DP-V1
SLACT 0	Контроль состояния ведомого устройства с помощью прикладной программы	DP-V0
WRREC 0	Запись ациклических наборов данных ведомого устройства	DP-V1

Таблица 59: Обзор функциональных блоков PROFIBUS DP

i

Ведущие устройства HIMA PROFIBUS DP работают со степенью нагрузки DP-V1. Ведомые устройства HIMA PROFIBUS DP работают со степенью нагрузки DP-V0. Учтите, что не все функциональные блоки ведущего устройства HIMA PROFIBUS DP могут контролировать ведомое устройство HIMA PROFIBUS DP.

5.9.1 Функциональный блок MSTAT

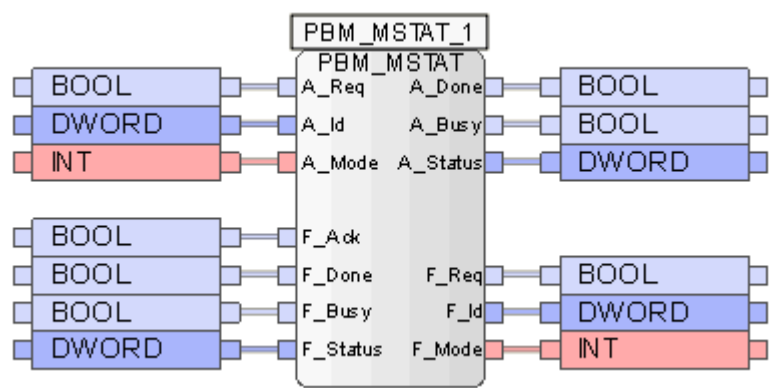


Рисунок 33: Функциональный блок MSTAT

С помощью функционального блока **MSTAT** (с DP-V0) ведущее устройство PROFIBUS DP может контролироваться прикладной программой. Благодаря этому существует возможность переключать ведущее устройство PROFIBUS DP с помощью механического переключателя на физическом входе или с помощью таймера в одно из следующих состояний:

- 0: OFFLINE
- 1: STOP
- 2: CLEAR
- 3: OPERATE

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом A:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	Передний фронт запускает блок	BOOL
A_ID	ID ведущего устройства (не используется)	DWORD
A_Mode	Ведущее устройство PROFIBUS DP может быть установлено на следующие состояния: 0: OFFLINE 1: STOP 2: CLEAR 3: OPERATE	INT

Таблица 60: A-входы Функциональный блок MSTAT

A-выходы	Описание	Тип
A_Done	TRUE: Ведущее устройство PROFIBUS DP было установлено на определенное на входе A_Mode состояние.	BOOL
A_Busy	TRUE: Установка ведущего устройства PROFIBUS DP еще не завершена.	BOOL
A_Status	Состояние или код ошибки (см. главу 5.11)	DWORD

Таблица 61: A-выходы Функциональный блок MSTAT

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение с функциональным блоком MSTAT в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока MSTAT в структуре (в папке Функциональные блоки) с функциональным блоком MSTAT (в прикладной программе) осуществляется посредством общих переменных. Эти переменные Вы должны заранее определить в редакторе переменных.

Соедините *F-Inputs* функционального блока **MSTAT** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **MSTAT** в структуре.

F-входы	Тип
F_ACK	BOOL
F_DONE	BOOL
F_BUSY	BOOL
F_STATUS	DWORD

Таблица 62: F-входы Функциональный блок MSTAT

Соедините *F-Outputs* функционального блока **MSTAT** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **MSTAT** в структуре.

F-выходы	Тип
F_REQ	BOOL
F_ID	DWORD
F_MODE	INT

Таблица 63: F-выходы Функциональный блок MSTAT

Чтобы создать функциональный блок MSTAT в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New**.
2. Выбрать функциональный блок **MSTAT**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **MSTAT** и выбрать **Edit**.
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **MSTAT** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **MSTAT** в прикладной программе.

Входы	Тип
M_ID	DWORD
MODE	INT
REQ	BOOL

Таблица 64: Входные системные переменные

Соедините следующие выходы функционального блока **MSTAT** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **MSTAT** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Таблица 65: Выходные системные переменные

Для использования функционального блока MSTAT необходимо:

1. В прикладной программе установить вход *A_Mode* на необходимое состояние. Если *A_Mode* не устанавливается, то после 2 шага выдается код ошибки на выходе *A_Status*, и ведущее устройство PROFIBUS DP не устанавливается.
2. В прикладной программе установить вход *A_Req* на TRUE.

i

Функциональный блок реагирует на передний фронт на *A_Req*.

-
- ☒ Выход *A_Busy* имеет статус TRUE, пока команда MSTAT не будет обработана. Затем выход *A_Busy* меняется на FALSE и выход *A_Done* на TRUE.

i

Если невозможно установить заданный режим, на выходе *A_Status* выдается код ошибки.

Текущий режим ведущего устройства можно определить по переменной состояния ведущего устройства (см. главу 5.10).

5.9.2 Функциональный блок RALRM

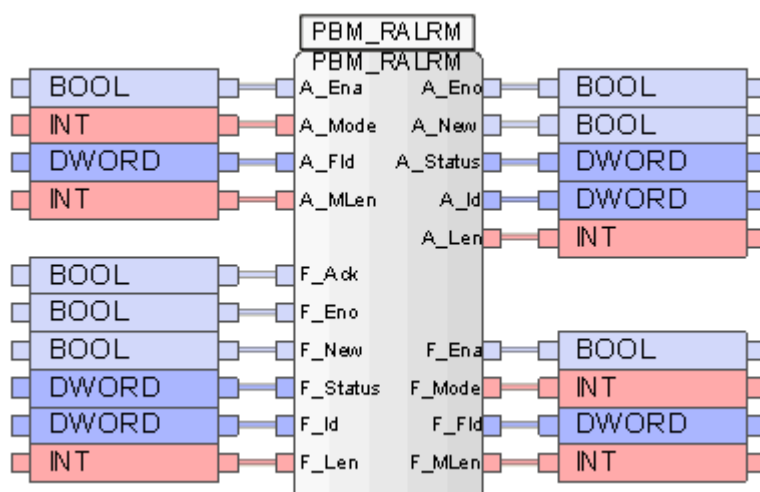


Рисунок 34: Функциональный блок RALRM

Функциональный блок **RALRM** (начиная с DP-V1) служит для анализа аварийных сигналов.

Аварийные сигналы представляют собой особую форму диагностических сообщений, которые обрабатываются в первую очередь. Аварийные сигналы сообщают приложению важные события, которые требуют реакции со стороны приложения (напр., WRREC). Она зависит от изготовителя и описана в руководстве по ведомому устройству PROFIBUS DP.

Функциональный блок **RALRM** в активном состоянии постоянно ожидает приема аварийных сообщений ведомых устройств. Если принимается аварийный сигнал, то выход **A_NEW** минимум на один цикл переключается на TRUE и считывает аварийные данные посредством аварийной телеграммы. Перед следующим аварийным сигналом **A_NEW** минимум на один цикл переходит на FALSE. Все аварийные сигналы подтверждаются неявно. Ни один аварийный сигнал не исчезает.

При использовании нескольких функциональных блоков **RALRM** прикладную программу необходимо конфигурировать таким образом, чтобы всегда был активен только один функциональный блок **RALRM**.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом A:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Ena	С помощью TRUE функциональный блок активируется	BOOL
A_Mode	Не используется	INT
A_FID	Не используется	DWORD
A_MLen	Максимальная ожидаемая длина принимаемых аварийных данных в байтах	INT

Таблица 66: A-входы Функциональный блок RALRM

A-выходы	Описание	Тип
A_Eno	TRUE: Функциональный блок активен FALSE: Функциональный блок не активен	BOOL
A_New	TRUE: Принят новый аварийный сигнал FALSE: Нет новых аварийных сигналов	BOOL
A_Status	Состояние или код ошибки (см. главу 5.11)	DWORD
A_ID	Идентификационный номер ведомого устройства, от которого поступил аварийный сигнал	DWORD
A_Len	Длина принятых аварийных данных в байтах	INT

Таблица 67: A-выходы Функциональный блок RALRM

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение с функциональным блоком RALRM в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **RALRM** в структуре (в папке Функциональные блоки) с функциональным блоком **RALRM** (в прикладной программе) осуществляется посредством общих переменных. Эти переменные Вы должны заранее определить в редакторе переменных.

Соедините *F-Inputs* функционального блока **RALRM** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **RALRM** в структуре.

F-входы	Тип
F_ACK	BOOL
F_ENO	BOOL
F_NEW	BOOL
F_STATUS	DWORD
F_ID	DWORD
F_LEN	INT

Таблица 68: F-входы Функциональный блок RALRM

Соедините *F-Outputs* функционального блока **RALRM** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **RALRM** в структуре.

F-выходы	Тип
F_Ena	BOOL
F_MODE	INT
F_FID	DWORD
F_MLEN	INT

Таблица 69: F-выходы Функциональный блок RALRM

Чтобы создать функциональный блок RALRM в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New**.
2. Выбрать функциональный блок **RALRM**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **RALRM** и выбрать **Edit**.
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **RALRM** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **RALRM** в прикладной программе.

Входы	Тип
EN	BOOL
F_ID	DWORD
MLEN	INT
MODE	INT

Таблица 70: Входные системные переменные

Соедините следующие выходы функционального блока **RALRM** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **RALRM** в прикладной программе.

Выходы	Тип
ACK	BOOL
ENO	BOOL
ID	DWORD
LEN	INT
NEW	BOOL
STATUS	DWORD

Таблица 71: Выходные системные переменные

Во вкладке Process Variables функционального блока **RALRM** в структуре необходимо определить переменные, структура которых подходит к аварийным данным. Если переменные не определены, то аварийные данные можно запросить, но не считать.

Аварийное сообщение содержит минимум четыре байта. Первые четыре байта содержат стандартные аварийные данные.

Чтобы упростить анализ стандартных аварийных сигналов, фирма HIMA предлагает вспомогательный функциональный блок **ALARM** (см. главу 5.10). Если Вы хотите использовать данный блок, объедините первые четыре байта в переменной типа DWORD и передайте данную переменную на вход *IN* вспомогательного функционального блока **ALARM**.

i

Если аварийная телеграмма содержит больше байтов, чем определено во вкладке «Данные», то принимается только определенное число байтов. Остальные отрезаются.

Аварийные данные	Описание
Байт 0	Длина аварийного сообщения в байтах (4...126)
Байт 1	Обозначение для типа аварийного сигнала 1: аварийный сигнал: диагностика 2: аварийный сигнал: процесс 3: аварийный сигнал: извлечение 4: аварийный сигнал: вставка 5: аварийный сигнал: статус 6: аварийный сигнал: обновление 31: неудачное расширение ведущего или ведомого устройства 32...126: определяется изготовителем: Значение должно содержаться в описании устройства, выпущенном изготовителем.
Байт 2	Номер слота компонента, от которого поступил аварийный сигнал
Байт 3	0: информация отсутствует 1: входящий аварийный сигнал, слот поврежден 2: исходящий аварийный сигнал, слот больше не поврежден 3: исходящий аварийный сигнал, слот все еще поврежден
Байт 4 до 126	Значение должно содержаться в описании устройства, выпущенном изготовителем

Таблица 72: Аварийные данные

i

Структура стандартных аварийных сигналов (байт 0...3) носит нормированный характер и одинакова для всех изготовителей. Информацию по используемым изготовителем байтам 4...126 Вы найдете в руководстве по ведомому устройству PROFIBUS DP.

Помните, что устройства, отвечающие стандарту DP-V0, не поддерживают аварийные телеграммы.

Для использования функционального блока RALRM необходимо:

1. Определить в прикладной программе на входе *A_Mlen* число максимально ожидаемых аварийных данных в байтах. Во время работы *A_Mlen* не может быть изменено.
2. Установить в прикладной программе вход *A_Ena* на TRUE.

i

В отличие от других блоков функциональный блок **RALRM** активен только тогда, когда вход *A_Ena* установлен на TRUE.

Если блок успешно запущен, то выход *A_Eno* переходит на TRUE. Если блок невозможно запустить, то на выходе *A_Status* выдается код ошибки.

Если возникает новый аварийный сигнал, то выход *A_New* минимум на один цикл переходит на TRUE. В это время выходы содержат доступные для анализа аварийные данные ведомого устройства, от которого поступил аварийный сигнал.

Затем выход *A_New* снова минимум на один цикл переходит на FALSE.

Выходы *A_ID* и *A_Len* сбрасываются на нуль, прежде чем можно будет принять и обработать следующее аварийное сообщение.

5.9.3 Функциональный блок RDIAG

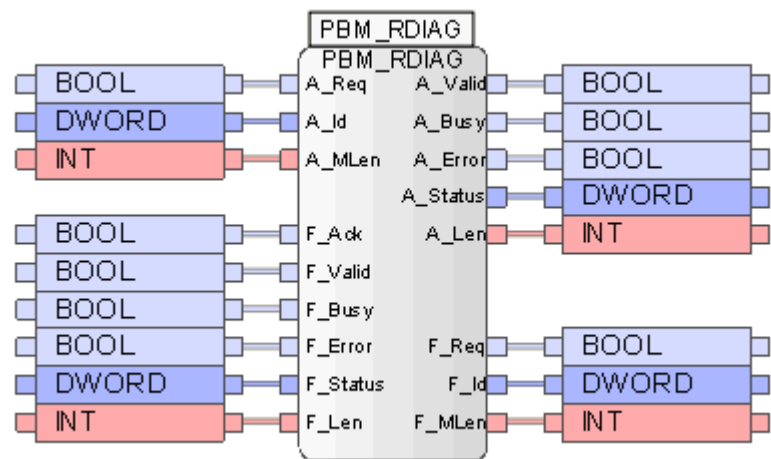


Рисунок 35: Функциональный блок RDIAG

Функциональный блок **RDIAG** (с DP-V0) служит для считывания текущего диагностического сообщения (6 байтов...240 байтов) ведомого устройства.

В ведущем устройстве HIMA PROFIBUS DP одновременно могут быть активными несколько блоков **RDIAG**.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом А:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «А» означает «Application» (приложение).

А-входы	Описание	Тип
A_Req	Передний фронт запускает запрос диагностического сообщения	BOOL
A_ID	Идентификационный номер ведомого устройства (см. главу 5.10)	DWORD
A_MLen	Максимальная ожидаемая длина считываемого диагностического сообщения в байтах	INT

Таблица 73: А-входы Функциональный блок RALRM

А-выходы	Описание	Тип
A_Valid	Принято и является действительным новое диагностическое сообщение	BOOL
A_Busy	TRUE: Считывание еще не завершено	BOOL
A_Error	TRUE: При считывании возникла ошибка	BOOL
A_Status	Состояние или код ошибки (см. главу 5.11)	DWORD
A_Len	Длина считанных диагностических данных в байтах	INT

Таблица 74: А-выходы Функциональный блок RALRM

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение с функциональным блоком **RDIAG** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **RDIAG** в структуре (в папке Функциональные блоки) с функциональным блоком **RDIAG** (в прикладной программе) осуществляется посредством общих переменных. Эти переменные Вы должны заранее определить в редакторе переменных.

Соедините *F-Inputs* функционального блока **RDIAG** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **RDIAG** в структуре.

F-входы	Тип
F_ACK	BOOL
F_VALID	BOOL
F_BUSY	BOOL
F_ERROR	BOOL
F_Status	DWORD
F_LEN	INT

Таблица 75: F-входы Функциональный блок RDIAG

Соедините *F-Outputs* функционального блока **RDIAG** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **RDIAG** в структуре.

F-выходы	Тип
F_Req	BOOL
F_Id	DWORD
F_MLen	INT

Таблица 76: F-выходы Функциональный блок RDIAG

Чтобы создать функциональный блок RDIAG в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New**.
2. Выбрать функциональный блок **RDIAG**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **RDIAG** и выбрать **Edit**.
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **RDIAG** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **RDIAG** в прикладной программе.

Входы	Тип
ID	DWORD
MLEN	INT
REQ	BOOL

Таблица 77: Входные системные переменные

Соедините следующие выходы функционального блока **RDIAG** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **RDIAG** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
Status	DWORD
VALID	BOOL

Таблица 78: Выходные системные переменные

Диагностические данные

Во вкладке **Data** необходимо определить переменные, структура которых должна подходить к диагностическим данным. Диагностическое сообщение содержит минимум шесть байтов и максимум 240 байтов. Первые четыре байта диагностического сообщения содержат стандартную диагностику.

Чтобы упростить анализ стандартных диагностических данных, фирма HIMA предлагает вспомогательный функциональный блок **STDDIAG** (см. главу 5.10). Если Вы хотите использовать данный блок, объедините первые четыре байта в переменной типа DWORD и передайте данную переменную на вход *IN* вспомогательного функционального блока **STDDIAG**.

i

Если диагностическая телеграмма содержит больше байтов, чем определено во вкладке «Данные», то принимается только определенное число байтов. Остальные отрезаются.

Диагностические данные	Описание
Байт 0	Байты 0...3 содержат стандартную диагностику. Стандартная диагностика может быть декодирована как переменная типа DWORD с помощью вспомогательного функционального блока STDDIAG .
Байт 1	
Байт 2	
Байт 3	Адрес шины ведущего устройства, которому присвоено ведомое устройство.
Байт 4	Старший байт (обозначение изготовителя).
Байт 5	Младший байт (обозначение изготовителя).
Байт 6...240	Значение должно содержаться в описании устройства, выпущенном изготовителем.

Таблица 79: Диагностические данные

i

Ведомые устройства HIMA передают диагностическую телеграмму длиной шесть байтов. Значение байта является стандартным.

Для ведомых устройств других изготовителей по функциям совпадают только первые шесть байтов.

Более подробная информация по диагностической телеграмме содержится в описании ведомого устройства от изготовителя.

Для использования функционального блока **RDIAG** необходимо:

1. Установить в прикладной программе адрес ведомого устройства на вход *A_ID*.
2. Определить в прикладной программе на входе *A_Mlen* число максимально ожидаемых диагностических данных в байтах.
3. В прикладной программе установить вход *A_Req* на TRUE.

i

Функциональный блок реагирует на передний фронт на *A_Req*.

Выход *A_Busy* является TRUE, пока диагностический запрос не будет обработан. Затем выход *A_Busy* переходит на FALSE и выходы *A_Valid* или *A_Error* на TRUE.

Если диагностическая телеграмма действительна, то выход *A_Valid* переходит на TRUE. Диагностические данные можно анализировать посредством определенных во вкладке «Данные» переменных. Выход *A_Len* содержит число диагностических данных в байтах, которые были фактически считаны.

Если диагностическая телеграмма не была считана успешно, то выход *A_Error* переходит на TRUE и на выходе *A_Status* выдается код ошибки.

5.9.4 Функциональный блок RDREC

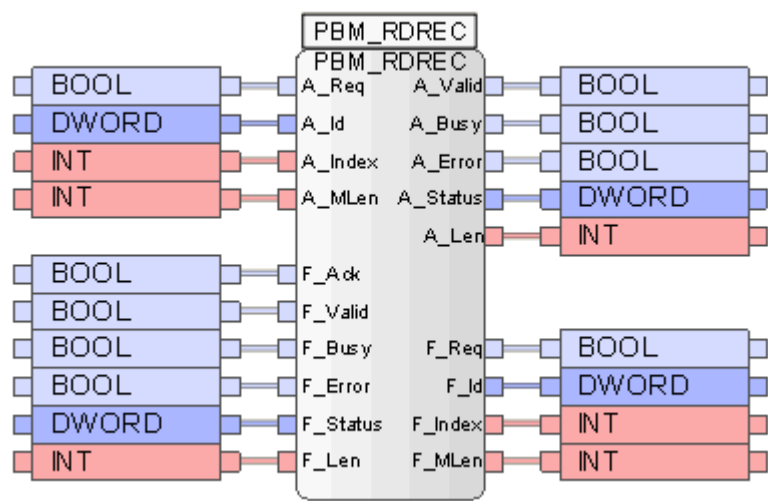


Рисунок 36: Функциональный блок RDREC

Функциональный блок **RDREC** служит для ациклического считывания набора данных с ведомого устройства, адрес которого указан на *A_Index*. Информацию о том, какие данные могут быть считаны, Вы найдете в руководстве по ведомому устройству.

Данная функция определена, начиная с DP-V1, и доступна в качестве опции.

В ведущем устройстве HIMA PROFIBUS DP одновременно может быть активировано до 32 **RDREC** и/или **WRREC** блоков.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом A:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	Передний фронт запускает запрос для считывания.	BOOL
A_Id	Идентификационный номер ведомого устройства, см. главу 5.10.	DWORD
A_Index	Номер считываемого набора данных. Значение должно содержаться в описании устройства, выпущенном изготовителем.	INT
A_MLen	Максимальная длина считываемых данных в байтах.	INT

Таблица 80: A-входы Функциональный блок RDREC

А-выходы	Описание	Тип
A_Valid	Новый набор данных принят и является действительным.	BOOL
A_Busy	TRUE: Процесс считывания еще не завершен.	BOOL
A_Error	TRUE: Возникла ошибка FALSE: Нет ошибки	BOOL
A_Status	Состояние или код ошибки, см. главу 5.11	DWORD
A_Len	Длина считанного набора данных в байтах.	INT

Таблица 81: А-выходы Функциональный блок RDREC

Входы и выходы функционального блока с префиксом F

Данные входы и выходы функционального блока создают соединение с функциональным блоком **RDREC** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **RDREC** в структуре (в папке Функциональные блоки) с функциональным блоком RDREC (в прикладной программе) осуществляется посредством общих переменных. Эти переменные Вы должны заранее определить в редакторе переменных.

Соедините *F-Inputs* функционального блока **RDREC** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **RDREC** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Valid	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Таблица 82: F-входы Функциональный блок RDREC

Соедините *F-Outputs* функционального блока **RDREC** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **RDREC** в структуре.

F-выходы	Тип
F_Req	BOOL
F_Id	DWORD
F_Index	INT
F_Mlen	INT

Таблица 83: F-выходы Функциональный блок RDREC

Чтобы создать функциональный блок RDREC в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New**.
2. Выбрать функциональный блок **RDREC**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **RDREC** и выбрать **Edit**.
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **RDREC** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **RDREC** в прикладной программе.

Входы	Тип
ID	DWORD
INDEX	INT
MLEN	INT
REQ	BOOL

Таблица 84: Входные системные переменные

Соедините следующие выходы функционального блока **RDREC** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **RDREC** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Таблица 85: Выходные системные переменные

Данные	Описание
Переменные не заданы	Во вкладке <i>Process Variables</i> может быть определена любая структура данных, которая должна все же подходить к структуре набора данных. Структура набора данных содержится в руководстве изготовителя по ведомому устройству.

Таблица 86: Данные

Для использования функционального блока **RDREC** необходимо:

1. Установить в прикладной программе адрес ведомого устройства на вход *A_ID*.
2. В прикладной программе установить специальный для ведомого устройства индекс для набора данных (руководство изготовителя) на входе *A_Index*.
3. В прикладной программе установить длину считываемого набора данных на входе *A_Len*.
4. В прикладной программе установить вход *A_Req* на TRUE.



Функциональный блок реагирует на передний фронт на *A_Req*.

Выход *A_Busy* является TRUE, пока не будет обработан запрос набора данных. Затем выход *A_Busy* переходит на FALSE и выходы *A_Valid* или *A_Error* на TRUE.

Если набор данных действительный, то выход *A_Valid* переходит на TRUE. Набор данных можно анализировать посредством определенных во вкладке «Данные» переменных. Выход *A_Len* содержит фактическую длину считанного набора данных.

Если набор данных не был считан успешно, то выход *A_Error* переходит на TRUE и на выходе *A_Status* выдается код ошибки.

5.9.5 Функциональный блок SLACT

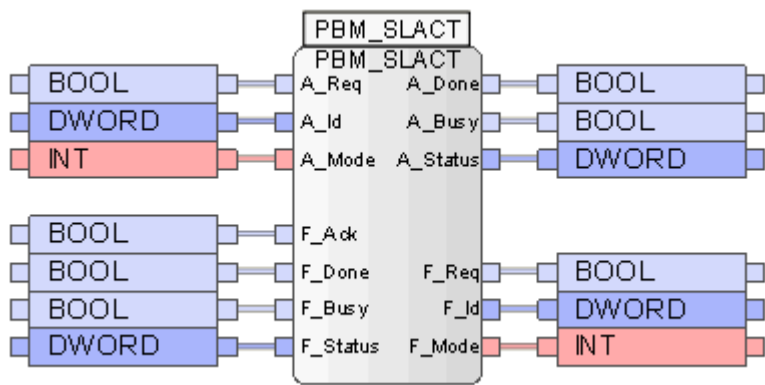


Рисунок 37: Функциональный блок SLACT

Функциональный блок **SLACT** (с DP-V0) служит для активации и деактивации ведомого устройства из прикладной программы ведущего устройства PROFIBUS DP. Благодаря этому существует возможность переключать ведомое устройство с помощью механического переключателя на физическом входе ведущего устройства PROFIBUS DP или с помощью таймера в одно из следующих состояний.

- ≠ 0: активно
- = 0: не активно

При использовании нескольких функциональных блоков **SLACT** прикладную программу необходимо конфигурировать таким образом, чтобы всегда был активен только один функциональный блок **SLACT**.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом A:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	Передний фронт запускает блок	BOOL
A_Id	Идентификационный номер ведомого устройства (см. главу 5.10)	DWORD
A_Mode	Состояние, в которое необходимо перевести ведомое устройство PROFIBUS DP: ≠ 0: активно (установлено соединение) = 0: не активно (деактивировано)	INT

Таблица 87: A-входы Функциональный блок SLACT

A-выходы	Описание	Тип
A_Done	TRUE: Ведомое устройство PROFIBUS DP было установлено на определенное на входе A_Mode состояние.	BOOL
A_Busy	TRUE: Установка ведомого устройства PROFIBUS DP еще не завершена.	BOOL
A_Status	Состояние или код ошибки (см. главу 5.11)	DWORD

Таблица 88: A-выходы Функциональный блок SLACT

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение с функциональным блоком **SLACT** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока SLACT в структуре (в папке Функциональные блоки) с функциональным блоком SLACT (в прикладной программе) осуществляется посредством общих переменных. Эти переменные Вы должны заранее определить в редакторе переменных.

Соедините *F-Inputs* функционального блока **SLACT** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **SLACT** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Status	DWORD

Таблица 89: F-входы Функциональный блок SLACT

Соедините *F-Outputs* функционального блока **SLACT** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **SLACT** в структуре.

F-выходы	Тип
F_Req	BOOL
F_Id	DWORD
F_Mode	INT

Таблица 90: F-выходы Функциональный блок SLACT

Чтобы создать функциональный блок SLACT в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New**.
2. Выбрать функциональный блок **SLACT**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **SLACT** и выбрать **Edit**.
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **SLACT** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **SLACT** в прикладной программе.

Входы	Тип
ID	DWORD
MODE	INT
REQ	BOOL

Таблица 91: Входные системные переменные

Соедините следующие выходы функционального блока **SLACT** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **SLACT** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Таблица 92: Выходные системные переменные

Для использования функционального блока SLACT необходимо:

1. Установить в прикладной программе вход *A_Mode* на необходимое состояние.
2. В прикладной программе установить идентификатор с адресом ведомого устройства на входе *A_ID*.
3. В прикладной программе установить вход *A_Req* на TRUE.

i

Функциональный блок реагирует на передний фронт на *A_Req*.

Выход *A_Busy* является TRUE, пока команда SLACT не будет обработана. Затем выход *A_Busy* меняется на FALSE и выход *A_Done* на TRUE.

На выходе *A_Status* выдается режим ведомого устройства, если установлен режим ведомого устройства.

На выходе *A_Status* выдается код ошибки, если режим ведомого устройства не установлен.

5.9.6 Функциональный блок WRREC

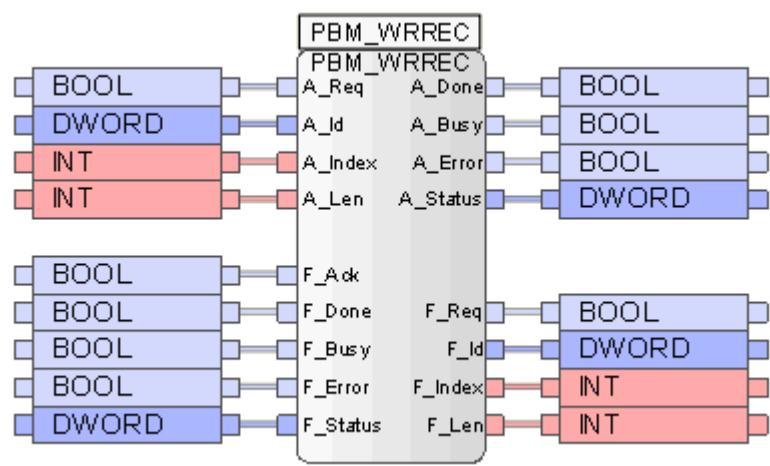


Рисунок 38: Функциональный блок WRREC

Функциональный блок **WRREC** (с DP-V1) служит для ациклической записи набора данных на ведомое устройство, адрес которого указан на *A_Index*. Информацию о том, какие данные могут быть записаны, Вы найдете в руководстве по ведомому устройству.

В ведущем устройстве HIMA PROFIBUS DP одновременно может быть активировано до 32 **RDREC** и/или **WRREC** блоков.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом А:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «А» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	Передний фронт запускает запрос для записи набора данных.	BOOL
A_ID	Идентификационный номер ведомого устройства (см. главу 5.10)	DWORD
A_Index	Номер записываемого набора данных. Значение должно содержаться в описании устройства, выпущенном изготовителем.	INT
A_Len	Длина записываемого набора данных в байтах.	INT

Таблица 93: А-входы Функциональный блок WRREC

A-выходы	Описание	Тип
A_Done	TRUE: Функциональный блок завершил процесс записи.	BOOL
A_Busy	TRUE: Функциональный блок еще не завершил процесс записи.	BOOL
A_Error	TRUE: Во время записи произошла ошибка.	BOOL
A_STATUS	Состояние или код ошибки (см. главу 5.11)	DWORD

Таблица 94: А-выходы Функциональный блок WRREC

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение с функциональным блоком **WRREC** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **WRREC** в структуре (в папке Функциональные блоки) с функциональным блоком **WRREC** (в прикладной программе) осуществляется посредством общих переменных. Эти переменные Вы должны заранее определить в редакторе переменных.

Соедините *F-Inputs* функционального блока **WRREC** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **WRREC** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD

Таблица 95: F-входы Функциональный блок **WRREC**

Соедините *F-Outputs* функционального блока **WRREC** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **WRREC** в структуре.

F-выходы	Тип
F_Req	BOOL
F_Id	DWORD
F_Index	INT
F_Len	INT

Таблица 96: F-выходы Функциональный блок **WRREC****Чтобы создать функциональный блок **WRREC** в структуре, необходимо:**

1. Открыть в структуре **Configuration, Resource, Protocols, PROFIBUS DP Master, Function Blocks, New**.
2. Выбрать функциональный блок **WRREC**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **WRREC** и выбрать **Edit**.
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **WRREC** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **WRREC** в прикладной программе.

Входы	Тип
ID	DWORD
INDEX	INT
LEN	INT
REQ	BOOL

Таблица 97: Входные системные переменные

Соедините следующие выходы функционального блока **WRREC** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **WRREC** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
DONE	BOOL
ERROR	BOOL
STATUS	DWORD

Таблица 98: Выходные системные переменные

Данные	Описание
Переменные не заданы	Во вкладке <i>Process Variables</i> может быть определена любая структура данных, которая должна все же подходить к структуре набора данных. Структура набора данных содержится в руководстве изготовителя по ведомому устройству.

Таблица 99: Данные

Для использования функционального блока WRREC необходимо:

1. Установить в прикладной программе адрес ведомого устройства на вход *A_ID*.
2. В прикладной программе установить специальный для ведомого устройства индекс для набора данных (руководство изготовителя) на входе *A_Index*.
3. В прикладной программе установить длину записываемого набора данных на входе *A_Len*.
4. В прикладной программе настроить набор данных, как определено во вкладке «Данные».
5. В прикладной программе установить вход *A_Req* на TRUE.

i

Функциональный блок реагирует на передний фронт на *A_Req*.

Выход *A_Busy* является TRUE, пока набор данных не будет записан. Затем выход *A_Busy* меняется на FALSE и выход *A_Done* на TRUE.

Если набор данных не был записан успешно, то выход *A_Error* переходит на TRUE и на выходе *A_Status* выдается код ошибки.

5.10 Вспомогательные функциональные блоки PROFIBUS

Вспомогательные функциональные блоки можно использовать для параметрирования и анализа входов и выходов функциональных блоков.

Доступны следующие вспомогательные функциональные блоки:

Вспомогательные функциональные блоки	Описание функции
ACTIVE (см. главу 5.10.1)	Ведомое устройство активно или неактивно
ALARM (см. главу 5.10.2)	Декодирование аварийных данных
DEID (см. главу 5.10.3)	Декодирование идентификационного номера
ID (см. главу 5.10.4)	ID функции генерирует из четырех байтов идентификатор
NSLOT (см. главу 5.10.5)	Создание последующего идентификационного номера для слотов
SLOT (см. главу 5.10.6)	Создание идентификационного номера слота с номером слота
STDDIAG (см. главу 5.10.7)	Декодирование стандартной диагностики ведомого устройства
LATCH	Используется только в других функциональных блоках
PIG	Используется только в других функциональных блоках
PIGII	Используется только в других функциональных блоках

Таблица 100: Обзор вспомогательных функциональных блоков

5.10.1 Вспомогательный функциональный блок ACTIVE

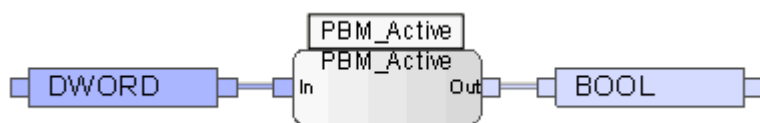


Рисунок 39: Вспомогательный функциональный блок ACTIVE

Вспомогательный функциональный блок ACTIVE определяет по стандартной диагностике ведомого устройства PROFIBUS DP, активно или неактивно в настоящий момент ведомое устройство.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 0).

Входы	Описание	Тип
IN	Стандартная диагностика ведомого устройства	DWORD

Таблица 101: Входы Вспомогательный функциональный блок ACTIVE

Выходы	Описание	Тип
OUT	TRUE: Ведомое устройство активно FALSE: Ведомое устройство неактивно	BOOL

Таблица 102: Выходы Вспомогательный функциональный блок ACTIVE

5.10.2 Вспомогательный функциональный блок ALARM
(декодирование аварийных данных)

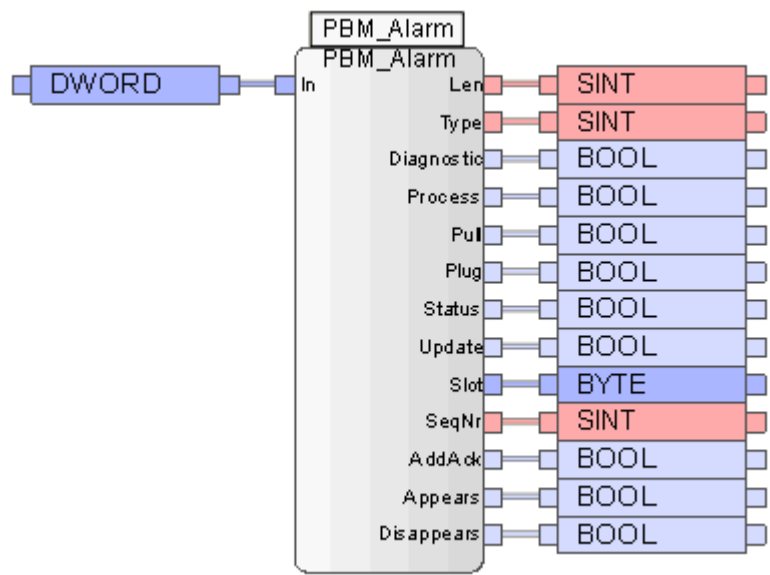


Рисунок 40: Вспомогательный функциональный блок ALARM

Вспомогательный функциональный блок **ALARM** декодирует стандартные аварийные данные ведомого устройства PROFIBUS DP.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы	Описание	Тип
IN	Стандартный аварийный сигнал	DWORD

Таблица 103: Входы Вспомогательный функциональный блок ALARM

Выход	Описание	Тип
Len	Длина общего аварийного сообщения	SINT
Type	1: аварийный сигнал: диагностика 2: аварийный сигнал: процесс 3: аварийный сигнал: извлечение 4: аварийный сигнал: вставка 5: аварийный сигнал: статус 6: аварийный сигнал: обновление Другие номера зарезервированы или используются изготовителем. Значение должно содержаться в описании устройства, выпущенном изготовителем.	SINT
Diagnostic	True = аварийный сигнал: диагностика	BOOL
Process	True = аварийный сигнал: процесс	BOOL
Pull	True = модуль извлечен	BOOL
Plug	True = модуль снова установлен	BOOL
Status	True = аварийный сигнал: состояние	BOOL
Update	True = аварийный сигнал: обновление	BOOL
Slot	Модуль, от которого поступил аварийный сигнал	BYTE
SeqNr	Порядковый номер аварийного сигнала	SINT

Выход	Описание	Тип																							
AddAck	TRUE означает, что ведомое устройство, от которого поступил данный аварийный сигнал, ожидает дополнительного подтверждения от приложения. Какое именно, необходимо посмотреть в руководстве изготовителя по ведомому устройству.	BOOL																							
Appears Disappears	<table border="1"> <thead> <tr> <th>Выход</th><th>Значение</th><th>Описание</th></tr> </thead> <tbody> <tr> <td>Appears</td><td>TRUE</td><td rowspan="2">Если оба являются FALSE, то в данный момент ошибок нет.</td></tr> <tr> <td>Disappears</td><td>FALSE</td></tr> <tr> <td>Appears</td><td>TRUE</td><td rowspan="2">Возникла ошибка и еще присутствует.</td></tr> <tr> <td>Disappears</td><td>FALSE</td></tr> <tr> <td>Appears</td><td>TRUE</td><td rowspan="2">Возникла ошибка и исчезает в настоящий момент.</td></tr> <tr> <td>Disappears</td><td>FALSE</td></tr> <tr> <td>Appears</td><td>TRUE</td><td rowspan="2">Если оба являются TRUE, то ошибка исчезает, но ведомое устройство еще не в порядке.</td></tr> <tr> <td>Disappears</td><td>FALSE</td></tr> </tbody> </table>	Выход	Значение	Описание	Appears	TRUE	Если оба являются FALSE, то в данный момент ошибок нет.	Disappears	FALSE	Appears	TRUE	Возникла ошибка и еще присутствует.	Disappears	FALSE	Appears	TRUE	Возникла ошибка и исчезает в настоящий момент.	Disappears	FALSE	Appears	TRUE	Если оба являются TRUE, то ошибка исчезает, но ведомое устройство еще не в порядке.	Disappears	FALSE	BOOL
Выход	Значение	Описание																							
Appears	TRUE	Если оба являются FALSE, то в данный момент ошибок нет.																							
Disappears	FALSE																								
Appears	TRUE	Возникла ошибка и еще присутствует.																							
Disappears	FALSE																								
Appears	TRUE	Возникла ошибка и исчезает в настоящий момент.																							
Disappears	FALSE																								
Appears	TRUE	Если оба являются TRUE, то ошибка исчезает, но ведомое устройство еще не в порядке.																							
Disappears	FALSE																								

Таблица 104: Выходы Вспомогательный функциональный блок ALARM

5.10.3 Вспомогательный функциональный блок DEID (декодирование идентификационного номера)

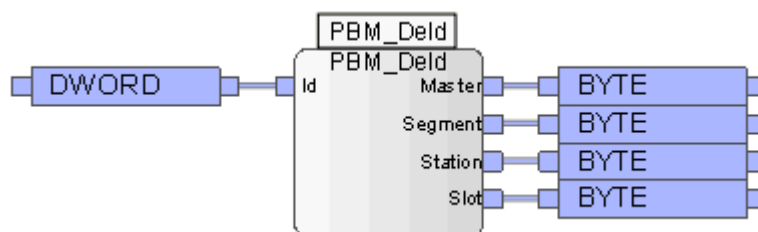


Рисунок 41: Вспомогательный функциональный блок DEID

Вспомогательный функциональный блок **DEID** декодирует идентификационный номер и раскладывает его на четыре составляющих.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы	Описание	Тип
Id	Идентификационный номер ведомого устройства	DWORD

Таблица 105: Входы Вспомогательный функциональный блок DEID

Выходы	Описание	Тип
Master	Адрес шины ведущего устройства	BYTE
Segment	Сегмент	BYTE
Station	Адрес шины ведомого устройства	BYTE
Slot	Номер слота или модуля	BYTE

Таблица 106: Выходы Вспомогательный функциональный блок DEID

5.10.4 Вспомогательный функциональный блок ID
(генерирование идентификационного номера)

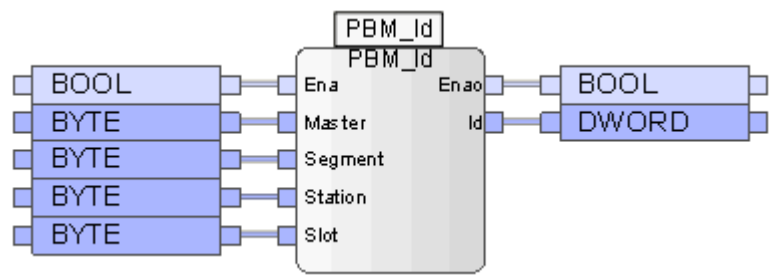


Рисунок 42: Вспомогательный функциональный блок ID

Вспомогательный функциональный блок **ID** генерирует из четырех байтов идентификатор (идентификационный номер), который используется другими вспомогательными функциональными блокам.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы	Описание	Тип
Ena	Не используется	BOOL
Master	Адрес шины	BYTE
Segment	Сегмент	BYTE
Station	Адрес шины ведомого устройства	BYTE
Slot	Номер слота или модуля	BYTE

Таблица 107: Входы Вспомогательный функциональный блок ID

Выходы	Описание	Тип
Enao	Не используется	BOOL
Id	Идентификационный номер ведомого устройства	DWORD

Таблица 108: Выходы Вспомогательный функциональный блок ID

5.10.5 Вспомогательный функциональный блок NSLOT

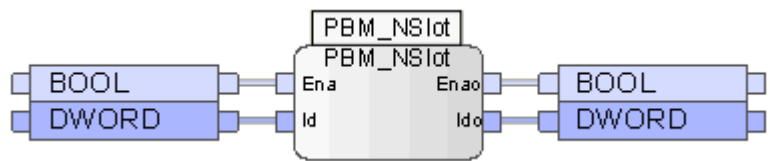


Рисунок 43: Вспомогательный функциональный блок NSLOT

Вспомогательный функциональный блок **NSLOT** генерирует из идентификатора новый идентификатор, который присваивает адрес следующему слоту в том же ведомом устройстве. Ena должен быть установлен на TRUE, чтобы вспомогательный функциональный блок работал.

Enao является TRUE, если результат на выходе Ido действителен.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы	Описание	Тип
Ena	Пока установлено TRUE, блок работает.	BOOL
Id	Идентификационный номер ведомого устройства.	DWORD

Таблица 109: Входы Вспомогательный блок NSLOT

Выходы	Описание	Тип
Enao	TRUE = результат действителен FALSE = нет других номеров слотов	BOOL
Ido	Идентификационный номер ведомого устройства.	DWORD

Таблица 110: Выходы Вспомогательный блок NSLOT

5.10.6 Вспомогательный функциональный блок SLOT

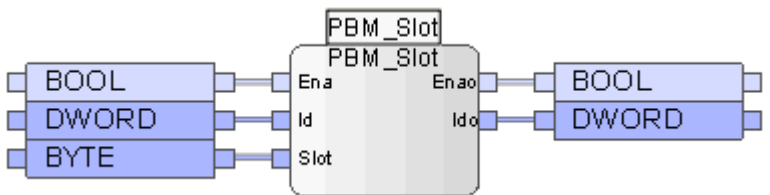


Рисунок 44: Вспомогательный функциональный блок SLOT

Вспомогательный функциональный блок **SLOT** генерирует из идентификатора и номера слота новый идентификатор, который присваивает адрес тому же ведомому устройству, что и прежний идентификатор, но с новым номером слота.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы	Описание	Тип
Ena	Не используется.	BOOL
Id	Логический адрес компонента ведомого устройства (ID ведомого устройства и номер слота).	DWORD
Slot	Номер нового слота или модуля.	BYTE

Таблица 111: Входы Вспомогательный блок SLOT

Выходы	Описание	Тип
Enao	Не используется.	BOOL
Ido	Идентификационный номер ведомого устройства.	DWORD

Таблица 112: Выходы Вспомогательный блок SLOT

5.10.7 Вспомогательный функциональный блок STDDIAG

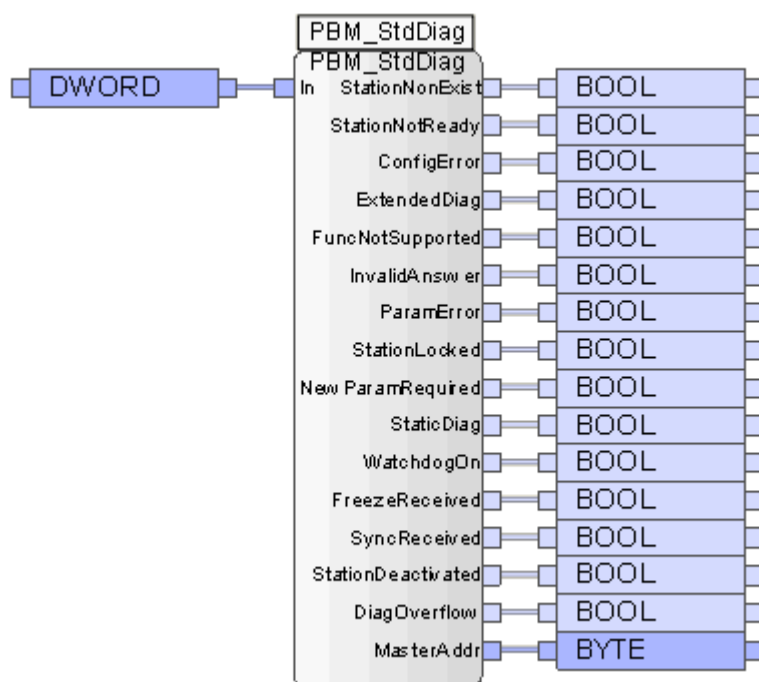


Рисунок 45: Вспомогательный функциональный блок STDDIAG

Вспомогательный функциональный блок стандартной диагностики **STDDIAG** декодирует стандартную диагностику ведомого устройства PROFIBUS DP.

Выходы типа BOOL функционального блока **STDDIAG** являются TRUE, если соответствующий бит установлен в стандартной диагностике.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы	Описание	Тип
IN	Стандартная диагностика ведомого устройства	DWORD

Таблица 113: Входы Вспомогательный блок STDDIAG

Выходы	Описание	Тип
StationNonExist	Ведомое устройство не существует	BOOL
StationNotReady	Ведомое устройство не готово	BOOL
ConfigError	Ошибка конфигурации	BOOL
ExtendedDiag	Выполняется расширенная диагностика	BOOL
FuncNotSupported	Функция не поддерживается	BOOL
InvalidAnswer	Недействительный ответ от ведомого устройства	BOOL
ParamError	Ошибка параметрирования	BOOL
StationLocked	Ведомое устройство заблокировано другим ведущим устройством	BOOL
NewParamRequired	Требуются новые данные параметрирования	BOOL
StaticDiag	Статическая диагностика	BOOL
WatchdogOn	Сторожевое устройство активировано	BOOL
FreezeReceived	Получена команда Freeze	BOOL
SyncReceived	Получена команда Sync	BOOL
StationDeactivated	Ведомое устройство деактивировано	BOOL
DiagOverflow	Переполнение диагностики	BOOL
MasterAddr	Адрес шины ведущего устройства	BYTE

Таблица 114: Выходы Вспомогательный блок STDDIAG

Чтобы считать стандартную диагностику ведомого устройства PROFIBUS DP, необходимо:

1. В структуре **Configuration, Resource, Protocols, PROFIBUS DP Master**.
2. Щелкнуть правой кнопкой мыши на **PROFIBUS Slave** и выбрать **Edit**.
3. Перетащить глобальную переменную типа DWORD в поле *Standard Diagnosis*.
4. Соединить эту глобальную переменную со входом функционального блока **STDDIAG**.

5.11 Коды ошибок функциональных блоков

Если функциональный блок не смог правильно выполнять команду, то на выходе **A_Status** выдается код ошибки. Значение кода ошибки указано в таблице ниже:

Код ошибки	Символ	Описание
16#40800800	TEMP_NOT_AVAIL	В настоящее время не доступно
16#40801000	INVALID_PARA	Недействительный параметр
16#40801100	WRONG_STATE	Ведомое устройство не поддерживает DP-V1
16#40808000	FATAL_ERR	Фатальная ошибка в программе
16#40808100	BAD_CONFIG	Ошибка конфигурации в области данных
16#40808200	PLC_STOPED	Управление остановлено
16#4080A000	READ_ERR	Ошибка при считывании данных
16#4080A100	WRITE_ERR	Ошибка при записи данных
16#4080A200	MODULE_FAILURE	Невозможно точно определить ошибку
16#4080B000	INVALID_INDEX	Недействительный индекс
16#4080B100	WRITE_LENGTH	Неверная длина при записи
16#4080B200	INVALID_SLOT	Недействительный номер слота
16#4080B300	TYPE_CONFLICT	Неверный тип
16#4080B400	INVALID_AREA	Неверная область считывания или записи
16#4080B500	STATE_CONFLICT	Ведущее устройство в неверном состоянии
16#4080B600	ACCESS_DENIED	Ведомое устройство неактивно (и т. п.)
16#4080B700	INVALID_RANGE	Неверная область считывания или записи
16#4080B800	INVALID_PARAMETER	Неверное значение параметра
16#4080B900	INVALID_TYPE	Неверный тип параметра
16#4080C300	NO_RESOURCE	Ведомое устройство отсутствует
16#4080BA00	BAD_VALUE	Недействительное значение
16#4080BB00	BUS_ERROR	Ошибка шины
16#4080BC00	INVALID_SLAVE	Недействительный Id ведомого устройства
16#4080BD00	TIMEOUT	Возникло время ожидания
16#4080C000	READ_CONSTRAIN	Ограничение считывания
16#4080C100	WRITE_CONSTRAIN	Ограничение записи
16#4080C200	BUSY	Блок данного вида уже активен
16#4080C300	NO_RESOURCE	Ведомое устройство не активно

Таблица 115: Коды ошибки функциональных блоков

5.12 Панель управления (ведущее устройство PROFIBUS DP)

В панели управления пользователь может проверять и регулировать настройки соединения ведущего устройства PROFIBUS DP. Для этого отображается актуальная информация по состоянию (напр., время цикла, состояние шины и т.д.) ведущего устройства и соответствующего ведомого устройства.

Чтобы открыть панель управления для контроля ведущего устройства PROFIBUS DP, необходимо:

1. В структуре выбрать **Resource**.
2. В строке меню выбрать **Online**.
3. В **System Log-in** ввести данные доступа, чтобы открыть панель управления ресурса.
4. В структуре панели управления выбрать **PROFIBUS DP Master**.

5.12.1 Контекстное меню (ведущее устройство PROFIBUS)

В контекстном меню выбранного ведущего устройства PROFIBUS DP можно выбрать следующие команды:

Offline:

Выключает выбранное ведущее устройство PROFIBUS DP. Если ведущее устройство выключено, то никаких действий не происходит.

Stop:

Останавливает выбранное ведущее устройство PROFIBUS DP. Ведущее устройство PROFIBUS DP продолжает участвовать в протоколе маркера, но не передает данные на ведомое устройство.

Clear:

При нажатии кнопки CLEAR выбранное ведущее устройство PROFIBUS DP переводится в безопасное состояние и выполняет теперь обмен безопасными данными с ведомым устройством. Выходные данные, отправленные на ведомое устройство, содержат только нули. Безопасные ведомые устройства получают безопасные телеграммы, не содержащие данных. Входные данные от ведомого устройства игнорируются ведущим устройством PROFIBUS DP и вместо этого используются предустановленные значения по умолчанию в прикладной программе.

Operate:

Запускает выбранное ведущее устройство PROFIBUS DP. Ведущее устройство PROFIBUS DP выполняет циклический обмен входными/выходными данными с ведомым устройством.

Reset Statistic:

Кнопка сброса статистических значений сбрасывает на нуль статистические данные (время цикла мин., макс., и т.д.).

5.12.2 Поле индикации (ведущее устройство PROFIBUS)

В поле индикации отображаются следующие значения выбранного ведущего устройства PROFIBUS DP.

Элемент	Описание
Name	Имя ведущего устройства PROFIBUS DP
Baud rate [bps]	Скорость передачи в бодах для ведущего устройства. Ведущее устройство может осуществлять коммуникацию с любой скоростью передачи в бодах, которая определена стандартом. Время цикла может составлять минимум 2 мс.
Fieldbus interface	FB1, FB2
Fieldbus address	Адрес шины ведущего устройства (0...125).
Master State	Отображает текущее состояние протокола (см. главу 5.12.3). 0 = OFFLINE 1 = STOP 2 = CLEAR 3 = OPERATE 100 = UNDEFINED
Bus state	Код ошибки шины 0...6: 0 = ОК 1 = Ошибка в адресе: Адрес ведущего устройства на шине уже существует. 2 = Неисправность шины: На шине зарегистрирована неисправность, напр., неверно подключена оконечная нагрузка шины, несколько участников осуществляют передачу одновременно. 3 = Ошибка протокола: Принят пакет с неправильной кодировкой. 4 = Ошибка аппаратного обеспечения: Получено сообщение об ошибке аппаратного обеспечения, напр., при слишком коротком настроенном времени. 5 = Неизвестная ошибка: Ведущее устройство изменило состояние по неизвестной причине. 6 = Сброс значений контроллера: При серьезной неисправности шины микросхема контроллера переходит в неопределенное состояние и происходит сброс. Код ошибки отображается до тех пор, пока ошибка шины не будет устранена.
Bus error count	Число ошибок шины до настоящего момента.
CPU Load (planned) [%]	Индикация запланированной нагрузки модуля COM для данного протокола.
CPU Load (actual) [%]	Фактическая нагрузка модуля COM для данного протокола.
MSI [ms]	Мин. интервал ведомого устройства в мс, срабатывание 0,1 мс.
TTR [ms]	Контрольное время обращения в мс, срабатывание 0,1 мс.
Last Cycle Time [ms]	Последнее время цикла PROFIBUS DP [мс].
Minimum cycle time [ms]	Минимальное время цикла PROFIBUS DP [мс].
Average cycle time [ms]	Среднее время цикла PROFIBUS DP [мс].
Maximum cycle time [ms]	Максимальное время цикла PROFIBUS DP [мс].

Таблица 116: Поле индикации ведущего устройства PROFIBUS

5.12.3 Состояние ведущего устройства PROFIBUS DP

Состояние ведущего устройства отображается в поле индикации панели управления и может анализироваться с помощью переменной состояния ведущего устройства «Состояние соединения» в прикладной программе.

Состояние ведущего устройства	Состояние ведущего устройства
OFFLINE	Ведущее устройство выключено и на шине не происходит никаких действий.
STOP	Ведущее устройство участвует в протоколе маркера, но не передает данные на ведомое устройство.
CLEAR	Ведущее устройство находится в безопасном состоянии и выполняет обмен данными с ведомым устройством. <ul style="list-style-type: none"> Выходные данные, отправленные на ведомое устройство, содержат только нули. Безопасные ведомые устройства получают безопасные телеграммы, не содержащие данных. Входные данные от ведомого устройства игнорируются, и вместо этого используются предустановленные значения по умолчанию.
OPERATE	Ведущее устройство находится в рабочем режиме и выполняет циклический обмен входными/выходными данными с ведомым устройством.
UNDEFINED	Выполняется обновление встроенной программы модуля ведущего устройства PROFIBUS DP.

Таблица 117: Состояние ведущего устройства PROFIBUS DP

5.12.4 Поведение ведущего устройства PROFIBUS DP

Поведение ведущего устройства PROFIBUS DP в зависимости от рабочего состояния системы управления.

Состояние Система управления	Поведение ведущего устройства PROFIBUS DP HIMA
STOP ^{*)}	Система управления в режиме STOP, затем ведущее устройство в состоянии OFFLINE.
RUN	Система управления в режиме RUN, ведущее устройство выполняет попытку перейти в состояние OPERATE.
STOP	Система управления в режиме STOP, ведущее устройство переходит в состояние CLEAR. Если ведущее устройство уже находится в режиме STOP или OFFLINE, оно остается в этом состоянии.

^{*)} После включения системы управления или после загрузки конфигурации.

Таблица 118: Поведение ведущего устройства PROFIBUS DP

5.12.5 Функция светодиода FBx у ведущего устройства PROFIBUS DP

Состояние последовательной связи PROFIBUS DP отображается с помощью светодиода FBx соответствующих конфигурированных последовательных интерфейсов (fb1, fb2).

Светодиод FBx	Описание
ВЫКЛ	Отсутствует или недействительная конфигурация ведущего устройства PROFIBUS DP.
Мигает с периодичностью в 2 с	Действительная конфигурация. Ведущее устройство PROFIBUS DP находится в состоянии OFFLINE или STOP.
ВКЛ	Ведущее устройство PROFIBUS DP находится в состоянии OPERATE или CLEAR и выполняет обмен данными со всеми активированными ведомыми устройствами.
Мигает с периодичностью в 1 с	Минимум одно ведомое устройство вышло из строя.

Таблица 119: Светодиод FBx (ведущее устройство PROFIBUS DP)

5.13 Ведомое устройство HIMA PROFIBUS DP

В данной главе описываются свойства ведомого устройства HIMA PROFIBUS DP, а также функции меню в SILworX, которые потребуются для конфигурирования ведомого устройства HIMA PROFIBUS DP.

Необходимое оснащение и требования к системе

Элемент	Описание
HIMA controller	HIMax с модулем COM.
COM module	Модуль COM HIMax на используемом последовательном интерфейсе полевой шины (FB1 или FB2) должен быть оснащен опциональным субмодулем ведомого устройства HIMA PROFIBUS DP. Распределение интерфейсов, см. главу 3.5.
Activation	Активация посредством вставного модуля, см. главу 3.5.2.

Таблица 120: Оснащение и требования к системе ведомого устройства HIMA PROFIBUS DP

Свойства ведомого устройства PROFIBUS DP

Элемент	Описание
Type of HIMA PROFIBUS DP slave	DP-V0
Transfer rate	9,6 кбит/с...12 Мбит/с
Bus address	0...125
Max. number of slaves	Для каждого модуля COM можно конфигурировать одно ведомое устройство HIMA PROFIBUS DP.
Process data volume of a HIMA PROFIBUS DP slave	Вывод DP: макс. 192 байта Ввод DP: макс. 240 байта Всего: макс. 256 байта
Protocol watchdog	Если COM находится в состоянии RUN и потеряно соединение с ведущим устройством PROFIBUS DP, то это распознается ведомым устройством по истечении времени ожидания сторожевого устройства (должно быть параметрировано в ведущем устройстве). В этом случае выходные данные DP (входные данные с точки зрения ресурса) сбрасываются на предустановленное значение по умолчанию и флажок <i>Data Valid</i> (переменная состояния протокола ведомого устройства DP) устанавливается на FALSE.

Таблица 121: Свойства ведомого устройства HIMA PROFIBUS DP

5.13.1 Создание ведомого устройства HIMA PROFIBUS DP

Чтобы создать новое ведомое устройство HIMA PROFIBUS DP, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols**.
2. В контекстном меню протокола выбрать **New, PROFIBUS DP Slave**, чтобы добавить новое ведомое устройство PROFIBUS DP.
3. В контекстном меню ведомого устройства PROFIBUS DP выбрать **Edit**.
4. Во вкладке **Properties** выбрать **Modul** и **Interfaces**.

5.14 Функции меню ведомого устройства PROFIBUS DP

5.14.1 Редактирование (Edit)

Диалоговое окно **Edit** ведущего устройства PROFIBUS DP содержит следующие вкладки.

Переменные процесса

Во вкладке **Process Variables** создаются переменные для приема и передачи.

Входные переменные (Input Variables)

Переменные, которые должны приниматься данной системой управления, вводятся в область *Input Signals*.

В области *Input Signals* можно создавать любые переменные. Смещения и типы переменных должны совпадать со смещениями и типами переменных передачи участника коммуникации.

Выходные переменные (Output Variables)

Переменные для циклического обмена данными, которые должны передаваться этой системой управления, вводятся в область *Output Signals*.

В области *Output Signals* можно создавать любые переменные. Смещения и типы переменных должны совпадать со смещениями и типами переменных приема участника коммуникации.

Системные переменные

Во вкладке **System Variables** определяется, какие переменные должны считываться системой управления.

Вкладка **System Variables** содержит следующие системные переменные, которые позволяют анализировать состояние ведомого устройства PROFIBUS DP в прикладной программе.

Элемент	Описание
Current baud rate [UDINT]	Скорость передачи в бодах, с которой протокол ведомого устройства PROFIBUS-DP работает в настоящий момент.
Data valid [BOOL]	<p>Если переменная состояния <i>Data Valid</i> установлена на TRUE, то ведомое устройство приняло действительные данные импорта ведущего устройства.</p> <p>Переменная состояния установлена на FALSE, если у ведомого устройства истекло время сторожевого устройства.</p> <p>Значение по умолчанию: FALSE</p> <p>Указание:</p> <p>Если сторожевое устройство ведомого устройства не активировано ведущим устройством и соединение прервано, то переменная состояния <i>Data Valid</i> сохраняет значение TRUE, поскольку ведомое устройство PROFIBUS DP не может распознать потерю соединения.</p> <p>Данное обстоятельство необходимо обязательно учитывать при использовании этих переменных!</p>

Элемент	Описание
Error code [DWORD]	<p>Если в протоколе ведомого устройства PROFIBUS-DP возникла ошибка, то она переносится в данную переменную. Отображается возникшая актуальная ошибка.</p> <p>Возможные значения (шестнадцатеричные):</p> <p>0x00: Нет ошибки</p> <p>0xE1: Неверное параметрирование ведущим устройством PROFIBUS DP</p> <p>0xD2: Неверное конфигурирование ведущим устройством PROFIBUS DP</p> <p>Значение по умолчанию: 0x00</p>
Master ID [USINT]	<p>Это ID ведущего устройства PROFIBUS DP, которое параметрировало и конфигурировало собственное ведомое устройство PROFIBUS DP.</p> <p>Возможные значения (десятичные):</p> <p>0–125: ID ведущего устройства</p> <p>255: Ведомое устройство в настоящее время не присвоено ни одному ведущему устройству</p> <p>Значение по умолчанию: 0xFF</p>
Protocol state [BYTE]	<p>Описывает состояние протокола ведомого устройства PROFIBUS-DP.</p> <p>Возможные значения (шестнадцатеричные):</p> <p>0xE1: Система управления отсоединена от шины либо не активна.</p> <p>0xD2: Система управления ожидает конфигурирования ведущим устройством PROFIBUS DP.</p> <p>0xC3: Система управления выполняет циклический обмен данными с ведущим устройством PROFIBUS DP.</p> <p>Значение по умолчанию: 0xE1</p>
Slave ID [USINT]	<p>Данная переменная содержит ID ведомого устройства PROFIBUS DP системы управления, с которой она работает на шине. Данный ID ведомого устройства предварительно конфигурирован пользователем посредством PADT.</p> <p>Возможные значения (десятичные):</p> <p>0–125: ID ведомого устройства PROFIBUS DP собственной системы управления</p> <p>Значение по умолчанию: 0xFF</p>
Watchdog time [UINT]	<p>Параметрируемое в ведущем устройстве время сторожевого устройства в миллисекундах. См. главу 5.6.3.</p>

Таблица 122: Системные переменные ведомого устройства PROFIBUS DP

5.14.2 Свойства

Вкладка **Properties** ведомого устройства HIMA PROFIBUS DP содержит следующие параметры для конфигурирования ведомого устройства PROFIBUS DP.

Заданные значения для параметров *In one cycl* и *Refresh Rate [ms]* обеспечивают быстрый обмен данными PROFIBUS DP между модулем COM и аппаратным обеспечением ведомого устройства системы управления HIMAх.

Данные параметры необходимо изменять только в том случае, если снижение нагрузки COM требуется для приложения, и это допустимо для процесса.

i

Изменение параметров рекомендуется доверять только опытному программисту. Увеличение времени обновления аппаратного обеспечения COM/PROFIBUS DP означает также, что увеличится фактическое время обновления данных PROFIBUS DP. Необходимо проверять временные требования устройства.

Учитывайте также параметр **Min. Slave Intervall [ms]** (см. главу 5.3.2 во вкладке Время), который определяет минимальное время обновления данных PROFIBUS DP между ведущим устройством PROFIBUS DP и ведомым устройством PROFIBUS DP.

Элемент	Описание
Type	Ведомое устройство PROFIBUS DP.
Name	Имя ведомого устройства PROFIBUS DP.
Within one cycle	Активировано: Обмен всеми данными протокола с ЦПУ на COM в течение одного цикла ЦПУ. Деактивировано: Обмен всеми данными протокола с ЦПУ на COM, распределенный на несколько циклов ЦПУ по 900 байтов на направление. Таким образом, при необходимости можно снизить время цикла системы управления. Значение по умолчанию: Активировано
Module	Выбор модуля COM, на котором обрабатывается данный протокол.
Use Max CPU Load	Активировано: Сохранить предельное значение нагрузки ЦПУ из поля Max. CPU Load [%]. Деактивировано: Не использовать предельное значение нагрузки ЦПУ для данного протокола.
Max. CPU Load [%]	Максимальная нагрузка модуля ЦПУ, которая может создаваться при обработке протокола. Диапазон значений: 1...100% Значение по умолчанию: 30%
Station address	Адрес станции ведомого устройства. На шине может быть только один адрес станции ведомого устройства. Диапазон значений: 1...125 Значение по умолчанию: 0
Refresh Rate [ms]	Время обновления в миллисекундах, когда происходит обмен данными протокола между COM и аппаратным обеспечением ведомого устройства PROFIBUS DP. Диапазон значений: 4 ...1000 Значение по умолчанию: 10

Элемент	Описание																																												
Interface	Интерфейс полевой шины, которая используется для ведомого устройства PROFIBUS DP. Диапазон значений: fb1, fb2 Значение по умолчанию: отсутствует																																												
Baud rate [bps]	Скорость передачи в бодах, с которой эксплуатируется шина. Возможные значения: <table><tr><th>Значение</th><th>Скорость передачи в бодах</th><th>FB1</th><th>FB2</th></tr><tr><td>9600</td><td>9,6 кбит/с</td><td>X</td><td>X</td></tr><tr><td>19200</td><td>16,2 кбит/с</td><td>X</td><td>X</td></tr><tr><td>45450</td><td>45,45 кбит/с</td><td>X</td><td>X</td></tr><tr><td>93750</td><td>93,75 кбит/с</td><td>X</td><td>X</td></tr><tr><td>187500</td><td>187,5 кбит/с</td><td>X</td><td>X</td></tr><tr><td>500000</td><td>500 кбит/с</td><td>X</td><td>X</td></tr><tr><td>1500000</td><td>1,5 Мбит/с</td><td>X</td><td>X</td></tr><tr><td>3000000</td><td>3 Мбит/с</td><td>X</td><td>-</td></tr><tr><td>6000000</td><td>6 Мбит/с</td><td>X</td><td>-</td></tr><tr><td>12000000</td><td>12 Мбит/с</td><td>X</td><td>-</td></tr></table>	Значение	Скорость передачи в бодах	FB1	FB2	9600	9,6 кбит/с	X	X	19200	16,2 кбит/с	X	X	45450	45,45 кбит/с	X	X	93750	93,75 кбит/с	X	X	187500	187,5 кбит/с	X	X	500000	500 кбит/с	X	X	1500000	1,5 Мбит/с	X	X	3000000	3 Мбит/с	X	-	6000000	6 Мбит/с	X	-	12000000	12 Мбит/с	X	-
Значение	Скорость передачи в бодах	FB1	FB2																																										
9600	9,6 кбит/с	X	X																																										
19200	16,2 кбит/с	X	X																																										
45450	45,45 кбит/с	X	X																																										
93750	93,75 кбит/с	X	X																																										
187500	187,5 кбит/с	X	X																																										
500000	500 кбит/с	X	X																																										
1500000	1,5 Мбит/с	X	X																																										
3000000	3 Мбит/с	X	-																																										
6000000	6 Мбит/с	X	-																																										
12000000	12 Мбит/с	X	-																																										

Таблица 123: Свойства ведомого устройства: Вкладка Общее

5.15 Панель управления (ведомое устройство PROFIBUS DP)

В панели управления пользователь может проверять и регулировать настройки соединения ведомого устройства PROFIBUS DP. Для этого отображается актуальная информация по состоянию (напр., время цикла, состояние шины и т.д.) ведомого устройства.

Чтобы открыть панель управления для контроля ведомого устройства PROFIBUS DP, необходимо:

1. В структуре выбрать **Resource**.
2. В **строке меню** выбрать Online.
3. В **System Log-in** ввести данные доступа, чтобы открыть панель управления ресурса.
4. В структуре панели управления выбрать **PROFIBUS DP Slave**.

5.15.1 Контекстное меню (ведомое устройство PROFIBUS DP)

В контекстном меню выбранного ведомого устройства PROFIBUS DP можно выбрать следующие команды:

Активировать:

Активирует выбранное ведомое устройство, которое теперь может обмениваться данными с ведущим устройством PROFIBUS DP.

Деактивировать:

Деактивирует выбранное ведомое устройство. Коммуникация завершена.

5.15.2 Поле индикации (ведомое устройство PROFIBUS DP)

В поле индикации отображаются следующие значения выбранного ведущего устройства PROFIBUS DP.

Элемент	Описание
Name	Имя ведомого устройства PROFIBUS DP.
Fieldbus interface	Присвоенный интерфейс полевой шины ведомого устройства.
Protocol State	Состояние соединения 0 = Деактивировано, 1 = Неактивно (пытается установить соединение), 2 = Соединение установлено
Error state	См. главу 5.14.1.
Timeout	Параметрируемое в ведущем устройстве время сторожевого устройства в миллисекундах. См. главу 5.6.3.
Watchdog time [ms]	Настраивается в ведущем устройстве. См. главу 5.6.3.
Fieldbus address	См. главу 5.14.2.
Master address	Имя ведущего устройства PROFIBUS DP.
Baud rate [bps]	Актуальная скорость передачи в бодах. См. главу 5.14.2.
CPU Load (planned) [%]	Индикация запланированной нагрузки модуля COM для данного протокола.
CPU Load (actual) [%]	Фактическая нагрузка модуля COM для данного протокола.

Таблица 124: Поле индикации (PROFIBUS DP Slave)

5.16 Функция светодиода FBx у ведомого устройства PROFIBUS DP

Модуль COM указывает состояние локального протокола ведомого устройства PROFIBUS-DP с помощью одного из светодиодов, присвоенных соответствующему интерфейсу полевой шины. Состояния данных светодиодов указаны в таблице ниже:

Светодиод FBx	Цвет	Описание
ВЫКЛ	Желтый	Протокол ведомого устройства PROFIBUS-DP не активен! То есть система управления находится в состоянии STOP или ведомое устройство PROFIBUS DP не конфигурировано.
Мигает с периодичностью в 2 секунды	Желтый	Обмен данными не происходит! ведомое устройство PROFIBUS DP конфигурировано и готово.
ВКЛ	Желтый	Протокол ведомого устройства PROFIBUS-DP активен и находится в состоянии обмена данными с ведущим устройством PROFIBUS DP.
ВЫКЛ	Красный	Протокол ведомого устройства PROFIBUS-DP без сбоя.
Мигает	Красный	Следующие события ведут к сбою. Конфигурации ведущего и/или ведомого устройства PROFIBUS DP содержат ошибки либо не подходят друг к другу. Превышение бюджета времени вычислений. Если через 5 секунд не возникает событие ошибки, то индикация переходит в состояние «Protocol not disturbed».

Таблица 125: Светодиод FBx (PROFIBUS DP Slave)

6 Modbus

Соединение Modbus систем HIMax практически со всеми системами управления процессом и визуализации может осуществляться либо напрямую через интерфейсы RS485, либо через интерфейсы систем управления. Системы HIMax могут эксплуатироваться как в качестве ведущего, так и в качестве ведомого устройства.

Функциональность Modbus облегчает, прежде всего, привязку панелей управления или других систем управления. Благодаря интенсивному распространению и использованию в проектах по всему миру Modbus был много раз испытан на практике.

- Ведущее устройство Modbus (см. главу 6.1)
- ведомое устройство Modbus (см. главу 6.4.1)

i

При использовании интерфейсов Ethernet в качестве канала передачи система управления HIMax и участник коммуникации должны находиться в одной и той же подсети или при использовании маршрутизатора иметь соответствующие записи маршрута.

6.1 Ведущее устройство Modbus HIMA

Передачу данных между ведущим устройством Modbus HIMA и ведомым устройством Modbus Вы можете конфигурировать через последовательный интерфейс (RS485) или через TCP/UDP (Ethernet). Кроме того, Вы можете использовать ведущее устройство HIMA Modbus в качестве шлюза (Modbus от TCP/UDP -> RS485).

Необходимое оснащение и требования к системе

Элемент	Описание
HIMA controller	HIMax с модулем COM
COM module	Если используется Modbus RTU, то модуль COM на используемых последовательных интерфейсах полевой шины (FB1 или FB2) должен быть оснащен опциональным submodule HIMA RS485. Распределение интерфейсов, см. главу 3.5.
Activation	Любую функцию ведущего устройства Modbus необходимо деблокировать отдельно, см. главу 3.5.2. Ведущее устройство Modbus RTU Ведущее устройство Modbus TCP

Таблица 126: Оснащение и требования к системе ведущего устройства Modbus

Свойства ведущего устройства Modbus

Свойство	Описание																									
Modbus master	Для каждого модуля COM можно конфигурировать одно ведущее устройство Modbus. Ведущее устройство Modbus может одновременно: обслуживать ведомые устройства TCP/UDP, последовательные ведомые устройства на нескольких последовательных шинах и может служить в качестве шлюза от Modbus TCP на Modbus RTU.																									
Max. number of Modbus slaves	Ведущее устройство Modbus может обслуживать до 247 ведомых устройств. 121 ведомое устройство Modbus на один последовательный интерфейс 64 ведомых устройства TCP через соединение TCP/IP. 247 ведомых устройств UDP через соединение UDP/IP.																									
Max. number of request telegrams	Можно конфигурировать до 988 телеграмм-запросов для одного ведущего устройства Modbus.																									
Max. length of the request telegram	См. спецификацию <i>Modbus Application Protocol Specification</i> www.modbus.org																									
Max. size of process data	Всего можно передавать 128 кБ данных и принимать 128 кБ данных. <div><div>1</div><div>Байты состояния ведущего устройства и байты состояния каждого присвоенного ведомого устройства необходимо отнять от значения макс. размера данных процесса (128 кБ).</div></div>																									
Display format of the MODBUS data	Системы управления HIMax используют формат Big Endian для передачи данных. Пример 32 бит данных (напр., DWORD, DINT): <table><tr><td>32 бит данных (hex)</td><td colspan="4">0x12345678</td></tr><tr><td>Смещение в памяти</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>Big Endian (HIMax)</td><td>12</td><td>34</td><td>56</td><td>78</td></tr><tr><td>Middle Endian (H51q)</td><td>56</td><td>78</td><td>12</td><td>34</td></tr><tr><td>Little Endian</td><td>78</td><td>56</td><td>34</td><td>12</td></tr></table>	32 бит данных (hex)	0x12345678				Смещение в памяти	0	1	2	3	Big Endian (HIMax)	12	34	56	78	Middle Endian (H51q)	56	78	12	34	Little Endian	78	56	34	12
32 бит данных (hex)	0x12345678																									
Смещение в памяти	0	1	2	3																						
Big Endian (HIMax)	12	34	56	78																						
Middle Endian (H51q)	56	78	12	34																						
Little Endian	78	56	34	12																						

Таблица 127: Свойства ведущего устройства Modbus

Согласно норме допускается всего три промежуточных усилителя, таким образом, возможно максимум 121 ведомое устройство на один последовательный интерфейс ведущего устройства. Ведущее устройство Modbus может одновременно обслуживать максимум 64 ведомых устройства TCP/IP и 247 ведомых устройств UDP/IP.

6.2 Пример Modbus

На данном примере ведущее устройство Modbus обменивается данными с ведомым устройством Modbus HIMA через Modbus TCP.

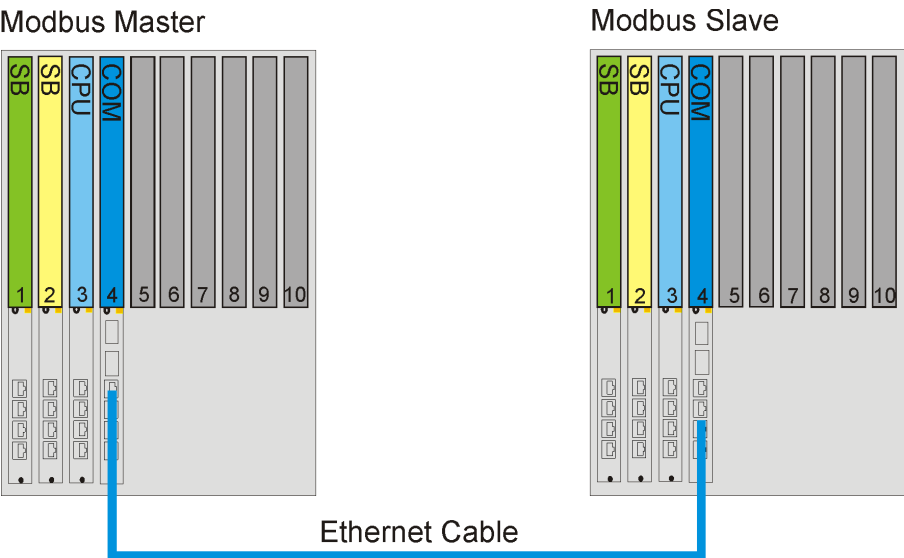


Рисунок 46: Связь через Modbus TCP

Для данного примера в SILworX необходимо создать следующие глобальные переменные:

Глобальная переменная	Описание	Тип
Var1	Master->Slave	INT
Var2	Master->Slave	SINT
Var3	Master->Slave	SINT
Var4	Master<-Slave	SINT
Var5	Master<-Slave	REAL
Var6	Master<-Slave	BOOL
Var7	Master<-Slave	BOOL
Var8	Master<-Slave	BOOL

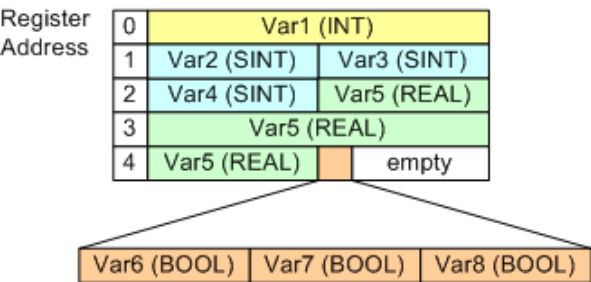


Рисунок 47: Массив переменных в стандартном режиме Modbus

6.2.1 Конфигурация ведущего устройства Modbus TCP

Чтобы создать ведущее устройство Modbus HIMA, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols**.
2. В контекстном меню протокола выбрать **New, Modbus Master**, чтобы добавить новое ведущее устройство Modbus.
3. В контекстном меню ведущего устройства Modbus выбрать **Properties, General**.
4. Выбрать **COM Modul**.
Остальные параметры сохраняют значения по умолчанию.

Чтобы создать в ведущем устройстве Modbus соединение с ведомым устройством Modbus TCP, необходимо:

1. В структуре открыть **Resource, Protocols, Modbus Master, Ethernet Slaves**.
2. Щелкнуть правой кнопкой мыши на ведомые устройства **Ethernet Slaves** и в контекстном меню выбрать **New**.
3. Выбрать из списка **TCP/UDP Slave** и подтвердить с помощью **OK**.
4. Конфигурирование ведомого устройства TCP/UDP в ведущем устройстве Modbus: выбрать **Edit** для присвоения системных переменных, см. главу 6.4.8. Выбрать **Properties** для конфигурирования свойств, см. главу 6.4.9. В свойствах ведомого устройства внести **IP Address** ведомого устройства TCP/UDP. Остальные параметры сохраняют значения по умолчанию.

Чтобы определить в ведущем устройстве Modbus телеграмму-запрос для записи выходных переменных, необходимо:

1. Щелкнуть правой кнопкой мыши на ведомое устройство **TCP/UDP-Slave** и выбрать в контекстном меню **New**.
2. Выбрать из списка телеграмму-запрос **Write Multiple Registers (16)**.

Заголовок телеграммы	Начальный адрес вкладки	Количество вкладки	CRC
Slave Address Function 16	0	2	Check sum

3. Щелкнуть правой кнопкой мыши на телеграмму-запрос **Write Multiple Registers (16)** и выбрать в контекстном меню **Edit**.
3. В **Object Panel** выбрать следующие глобальные переменные и перетащить их с помощью Drag&Drop во вкладку **Output Variables**.

Глобальная переменная	Описание	Тип
Var1	Master->Slave	INT
Var2	Master->Slave	SINT
Var3	Master->Slave	SINT

4. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Output Variables**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Чтобы определить в ведущем устройстве Modbus телеграмму-запрос для считывания входных переменных, необходимо:

1. Щелкнуть правой кнопкой мыши на ведомое устройство **TCP/UDP** и выбрать в контекстном меню **New**.
2. Выбрать из списка телеграмму-запрос **Read Holding Registers (03)**.

Заголовок телеграммы	Начальный адрес вкладки	Количество вкладок	CRC
Slave Address Function 03	2	3	Check sum

3. Щелкнуть правой кнопкой мыши на телеграмму-запрос **Read Holding Registers (03)** и выбрать в контекстном меню **Edit**.
5. В **Object Panel** выбрать следующие глобальные переменные и перетащить их с помощью Drag&Drop во вкладку **Input Variables**.

Глобальная переменная	Описание	Тип
Var4	Master<-Slave	SINT
Var5	Master<-Slave	REAL
Var6	Master<-Slave	BOOL
Var7	Master<-Slave	BOOL
Var8	Master<-Slave	BOOL

6. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Output Variables**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Чтобы проверить конфигурацию ведущего устройства Modbus TCP, необходимо:

1. Открыть контекстное меню ведущего устройства Modbus TCP и выбрать **Verification**.

Чтобы проверить конфигурацию ведущего устройства Modbus TCP, необходимо:

1. Открыть контекстное меню ведущего устройства Modbus TCP и выбрать **Verification**.
2. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.

6.2.2 Конфигурация ведомого устройства Modbus TCP

Чтобы создать новое ведомое устройство Modbus HIMA, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols**.
2. В контекстном меню выбрать **New, Modbus Slave**, чтобы добавить новое ведомое устройство Modbus.
3. В контекстном меню ведомого устройства Modbus выбрать **Edit, General**, выбрать следующие настройки
 - **модуль COM**
 - Адресация через бит (1 бит) и регистр (16 бит)

Остальные параметры сохраняют значения по умолчанию.

Чтобы конфигурировать выходные переменные ведомого устройства Modbus, необходимо:

1. В контекстном меню ведомого устройства Modbus выбрать **Edit, Register Variables**.
2. В **Object Panel** выбрать следующие глобальные переменные и перетащить их с помощью Drag&Drop в область **Register Outputs**.

Глобальная переменная	Описание	Тип
Var4	Master<-Slave	SINT
Var5	Master<-Slave	REAL
Var6	Master<-Slave	BOOL
Var7	Master<-Slave	BOOL
Var8	Master<-Slave	BOOL

3. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Register Outputs**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Чтобы конфигурировать входные переменные ведомого устройства Modbus, необходимо:

1. В контекстном меню ведомого устройства Modbus выбрать **Edit, Register Variables**.
2. В **Object Panel** выбрать следующие глобальные переменные и перетащить их с помощью Drag&Drop в область **Register Inputs**.

Глобальная переменная	Описание	Тип
Var1	Master->Slave	INT
Var2	Master->Slave	SINT
Var3	Master->Slave	SINT

3. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Register Inputs**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Чтобы проверить конфигурацию ведомого устройства Modbus TCP, необходимо:

1. Открыть контекстное меню ведомого устройства TCP и выбрать **Verification**.
2. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.

Чтобы создать код для систем управления, необходимо:

1. Запустить генератор кода ведущего устройства и ресурс ведомого устройства.
2. Убедиться, что коды сгенерированы без ошибок.
3. Загрузить соответствующие коды в системы управления ведущего и ведомого устройства.

6.3 Функции меню ведущего устройства Modbus HIMA

6.3.1 Редактирование (Edit)

Диалоговое окно **Edit** ведущего устройства Modbus содержит следующие вкладки.

Системные переменные

Вкладка **System Variables** содержит системные переменные, которые позволяют проанализировать состояние ведущего устройства Modbus в прикладной программе и управлять ведущим устройством Modbus.

Элемент	Описание
Number of bad slave connections [DWORD]	Количество соединений, содержащих ошибки, с ведомыми устройствами Modbus, которые находятся в состоянии Activated (Активировано). Деактивированные ведомые устройства Modbus здесь не учитываются.
Modbus Master Activation Control [BYTE]	Вместе с этим, ведущее устройство Modbus может быть остановлено или запущено прикладной программой. 0:START 1:STOP
Modbus Master Bus Error [BYTE]	Ошибка шины на RS485, напр., ошибка телеграммы (неизвестные коды и т.д.), ошибка длины.
Modbus Master State [BYTE]	Состояние ведущего устройства Modbus отображает текущее состояние протокола: 1: OPERATE 0: OFFLINE
Reset all slave errors [BYTE]	При смене FALSE->TRUE все ошибки ведомого устройства и ошибки шины сбрасываются.

Таблица 128: Системные переменные ведущего устройства Modbus

6.3.2 Свойства

Функция меню **Properties** (Свойства) из контекстного меню ведущего устройства Modbus открывает диалоговое окно *Properties*.

Диалоговое окно содержит следующие вкладки:

Общие положения

Во вкладке **General** вводится имя и описание для ведущего устройства Modbus. Кроме того, здесь настраиваются параметры, если ведущее устройство Modbus должно дополнительно работать как TCP и/или шлюз UDP.

Элемент	Описание
Type	Ведущее устройство Modbus
Name	Имя ведущего устройства Modbus
Description	Описание ведущего устройства Modbus
Module	Выбор модуля COM, на котором обрабатывается данный протокол.
Use Max CPU Load	Активировано: Сохранить предельное значение нагрузки ЦПУ из поля Max. CPU Load [%]. Деактивировано: Не использовать предельное значение нагрузки ЦПУ для данного протокола.
Max. CPU Load [%]	Максимальная нагрузка модуля ЦПУ, которая может создаваться при обработке протокола. Диапазон значений: 1...100% Значение по умолчанию: 30%
Enable TCP Gateway	Если активирован шлюз TCP Modbus, то необходимо конфигурировать минимум один интерфейс RS485 Modbus.
TCP Server Port	Стандарт: 502 Можно конфигурировать также другие порты TCP. При этом следует учитывать расположение портов в <i>Internet Corporation for Assigned Names and Numbers</i> (ICANN).
Maximum Number of TCP connections operating as server.	Максимальное количество одновременно открытых соединений TCP в качестве сервера. Диапазон значений: 1...64 Значение по умолчанию: 5
Enable UDP gateway	Если активирован шлюз UDP Modbus, то необходимо конфигурировать минимум один интерфейс RS485 Modbus.
UDP Port	Стандарт: 502 Можно конфигурировать также другие порты UDP. При этом следует учитывать расположение портов в <i>Internet Corporation for Assigned Names and Numbers</i> (ICANN).
Maximum length of the queue	Длина очереди шлюза для еще неотвеченных телеграмм-запросов от других ведущих устройств. Это учитывается только в том случае, если активирован шлюз. Диапазон значений: 1...20 Значение по умолчанию: 3

Таблица 129: Общие свойства ведущего устройства Modbus

ЦПУ/COM

Значения по умолчанию для параметров обеспечивают высокоскоростной обмен данными Modbus между модулем COM и модулем ЦПУ в системе управления HIMax.

Данные параметры необходимо изменять только в том случае, если снижение нагрузки COM и/или ЦПУ требуется для применения, и это допустимо для процесса.

i

Изменение параметров рекомендуется доверять только опытному программисту.

Увеличение времени обновления COM и ЦПУ означает также, что увеличится фактическое время обновления данных Modbus. Необходимо проверять временные требования устройства.

Элемент	Описание
Refresh Rate [ms]	<p>Время обновления в миллисекундах, когда происходит обмен данными протокола между COM и ЦПУ.</p> <p>Если значение <i>Refresh Rate</i> равняется нулю или меньше, чем время цикла системы управления, то обмен данными осуществляется так быстро, насколько это возможно.</p> <p>Диапазон значений: $0 \dots (2^{31} - 1)$</p> <p>Значение по умолчанию: 0</p>
Within one cycle	<p>Активировано: Обмен всеми данными протокола с ЦПУ на COM в течение одного цикла ЦПУ.</p> <p>Деактивировано: Обмен всеми данными протокола с ЦПУ на COM, распределенный на несколько циклов ЦПУ по 900 байтов на направление.</p> <p>Таким образом, при необходимости можно снизить время цикла системы управления.</p> <p>Значение по умолчанию: Активировано</p>

Таблица 130: Параметры COM/ЦПУ

6.4 Функции Modbus (телеграмма-запрос)

Благодаря функциям Modbus у вас есть возможность записывать и считывать переменные в оба направления. Считываться и записываться могут отдельные переменные или несколько следующих друг за другом переменных.

i

Более подробную информацию по Modbus Вы найдете в спецификации *Modbus Application Protocol Specification* www.modbus.org

Элемент	Код	Тип	Значение
READ COILS	01	BOOL	Считывание нескольких переменных (BOOL) с ведомого устройства.
READ DISCRETE INPUTS	02	BOOL	Считывание нескольких переменных (BOOL) с ведомого устройства.
READ HOLDIN G REGISTERS	03	WORD	Считывание нескольких переменных любого типа с ведомого устройства.
READ INPUT REGISTERS	04	WORD	Считывание нескольких переменных любого типа с ведомого устройства.
WRITE SINGLE COIL	05	BOOL	Запись отдельного сигнала (BOOL) в ведомое устройство.
WRITE SINGLE REGISTER	06	WORD	Запись отдельного сигнала (WORD) в ведомое устройство.
WRITE MULTIPLE COILS	15	BOOL	Запись нескольких переменных (BOOL) в ведомое устройство.
WRITE MULTIPLE REGISTERS	16	WORD	Запись нескольких переменных любого типа в ведомое устройство.
READ WRITE HOLDIN G REGISTERS	23	WORD	Считывание и запись нескольких переменных любого типа в ведомое устройство и с него.

Таблица 131: Функции Modbus

Чтобы создать новую телеграмму-запрос для ведомого устройства TCP/UDP, необходимо:

1. В структуре **Resource, Protocols, Modbus Master, Ethernet Slaves** выбрать **TCP/UDP Slave**.
2. Щелкнуть правой кнопкой мыши на **TCP/UDP Slave** и выбрать в контекстном меню **New**.
3. В диалоговом окне **New Object** выбрать **Request Telegram**.

Чтобы создать новую телеграмму-запрос для ведомого устройства шлюза, необходимо:

1. В структуре **Resource, Protocols, Modbus Master, Modbus Gateway** выбрать **Gateway Slave**.
2. Щелкнуть правой кнопкой мыши на **Gateway Slave** и выбрать в контекстном меню **New**.
3. В диалоговом окне **New Object** выбрать **Request Telegram**.

Чтобы создать новую телеграмму-запрос для ведомого устройства RS485 Modbus, необходимо:

1. В структуре **Resource, Protocols, Modbus Master, Serial Modbus** выбрать **Modbus Slave**.
2. Щелкнуть правой кнопкой мыши на **Modbus Slave** и выбрать в контекстном меню **New**.
3. В диалоговом окне **New Object** выбрать **Request Telegram**.

6.4.1 Телеграмма-запрос для считывания

С помощью функции считывания можно считывать переменные с ведомого устройства.

Телеграмма-запрос ведущего устройства Modbus содержит наряду с функцией Modbus начальный адрес области считывания/записи.

Для считывания переменных ведущее устройство Modbus передает *Read Request Telegram* на ведомое устройство Modbus.

Ведомое устройство Modbus отправляет телеграмму-ответ с запрашиваемыми переменными обратно на ведущее устройство Modbus.

Чтобы конфигурировать телеграмму-запрос для считывания, необходимо:

1. В структуре выбрать **Request Telegram** для конфигурирования.
2. Щелкнуть правой кнопкой мыши на **Request Telegram** и выбрать в контекстном меню **Edit**.
3. В **Object Panel** выбрать глобальную переменную, которая должна выступать в качестве переменной приема Modbus, и перетащить ее с помощью Drag&Drop в пустое место в области **Input Signals**.
4. Повторить данное действие для каждой последующей переменной приема Modbus.
5. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Input Signals**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Предлагаются следующие телеграммы *Read Request Telegrams*:

Read Coils (01)

Считывание нескольких переменных (BOOL) с ведомого устройства.

Элемент	Значение
Type	Функция Modbus Read Coils (01)
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the read area	0...65535

Таблица 132: Телеграмма-запрос Read Coils (01)

Read Discrete Inputs (02)

Считывание нескольких переменных (BOOL) с ведомого устройства.

Элемент	Значение
Type	Функция Modbus Read Discrete Inputs (02)
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the read area	0...65535

Таблица 133: Телеграмма-запрос Read Discrete Inputs (02)

Read Holding Registers (03)

Считывание нескольких переменных любого типа с ведомого устройства.

Элемент	Значение
Type	Функция Modbus Read Holding Registers (03)
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the read area	0...65535

Таблица 134: Телеграмма-запрос Read Holding Registers

Read Input Registers (04)

Считывание нескольких переменных любого типа с ведомого устройства

Элемент	Значение
Type	Функция Modbus Read Input Registers (04)
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the read area	0...65535

Таблица 135: Телеграмма-запрос Read Input Registers

Телеграмма-запрос для считывания и записи

Для считывания и записи переменных ведущее устройство Modbus передает *Read/Write Request Telegram* на ведомое устройство Modbus.

Сначала ведущее устройство Modbus записывает определенные переменные записи в определенную область импорта ведомого устройства Modbus.

Затем ведущее устройство Modbus считывает определенные переменные считывания в определенной области экспорта ведомого устройства Modbus.

i

Функции записи и считывания в *Read/Write Request Telegram* не зависят друг от друга, они просто передаются в общей телеграмме-запросе.

Однако *Read/Write Request Telegram* используется таким образом, чтобы переменные, записанные в ведущем устройстве, считывались обратно. Таким образом проверяется, правильно ли были записаны переданные переменные.

Чтобы конфигурировать телеграмму-запрос для считывания и записи, необходимо:

1. В структуре выбрать **Request Telegram** для конфигурирования.
2. Щелкнуть правой кнопкой мыши на **Request Telegram** и выбрать в контекстном меню **Edit**.

Чтобы конфигурировать переменные для чтения, необходимо:

1. Выберите в **Object Panel** глобальную переменную, которую Вы хотите соединить с новой переменной приема Modbus, и перетащите ее с помощью Drag&Drop в столбец **Global Variable** переменной приема Modbus.
2. Повторить данное действие для каждой последующей переменной приема Modbus.
3. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Input Signals**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Чтобы конфигурировать переменные для записи, необходимо:

1. Выберите в **Object Panel** глобальную переменную, которую Вы хотите соединить с новой переменной передачи Modbus и перетащите ее с помощью Drag&Drop в столбец **Global Variable** переменной передачи Modbus.
2. Повторить данное действие для каждой последующей переменной передачи Modbus.
3. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Output Signals**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Вкладка Read Write Holding (23)

Считывание и запись нескольких переменных любого типа в область импорта ведомого устройства и из нее.

Элемент	Значение
Type	Функция Modbus <i>Read Write Holding Registers (23)</i>
Name	Любое однозначное имя, для функции Modbus
Description	Описание функции Modbus
Start address of the read area	0...65535
Start address of the write area	0...65535

Таблица 136: Вкладка Read Write Holding

6.4.2 Телеграмма-запрос для записи

С помощью функций записи переменные записываются только в область импорта ведомого устройства.

Телеграмма-запрос ведущего устройства Modbus содержит наряду с функцией Modbus начальный адрес области считывания/записи.

Для записи переменных ведущее устройство Modbus передает *Write Request Telegram* на ведомое устройство Modbus.

Ведомое устройство Modbus записывает принятые переменные в свою область импорта.

В диалоговом окне *Variable Connections* телеграммы *Write Request Telegram* необходимо ввести переменные, которые записывает ведущее устройство Modbus для ведомого устройства Modbus.

Чтобы конфигурировать телеграмму-запрос для записи, необходимо:

1. В структуре выбрать **Request Telegram** для конфигурирования.
2. Щелкнуть правой кнопкой мыши на **Request Telegram** и выбрать в контекстном меню **Edit**.
3. В **Object Panel** выбрать глобальную переменную, которая должна выступать в качестве переменной передачи Modbus, и перетащить ее с помощью Drag&Drop в пустое место в области **Send Signals**.
4. Повторить данное действие для каждой последующей переменной передачи Modbus.
5. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Send Signals** и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

Предлагаются следующие телеграммы *Write Request telegrams*:

Write Multiple Coils (15)

Запись нескольких переменных (BOOL) в область импорта ведомого устройства.

Элемент	Значение
Type	Функция Modbus «Write Multiple Coils (15)»
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the write area	0...65535

Таблица 137: Телеграмма-запрос Write Multiple Coils (15)

Write Multiple Registers (16)

Запись нескольких переменных любого типа в область импорта ведомого устройства.

Элемент	Значение
Type	Функция Modbus «Write Multiple Registers (16)»
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the write area	0...65535

Таблица 138: Телеграмма-запрос Write Multiple Registers (16)

Write Single Coil (05)

Запись отдельных переменных (BOOL) в область импорта ведомого устройства.

Элемент	Значение
Type	Функция Modbus «Write Single Coil (05)»
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the write area	0...65535

Таблица 139: Телеграмма-запрос Write Single Coil (05)

Write Single Register (06)

Запись отдельных переменных (WORD) в область импорта ведомого устройства.

Элемент	Значение
Type	Функция Modbus «Write Single Register (06)»
Name	Любое однозначное имя для функции Modbus
Description	Описание функции Modbus
Start address of the write area	0...65535

Таблица 140: Телеграмма-запрос Write Single Register

6.4.3 Ведомые устройства Ethernet (ведомые устройства TCP/UDP)

Ведущее устройство Modbus может устанавливать связь с 64 ведомыми устройствами TCP/IP и 247 ведомыми устройствами UDP/IP.

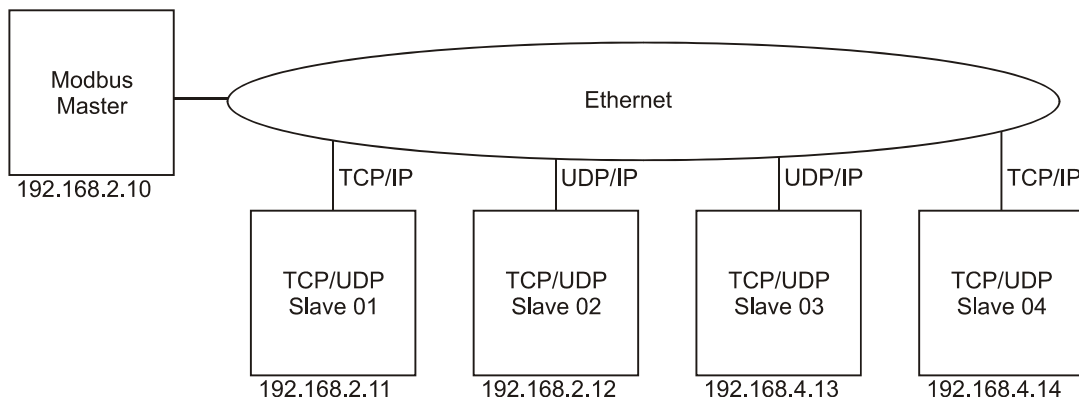


Рисунок 48: Сеть Modbus

Чтобы создать в ведущем устройстве Modbus новое соединение с ведомым устройством TCP/UDP:

1. В структуре открыть **Resource, Protocols, Modbus Master, Ethernet Slaves**.
2. Щелкнуть правой кнопкой мыши на ведомые устройства **Ethernet** и в контекстном меню выбрать **New**.
3. Выбрать из списка **TCP/UDP Slaves** и подтвердить с помощью **OK**.
4. Конфигурирование ведомого устройства TCP/UDP в ведущем устройстве Modbus:
Edit для присвоения системных переменных, см. главу 6.4.4.
Properties для конфигурирования свойств, см. главу 6.4.5.

i

Если ведомые устройства TCP/UDP и ведущее устройство Modbus находятся в различных подсетях, то в таблицу маршрутов необходимо внести соответствующие маршруты, определенные для конкретного пользователя.

6.4.4 Системные переменные ведомого устройства TCP/UDP

Вкладка *System Variables* содержит системные переменные, которые позволяют проанализировать и контролировать состояние ведомого устройства TCP/UDP в прикладной программе.

Состояние ведомого устройства TCP/UDP можно проанализировать в прикладной программе с помощью следующих переменных состояния:

Элемент	Описание
Modbus Slave activation control [BYTE]	Ведомое устройство TCP/UDP может быть активировано или деактивировано прикладной программой. 0: Активировать 1: Деактивировать
Modbus Slave Error [USINT]	Код ошибки Коды ошибки 0x01...0x0b соответствуют кодам исключительного условия спецификации протокола Modbus. 0x00: Нет ошибки Коды исключительного условия: 0x01: Недействительный код функции 0x02: Недействительная адресация 0x03: Недействительные данные 0x04: (не используется) 0x05: (не используется) 0x06: Устройство занято (только шлюз) 0x08: (не используется) 0x0a: (не используется) 0x0b: Нет ответа от ведомого устройства (только шлюз) Специальные коды NIMA: 0x10: Принят поврежденный кадр 0x11: Принят кадр с неверным ID процесса передачи 0x12: Принят неожиданный ответ 0x13: Принят ответ через неверное соединение 0x14: Неверный ответ на запрос записи 0xff: Время ожидания ведомого устройства
Modbus Slave State [BYTE]	Состояние соединения ведомого устройства TCP/UDP: 0: Деактивировано 1: Соединение отсутствует 2: Соединение установлено

Таблица 141: Системные переменные ведомого устройства TCP/UDP

6.4.5 Свойства ведомого устройства TCP/UDP

Для конфигурирования соединения с ведомым устройством TCP/UDP в ведущем устройстве Modbus необходимо настроить следующие параметры.

Элемент	Описание
Type	Ведомое устройство TCP/UDP
Name	Любое однозначное имя для ведомого устройства TCP/UDP
Description	Любое однозначное описание для ведомого устройства TCP/UDP
IP address	IP-адрес ведомого устройства TCP/UDP
Port	Стандарт: 502 Можно конфигурировать также другие порты TCP/UDP. При этом следует учитывать расположение портов в Internet Corporation for Assigned Names and Numbers (ICANN).
Type of communication IP protocol	TCP или UDP Значение по умолчанию: TCP
TCP connection only on demand	Если выбран протокол передачи TCP, то пользователь может настроить, должно ли автоматически прерываться соединение с данным ведомым устройством после каждого обмена данными. TRUE: Прервать соединение. FALSE: Не прерывать соединение. Значение по умолчанию: FALSE
Master-Slave Data Exchange [ms]	Интервал для обмена данными с данным ведомым устройством 1 до ($2^{31}-1$). Если ведомое устройство не достигается после <i>максимального количества попыток</i> , то интервал обмена данными между <i>ведущим и ведомым устройством</i> увеличивается в четыре раза.
Maximum Number of Resend	Максимальное количество повторов передачи, если ведомое устройство не отвечает. Количество повторов передачи можно настраивать на любое значение (0...65535). При TCP/IP всегда нуль, не изменяется. Рекомендуемое количество повторов составляет от нуля до восьми.
Receive Timeout [ms]	Время ожидания приема для данного ведомого устройства [мс]. По истечении данного времени снова выполняется попытка передачи.

Таблица 142: Параметр конфигурации

6.4.6 Шлюз Modbus (шлюз TCP/UDP)

Ведущее устройство Modbus может работать в качестве шлюза Modbus. В данном режиме запросы ведущего устройства, которые шлюз принимает через Ethernet, передаются на подключенные к шлюзу ведомые устройства RS485 и/или ведомые устройства Ethernet. Соответствующим образом ответы ведомых устройств передаются на ведущее устройство Modbus через шлюз.

Через последовательный интерфейс можно присвоить адрес до 121 последовательному ведомому устройству Modbus.

Диапазон адресов ведомого устройства составляет от 1 до 247. ведущему устройству Modbus 2 (шлюз Modbus) требуется лицензия ведущего устройства Modbus, даже если используется только шлюз Modbus.

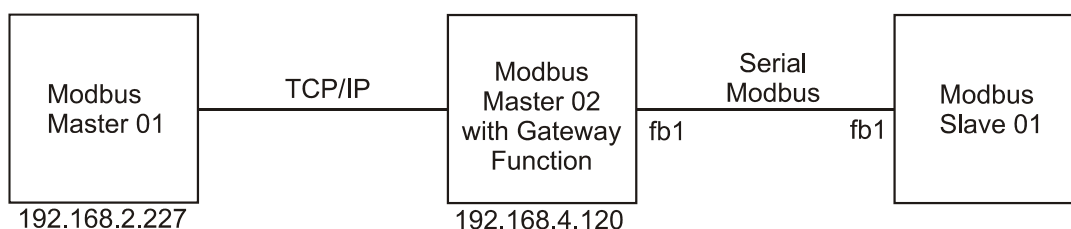


Рисунок 49: Шлюз Modbus

i

Если шлюз Modbus и ведущее устройство Modbus находятся в различных подсетях, то в таблицу маршрутов необходимо внести соответствующие маршруты, определенные для конкретного пользователя.

Ведущее устройство Modbus 1:

Чтобы создать в ведущем устройстве Modbus 1 соединение со шлюзом Modbus, необходимо:

1. В структуре открыть **Resource, Protocols, Modbus Master**.
2. Щелкнуть правой кнопкой мыши на **Modbus Master** и выбрать в контекстном меню **New**.
3. Выбрать из списка **Modbus Gateway** и подтвердить с помощью **OK**.
4. Конфигурирование шлюза Modbus в ведущем устройстве Modbus 1: **Properties** для конфигурирования свойств, см. главу 6.4.7.
В свойствах ввести **IP-адрес** ведущего устройства Modbus 2 (шлюз Modbus).

Чтобы создать в ведущем устройстве Modbus 1 соединение с ведомым устройством-шлюзом, необходимо:

В ведущем устройстве Modbus 1 необходимо создать последовательное ведомое устройство в качестве ведомого устройства-шлюза.

1. В структуре открыть **Resource, Protocols, Modbus Master, Modbus Gateway**.
2. Щелкнуть правой кнопкой мыши на шлюз **Modbus** и выбрать в контекстном меню **New**.
3. Выбрать из списка **Gateway Slave** и подтвердить с помощью **OK**.
4. Конфигурирование ведомого устройства шлюза в ведущем устройстве Modbus 1: **Edit** для присвоения системных переменных, см. главу 6.4.8.
Properties для конфигурирования свойств, см. главу 6.4.9.
В свойствах ведомого устройства внести **Serial Address** ведомого устройства-шлюза.

Чтобы определить в ведущем устройстве Modbus 1 входные и выходные переменные для последовательного ведомого устройства, необходимо:

1. Щелкнуть правой кнопкой мыши на **Gateway Slave** (ведомое устройство-шлюз) и выбрать в контекстном меню **New**.
2. Выбрать из списка необходимую **телеграмму-запрос**.
3. Щелкнуть правой кнопкой мыши на соответствующую **Request Telegram** и выбрать в контекстном меню **Edit**. Во вкладке *Process Variables* ввести входные или выходные переменные.

Ведущее устройство Modbus 2 (шлюз Modbus):

В свойствах ведущего устройства Modbus 02 должна быть активирована функция шлюза. Таким образом, сконфигурированные в ведущем устройстве 01 ведомые устройства-шлюзы соединены с последовательными ведомыми устройствами.

Чтобы активировать в ведущем устройстве Modbus 2 функцию шлюза, необходимо:

1. В структуре открыть **Resource, Protocols, Modbus Master**.
2. Щелкнуть правой кнопкой мыши на **Modbus Master** (ведущее устройство Modbus) и выбрать в контекстном меню **Properties**.
3. Включить параметр **Enable TCP Gateway**, чтобы ведущее устройство Modbus работало дополнительно в качестве шлюза TCP.
4. Включить параметр **Enable UDP Gateway**, чтобы ведущее устройство Modbus работало дополнительно в качестве шлюза UDP.

Чтобы конфигурировать в ведущем устройстве Modbus 2 последовательный Modbus:

1. В структуре открыть **Resource, Protocols, Modbus Master**.
2. Щелкнуть правой кнопкой мыши на **Modbus Master** и выбрать в контекстном меню **New**.
3. Выбрать из списка **Serial Modbus** и подтвердить с помощью **OK**.
4. Конфигурирование **Serial Modbus Properties** выбрать и ввести интерфейсы, скорость передачи в бодах и т.д.

Чтобы конфигурировать в ведущем устройстве Modbus 2 соединение с последовательным ведомым устройством, необходимо:

1. В структуре открыть **Resource, Protocols, Modbus Master, Serial Modbus**.
2. Щелкнуть правой кнопкой мыши на **Serial Modbus** и выбрать в контекстном меню **New**.
3. Выбрать из списка **Modbus Slave** и подтвердить с помощью **OK**.
4. Выбрать конфигурацию **Modbus Slave Properties** и ввести **Slave Address** последовательного ведомого устройства.

Последовательное ведомое устройство

Чтобы конфигурировать последовательное ведомое устройство Modbus, необходимо:

1. В структуре открыть **Resource, Protocols, Modbus Slave**.
2. Щелкнуть правой кнопкой мыши на **Modbus Slave** и выбрать в контекстном меню **Edit**.
3. Ввести конфигурацию **Modbus Slave Properties** и **Slave Address** последовательного ведомого устройства.

6.4.7 Свойства шлюза

Ведущее устройство Modbus осуществляет связь со своими ведомыми устройствами Modbus через шлюз Modbus.

Для конфигурирования соединения со шлюзом Modbus в ведущем устройстве Modbus должны быть настроены следующие параметры.

Элемент	Описание
Type	Шлюз Modbus
Name	Любое однозначное имя для шлюза
Description	Любое однозначное описание для ведомого устройства TCP/UDP
Communication IP Protocol	TCP или UDP Значение по умолчанию: TCP
IP address	IP-адрес шлюза, через который ведущее устройство Modbus должно осуществлять связь со своими ведомыми устройствами Modbus. Значение по умолчанию: (0.0.0.0)
Port	Значение по умолчанию: 502

Таблица 143: Параметры соединения шлюза Modbus

6.4.8 Системные переменные ведомого устройства-шлюза

В редакторе **Edit** имеется три переменных состояния:

Элемент	Описание
Modbus Slave activation control [BYTE]	Ведомое устройство-шлюз может быть активировано или деактивировано прикладной программой. 0: Активировать 1: Деактивировать
Modbus Slave Error [USINT]	Параметры, как у ведомого устройства TCP/UDP, см. главу 6.4.4.
Modbus Slave State [BYTE]	Состояние соединения ведомого устройства-шлюза: 0: Деактивировано 1: Соединение отсутствует 2: Соединение установлено

Таблица 144: Переменные состояния ведомого устройства-шлюза

6.4.9 Свойства шлюза ведомого устройства-шлюза

Для конфигурирования соединения с ведомым устройством-шлюзом в ведущем устройстве Modbus необходимо настроить следующие параметры.

Элемент	Описание
Type	Ведомое устройство-шлюз
Name	Любое однозначное имя для ведомого устройства-шлюза
Description	Любое однозначное описание для ведомого устройства-шлюза
Slave Address	1...247
Остальные параметры, как ведомого устройства TCP/UDP, см. главу 6.4.5	

Таблица 145: Параметры соединения ведомого устройства-шлюза

6.4.10 Последовательный Modbus

Ведущее устройство Modbus может устанавливать связь с 247 последовательными ведомыми устройствами. Согласно норме допускается всего три промежуточных усилителя, таким образом, возможен максимум 121 участник шины на один последовательный интерфейс ведущего устройства.

i

Назначение штырьковых выводов подсоединений D-SUB (fb1, fb2) модуля COM HIMax, см. главу 3.5.1.

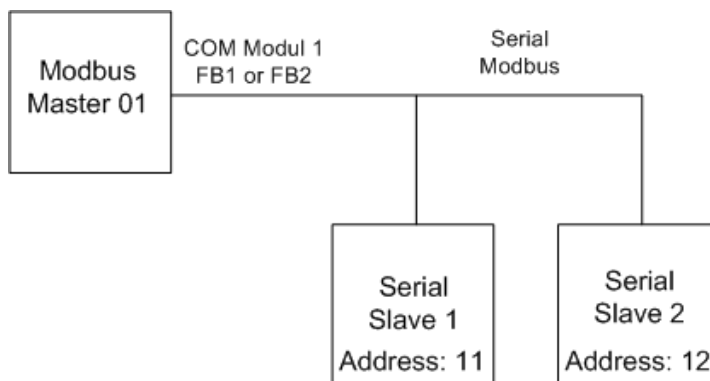


Рисунок 50: Последовательный Modbus

Ведущее устройство Modbus HIMA поддерживает передачу данных в формате RTU (Remote Terminal Unit).

Рама телеграммы RTU начинается и заканчивается заданными пользователем пустыми символами (Значение по умолчанию: 5 пустых символов).

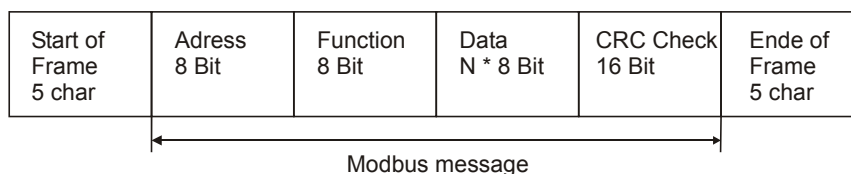


Рисунок 51: Телеграмма Modbus

Чтобы создать в ведущем устройстве Modbus последовательный Modbus, необходимо:

1. В структуре открыть **Resource, Protocols, Modbus Master, Serial Modbus**.
2. Щелкнуть правой кнопкой мыши на **Serial Modbus** и выбрать в контекстном меню **New**.
3. Выбрать из списка **Modbus Slave** и подтвердить с помощью **OK**.
4. Конфигурирование ведомого устройства Modbus в ведущем устройстве Modbus: **Edit** для присвоения системных переменных, см. 6.4.12. **Properties** для конфигурирования свойств, см. главу 6.4.13.

6.4.11 Свойства последовательного Modbus

Для конфигурирования последовательного Modbus в ведущем устройстве Modbus должны быть настроены следующие параметры.

Элемент	Описание
Type	Последовательный Modbus
Name	Имя последовательного Modbus выбирается пользователем
Description	Любое однозначное описание последовательного Modbus
Interface	Интерфейс полевой шины, используемый для ведущего устройства Modbus (fb1, fb2)
Baud rate [bps]	Скорость передачи для RS485 Возможные значения: 38400 бит/с 19200 бит/с 9600 бит/с 4800 бит/с 2400 бит/с 1200 бит/с 600 бит/с 300 бит/с Значение по умолчанию: 38400
Parity	Отсутствует Непрямой Прямой Значение по умолчанию: прямой
Stop Bits	Стандарт (регулирует число стоповых бит по четности: с четностью = 1 стоповый бит, без четности 2 стоповых бита). один стоповый бит два стоповых бита Значение по умолчанию: Стандарт
Number of idle chars	Число пустых символов в начале и в конце рамы телеграммы RTU. Диапазон значений: 0...65535 Значение по умолчанию: 5 символов

Таблица 146: Параметры последовательного ведущего устройства Modbus

6.4.12 Системные переменные ведомого устройства Modbus

В редакторе Edit имеется три переменных состояния (системные переменные).

Элемент	Описание
Modbus Slave activation control [BYTE]	Активировать или деактивировать ведомое устройство Modbus в прикладной программе. 0: Активировать 1: Деактивировать
Modbus Slave Error [USINT]	Параметры, как у ведомого устройства TCP/UDP, см. главу 6.4.5.
Modbus Slave State [BYTE]	Состояние соединения ведомого устройства Modbus: 0: Деактивировано 1: Соединение отсутствует 2: Соединение установлено

Таблица 147: Системные переменные ведомого устройства Modbus

6.4.13 Свойства ведомого устройства Modbus

Для конфигурирования соединения с последовательными ведомыми устройствами в ведущем устройстве Modbus должны быть настроены следующие параметры.

Элемент	Описание
Type	Ведомое устройство Modbus
Name	Имя ведомого устройства Modbus, выбирается пользователем
Description	Любое однозначное описание ведомого устройства Modbus
Slave Address	1...247
Остальные параметры, как ведомого устройства TCP/UDP, см. главу 6.4.5.	

Таблица 148: Параметры соединения ведущего устройства Modbus

i

Время ожидания приема при последовательном ведомом устройстве Modbus зависит от настроенной скорости передачи.

Если скорость передачи в бодах составляет 19200 [бит/с] или выше, то заданное значение может использоваться для времени ожидания приема. При скорости передачи в бодах ниже 19200 [бит/с] время ожидания приема необходимо увеличить.

6.5 Панель управления (ведущее устройство Modbus)

В панели управления пользователь может проверять и регулировать настройки соединения ведущего устройства Modbus. Для этого отображается актуальная информация по состоянию (напр., время цикла, состояние ведущего устройства и т.д.) ведущего устройства.

Чтобы открыть панель управления для контроля ведущего устройства Modbus, необходимо:

1. В структуре выбрать **Resource**.
2. В строке меню выбрать Online.
3. В **System Log-in** ввести данные доступа, чтобы открыть панель управления ресурса.
4. В структуре панели управления выбрать **Modbus Master**.

6.5.1 Контекстное меню (ведущее устройство Modbus)

В контекстном меню выбранного ведущего устройства Modbus можно выбрать следующие команды:

Offline

С помощью данной команды ведущее устройство Modbus останавливается.

Operate

С помощью данной команды ведущее устройство Modbus запускается.

Reset statistical data

Сбрасывает статистические данные (напр., количество ошибок шины, время цикла мин., макс., и т.д.) на нуль.

6.5.2 Поле индикации (ведущее устройство Modbus)

В поле индикации отображаются следующие значения выбранного ведущего устройства Modbus.

Элемент	Описание
Name	Имя ведущего устройства Modbus
Master State	Состояние ведущего устройства Modbus отображает текущее состояние протокола: OPERATE OFFLINE
Bus error count	Счетчик числа ошибок шины
Disturbed connections	Счетчик числа соединений со сбоями
CPU Load (planned)	См. главу 6.3.2
CPU Load (actual)	

Таблица 149: Поле индикации ведущего устройства Modbus

6.6 Панель управления (ведущее устройство > ведомое устройство Modbus)

В панели управления пользователь может проверять и активировать/деактивировать настройки участников коммуникации ведущего устройства Modbus. Для этого отображается актуальная информация по состоянию (напр., состояние ведомого устройства и т.д.) участника коммуникации.

Чтобы открыть панель управления для контроля соединения Modbus, необходимо:

1. В структуре выбрать **Resource**.
2. В **строке меню** выбрать Online.
3. В **System Log-in** ввести данные доступа, чтобы открыть панель управления ресурса.
4. В структуре панели управления выбрать **Modbus Master, Slave**.

6.7 Функция светодиода FBx у ведущего устройства Modbus

Модуль COM указывает состояние локального протокола ведущего устройства Modbus с помощью одного из светодиодов, присвоенных соответствующему интерфейсу полевой шины. Состояния данных светодиодов указаны в таблице ниже:

Светодиод FBx	Цвет	Описание
ВЫКЛ	Желтый	Протокол ведущего устройства Modbus не активен! То есть система управления находится в состоянии STOP или ведущее устройство Modbus не конфигурировано.
Мигает	Желтый	Протокол ведущего устройства Modbus активен и находится в состоянии обмена данными с ведомым устройством Modbus.
ВЫКЛ	Красный	Протокол ведущего устройства Modbus не имеет сбоев.
Мигает	Красный	Следующие события ведут к сбою. <ul style="list-style-type: none"> ▪ Принят неверный ответ или сообщение об ошибке от ведомого устройства ▪ Время ожидания для одного или нескольких ведомых устройств ▪ Превышение бюджета времени вычислений <p>Если через 5 секунд не возникает событие ошибки, то индикация переходит в состояние «Protocol not disturbed».</p>

Таблица 150: Светодиод FBx у ведущего устройства Modbus

6.8 Ведомое устройство Modbus HIMA

Ведомое устройство Modbus HIMA может одновременно через последовательный интерфейс (RS485) и через Ethernet (TCP/UDP) обслуживать несколько ведущих устройств Modbus.

Необходимое оснащение и требования к системе

Элемент	Описание
HIMA controller	HIMax с модулем COM
COM module	Если используется Modbus RTU, то модуль COM на используемых последовательных интерфейсах полевой шины (FB1 или FB2) должен быть оснащен опциональным submodule HIMA RS485. Распределение интерфейсов, см. главу 3.5.
Activation	Любую функцию ведомого устройства Modbus необходимо деблокировать отдельно, см. главу 3.5.2. Modbus Slave RTU Modbus Slave TCP

Таблица 151: Оснащение и требования к системе ведомого устройства Modbus HIMA

Ведомое устройство Modbus (Properties)

Элемент	Описание																									
Modbus Slave	Для каждого модуля COM можно конфигурировать ведомое устройство Modbus.																									
Number of masters	RTU: Ввиду устройства передачи RS485 только одно ведущее устройство Modbus может обращаться к ведомому устройству. TCP: Максимум 20 ведущих устройств могут обращаться к ведомому устройству.																									
Max. Size of Send data	128 кБ																									
Max. Size of Receive data	128 кБ																									
Display format of the MODBUS data	<p>Система управления HIMax использует формат Big Endian для передачи данных.</p> <p>Пример 32 бит данных (напр., DWORD, DIN T):</p> <table><tr><th>32 бит данных (hex)</th><th colspan="4">0x12345678</th></tr><tr><td>Смещение в памяти</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>Big Endian (HIMax)</td><td>12</td><td>34</td><td>56</td><td>78</td></tr><tr><td>Middle Endian (H51q)</td><td>56</td><td>78</td><td>12</td><td>34</td></tr><tr><td>Little Endian</td><td>78</td><td>56</td><td>34</td><td>12</td></tr></table>	32 бит данных (hex)	0x12345678				Смещение в памяти	0	1	2	3	Big Endian (HIMax)	12	34	56	78	Middle Endian (H51q)	56	78	12	34	Little Endian	78	56	34	12
32 бит данных (hex)	0x12345678																									
Смещение в памяти	0	1	2	3																						
Big Endian (HIMax)	12	34	56	78																						
Middle Endian (H51q)	56	78	12	34																						
Little Endian	78	56	34	12																						

Таблица 152: Свойства ведомого устройства Modbus

Чтобы создать новое ведомое устройство Modbus HIMA, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols**.
2. В контекстном меню протокола выбрать **New, Modbus Slave**, чтобы добавить новое ведомое устройство Modbus.
3. В контекстном меню ведущего устройства Modbus выбрать **Properties, General**.
4. Выбрать **Modul** и **Interfaces**.

6.9 Функции меню ведомого устройства Modbus HIMA

6.9.1 Свойства

Функция меню «Свойства» в контекстном меню ведущего устройства Modbus открывает диалоговое окно «Свойства». Диалоговое окно содержит следующие вкладки:

Общие положения

Во вкладке **General** настраиваются следующие параметры для ведомого устройства Modbus.

Элемент	Описание
Name	Имя ведомого устройства Modbus
Module	Выбор модуля COM, на котором обрабатывается данный протокол.
Use Max CPU Load	Активировано: Сохранить предельное значение нагрузки ЦПУ из поля <i>Max. CPU Load [%]</i> . Деактивировано: Не использовать предельное значение нагрузки ЦПУ для данного протокола.
Max. CPU Load [%]	Максимальная нагрузка ЦПУ модуля COM, которая может создаваться при обработке протокола. Диапазон значений: 1...100% Значение по умолчанию: 30%
Area for reading the function codes 1 and 3	Параметр определяет, из какой области данных должны считываться данные для кода функции 1 и 3. Диапазон значений: <ul style="list-style-type: none"> ▪ Область импорта ▪ Область экспорта (совместима с 51q)

Элемент	Описание
Area to Read function code 23	<p>Здесь пользователь может определить область ведомого устройства Modbus, из которой должен считываться код функции 23.</p> <p>Область импорта: Ведущее устройство имеет доступ к области импорта ведомого устройства для чтения и записи.</p> <p>Область экспорта: Ведущее устройство считывает данные из области экспорта ведомого устройства и записывает в область импорта ведомого устройства.</p> <p>Указание: Считывание и запись осуществляется за один цикл ЦПУ. То есть считанные данные были подготовлены в <u>последнем</u> цикле ЦПУ.</p>
Master monitoring time [ms]	<p>Время ожидания, в течение которого ведомое устройство должно получить от своего ведущего устройства минимум один запрос.</p> <p>Если в течение времени ожидания ведомое устройство не получает запрос, то «Состояние соединения ведущего устройства» устанавливается на «соединение отсутствует».</p> <p>Диапазон значений $1 \dots 2^{31} - 1$ [мс] Значение по умолчанию: 0=Выкл</p>
Исходные данные для времени ожидания ведущего устройства	<p>По истечении времени ожидания ведущего устройства «Connection Status» устанавливается на «not connected».</p> <p>В зависимости от конфигурации входные данные прикладной программы сбрасываются на предустановленные значения по умолчанию при использовании <i>Accept Initial Data</i> или при использовании <i>Retain Last Values</i> сохраняются последние значения.</p>
Режим адресации Modbus	<p>При адресации Modbus (доступе к данным) можно выбирать между следующими двумя различными режимами:</p> <p>Совместим с прежними версиями HIMatrix Адресация совместима с операционными системами COM HIMatrix до версии 11 (см. главу 6.9.8).</p> <p>Адресация через бит (1 бит) и регистр (16 бит) Соответствует стандартной адресации Modbus (см. 6.9.9).</p>

Таблица 153: Вкладка «Общее» для ведомого устройства Modbus HIMA

Смещения

Более подробная информация по смещениям, см. главу 6.9.10.

Последовательный интерфейс

Во вкладке **Serial Interface** настраиваются параметры последовательного интерфейса ведомого устройства Modbus.

i

Назначение штырьковых выводов подсоединений D-SUB (fb1, fb2) описано в техпаспортах соответствующих модулей HIMax.

Элемент	Описание
Interface	Выбор доступных интерфейсов полевых шин, которые могут использоваться для ведомого устройства Modbus (нет, fb1, fb2).
Slave Address	Адрес шины ведомого устройства Диапазон значений: 1...247
Baud rate [bps]	Скорость передачи для RS485 Возможные значения: 38400 бит/с 19200 бит/с 9600 бит/с 4800 бит/с 2400 бит/с 1200 бит/с 600 бит/с 300 бит/с
Parity	Диапазон значений: <ul style="list-style-type: none"> ▪ отсутствует ▪ не прямой ▪ прямой Значение по умолчанию: прямой
Stop Bits	Диапазон значений: <ul style="list-style-type: none"> ▪ Стандарт (регулирует число стоповых бит по четности: с четностью =1 стоповый бит, без четности =2 стоповых бита.) ▪ один стоповый бит ▪ два стоповых бита Значение по умолчанию: Стандарт
Number of idle chars	Число пустых символов в начале и в конце рамы телеграммы RTU. Диапазон значений: 0...65535 Значение по умолчанию: 5 символов

Таблица 154: Вкладка Serial Interface для ведомого устройства Modbus HIMA

Порты TCP и UDP

Во вкладке **TCP and UDP Ports** настраиваются параметры для портов TCP и UDP ведомого устройства Modbus.

Элемент	Описание
Enable TCP	Активировать соединение TCP/IP Значение по умолчанию: Деактивировано
TCP Port	Значение по умолчанию: 502
Maximum number of TCP Connections	Максимальное количество одновременно открытых соединений TCP в качестве сервера. Диапазон значений: 1...20 Значение по умолчанию: 3
UDP Enable	Активировать соединение UDP/IP Значение по умолчанию: Деактивировано
UDP Port	Значение по умолчанию: 502

Таблица 155: Вкладка TCP and UDP Ports для ведомого устройства Modbus HIMA

ЦПУ/COM

Заданные значения для параметров обеспечивают высокоскоростной обмен данными Modbus между модулем COM и модулем ЦПУ в системе управления HIMax.

Данные параметры необходимо изменять только в том случае, если снижение нагрузки COM и/или ЦПУ требуется для приложения, и это допустимо для процесса.

i

Изменение параметров рекомендуется доверять только опытному программисту.

Увеличение времени обновления COM и ЦПУ означает также, что увеличится фактическое время обновления данных Modbus. Необходимо проверять временные требования устройства.

Элемент	Описание
Refresh Rate [ms]	<p>Время обновления в миллисекундах, когда происходит обмен данными протокола между COM и ЦПУ.</p> <p>Если значение <i>Refresh Rate</i> равняется нулю или меньше, чем время цикла системы управления, то обмен данными осуществляется так быстро, насколько это возможно.</p> <p>Диапазон значений: $0 \dots (2^{31} - 1)$</p> <p>Значение по умолчанию: 0</p>
Within one cycle	<p>Активировано: Обмен всеми данными протокола с ЦПУ на COM в течение одного цикла ЦПУ.</p> <p>Деактивировано: Обмен всеми данными протокола с ЦПУ на COM, распределенный на несколько циклов ЦПУ по 900 байтов на направление. Таким образом, при необходимости можно снизить время цикла системы управления.</p> <p>Значение по умолчанию: Activated</p>

Таблица 156: Вкладка CPU/COM для ведомого устройства Modbus HIMAX

6.9.2 Системные переменные

Вкладка **System Variables** содержит системные переменные, которые позволяют проанализировать состояние ведомого устройства Modbus в прикладной программе и управлять ведомым устройством Modbus.

Элемент	Описание
Valid master requests [DWORD]	Количество действительных запросов ведущего устройства с момента последнего сброса всех счетчиков или включения.
Master requests [DWORD]	Количество всех запросов ведущего устройства с момента последнего сброса всех счетчиков или включения.
Master Request Timeout [DWORD]	Время ожидания, в течение которого ведомое устройство должно получить от своего ведущего устройства минимум один запрос. Если в течение времени ожидания ведомое устройство не получает запрос, то <i>Master Connection Status</i> устанавливается на 1, not connected . В зависимости от конфигурации входные данные прикладной программы сбрасываются на предустановленные значения по умолчанию или сохраняются последние значения.
Master connection state [BOOL]	0 = не контролируется (Master Request Timeout равно нулю) 1 = соединение отсутствует 2 = соединение установлено
Invalid Master requests [DWORD]	Количество недействительных запросов ведущего устройства с момента последнего сброса всех счетчиков или включения. Недействительными запросами считаются запросы, на которые ведомое устройство Modbus отправляет ведущему устройству Modbus ответ с кодом ошибки. Поврежденные пакеты, которые распознаются и отфильтровываются уже на уровне драйвера (ошибки кадрирования, ошибки при проверке с помощью циклического кода (CRC), ошибки длины), здесь не указаны; о них сообщается только при диагностике.
Reset all counters [BYTE]	С помощью данной системной переменной все счетчики в прикладной программе могут быть сброшены на нуль. Смена с 0 на 1 вызывает функцию сброса Значения > 1 обрабатываются как 1.

Таблица 157: Вкладка «Системные переменные» для ведомого устройства Modbus HIMA

6.9.3 Регистровая переменная

(вкладка доступа)

Во вкладку **Register Variables** необходимо ввести переменные, которым ведущее устройство присваивает 16-битный адрес (код функции 3, 4, 6, 16, 23).

6.9.4 Битовые переменные

(Доступ к битам либо флагам)

Во вкладку **Bit Variables** необходимо ввести переменные, которым ведущее устройство присваивает 1-битный адрес (код функции 1, 2, 5, 15).

6.9.5 Присвоение переменных передачи/приема

Во вкладке **Inputs** присваиваются все переменные, которые ведомое устройство Modbus получает от ведущего устройства Modbus.

Чтобы конфигурировать переменные передачи ведомого устройства Modbus, необходимо:

1. В структуре выбрать **Modbus Slave**, которое Вы хотите конфигурировать.
2. Щелкнуть правой кнопкой мыши на **Modbus Slave** и выбрать **Edit**.
3. Выбрать вкладку **Register Variables**.
4. В *Object Panel* выбрать **Variable** и перетащить с помощью Drag&Drop в область *Register Outputs*.
5. Повторить данное действие для каждой последующей переменной, которую Вы хотите определить в качестве переменной передачи для ведомого устройства Modbus.
6. Щелкнуть правой кнопкой мыши на область *Register Outputs* и выбрать **New Offsets**.

Чтобы конфигурировать переменные приема ведомого устройства Modbus, необходимо:

1. В структуре выбрать **Modbus Slave**, которое Вы хотите конфигурировать.
2. Щелкнуть правой кнопкой мыши на **Modbus Slave** и выбрать **Edit**.
3. Выбрать вкладку **Register Variables**.
4. В *Object Panel* выбрать **Variable** и перетащить с помощью Drag&Drop в область *Register Inputs*.
5. Повторить данное действие для каждой последующей переменной, которую Вы хотите определить в качестве переменной приема для ведомого устройства Modbus.
6. Щелкнуть правой кнопкой мыши на область *Register Inputs* и выбрать **New Offsets**.

6.9.6 Функции Modbus

Ведомое устройство Modbus HIMA поддерживает следующие функции Modbus.

Элемент	Код	Тип	Значение
READ COILS	01	BOOL	Считывание нескольких переменных (BOOL) из области импорта или экспорта ¹⁾ ведомого устройства. Максимальная длина данных процесса: 251 байт
READ DISCRETE INPUT	02	BOOL	Считывание нескольких переменных (BOOL) из области экспорта ведомого устройства. Максимальная длина данных процесса: 251 байт
READ HOLDING REGISTER	03	WORD	Считывание нескольких переменных любого типа из области импорта или экспорта ¹⁾ ведомого устройства. Максимальная длина данных процесса: 250 байтов
READ INPUT REGISTER	04	WORD	Считывание нескольких переменных любого типа из области экспорта ведомого устройства. Максимальная длина данных процесса: 250 байтов
WRITE SINGLE COIL	05	BOOL	Запись отдельного сигнала (BOOL) в область импорта ведомого устройства. Максимальная длина данных процесса: 1 байт
WRITE SINGLE REGISTER	06	WORD	Запись отдельного сигнала (WORD) в область импорта ведомого устройства. Максимальная длина данных процесса: 2 байта
Diagnostics	08	x	Только субкод 0: функция кольцевой проверки ведомого устройства
WRITE MULTIPLE COILS	15	BOOL	Запись нескольких переменных (BOOL) в область импорта ведомого устройства. Максимальная длина данных процесса: 247 байтов
WRITE MULTIPLE REGISTER	16	WORD	Запись нескольких переменных любого типа в область импорта ведомого устройства. Максимальная длина данных процесса: 246 байтов
READ WRITE MULTIPLE REGISTER	23	WORD	Считывание и запись нескольких переменных любого типа в область импорта или экспорта ведомого устройства и из них. Максимальная длина данных процесса: 242 байтов (телеграмма-запрос ведущего устройства Modbus) 250 байтов (ответ на ведущее устройство).
Read Device Identification	43	x	Передача идентификационных данных от ведомого устройства на ведущее устройство.

¹⁾ Область экспорта может быть выбрана только у ведомых устройств HIMA.

Таблица 158: Функции Modbus ведомого устройства Modbus HIMA

Коды функций 03, 04 и 16 поддерживают наряду с типом данных WORD (2 байта) также любые другие типы данных. Представление обоих параметров (начальный адрес, количество) запроса осуществляется следующим образом:

Начальный адрес обозначает индекс первых передаваемых переменных, в то время как количество определяет размер передаваемой области: $2 \times \text{Число байтов необходимо передать}$, если данная область заканчивается «ровно» на границе переменной.

Коды ошибок:

- Если ведущее устройство передает телеграмму с неизвестным кодом функции, то система управления отвечает с кодом ошибки 1 (Invalid Code).
- Если телеграмма ведущего устройства не совпадает с конфигурацией ведомого устройства Modbus (т. е., например, телеграмма-запрос не заканчивается ровно на границе переменной), то ведомое устройство отвечает с кодом ошибки 2 (Invalid Data).
- Если ведущее устройство передает телеграмму с неверными значениями (напр., поле длины), то ведомое устройство отвечает с кодом ошибки 3 (Invalid Value).

Коммуникация осуществляется только в состоянии RUN модуля COM. Запросы ведущего устройства во всех других рабочих состояниях модуля COM остаются без ответа.

Указание по функции Modbus: Read Device Identification (43)

Ведомое устройство Modbus HIMax передает идентификационные данные на ведущее устройство и поддерживает следующие идентификационные номера объекта:

Basic:

0x00 VendorName «HIMA Paul Hildebrandt GmbH»

0x01 ProductCode «<Modul Seriennummer>»

0x02 MajorMinorRevision «<COM Vx.y CRC>»

Regular:

0x03 VendorUrl «http://www.hima.de»

0x04 ProductName «HIMax»

0x05 ModelName «HIMax»

0x06 UserApplicationName «-----[S.R.S.]»

Extended:

0x80 пусто «-----»

0x81 пусто «-----»

0x82 пусто «-----»

0x83 пусто «-----»

0x84 пусто «-----»

0x85 пусто «-----»

0x86 CRC файла modbus.config «<0x234adcef>»

(файл конфигурации протокола ведомого устройства в файловой системе ЦПУ.

Для сравнения с данными в SILworX в Онлайн/сравнение версии).

Поддерживаются следующие идентификационные коды ReadDevice:

(1) Read Basic device identification (stream access)

(2) Read regular device identification (stream access)

(3) Read extended device identification (stream access)

(4) Read one specific identification object (individual access)

(Более подробную информацию по Modbus Вы найдете в спецификации «Application Protocol Specification» www.modbus.org)

6.9.7 Специальные коды функций HIMA:

Специальные коды функций HIMA соответствуют стандартным кодам функций Modbus. Единственной разницей является максимально допустимая длина данных процесса равная 1100 байтов и формат заголовка запроса и ответа:

Элемент	Код	Тип	Значение
Read Coils Extended	100 (0x64)	BOOL	Соответствует коду функции 01 Считывание нескольких переменных (BOOL) из области импорта или экспорта ¹⁾ ведомого устройства. Максимальная длина данных процесса: 1100 байтов.
Read Discrete Inputs Extended	101 (0x65)	BOOL	Соответствует коду функции 02 Считывание нескольких переменных (BOOL) из области экспорта ведомого устройства. Максимальная длина данных процесса: 1100 байтов.
Read Holding Registers Extended	102 (0x66)	WORD	Соответствует коду функции 03 Считывание нескольких переменных любого типа из области импорта или экспорта ¹⁾ ведомого устройства. Максимальная длина данных процесса: 1100 байтов.
Read Input Registers Extended	103 (0x67)	WORD	Соответствует коду функции 04 Считывание нескольких переменных любого типа из области экспорта ведомого устройства. Максимальная длина данных процесса: 1100 байтов.
Write Multiple Coils Extended	104 (0x68)	BOOL	Соответствует коду функции 15 Запись нескольких переменных (BOOL) в область импорта ведомого устройства. Максимальная длина данных процесса: 1100 байтов.
Write Multiple Registers Extended	105 (0x69)	WORD	Соответствует коду функции 16 Запись нескольких переменных любого типа в область импорта ведомого устройства. Максимальная длина данных процесса: 1100 байтов.
Read/Write Multiple Registers Extended	106 (0x6A)	WORD	Соответствует коду функции 23 Считывание и запись нескольких переменных любого типа в область импорта или экспорта ведомого устройства и из них. Максимальная длина данных процесса: 1100 байтов (телеграмма-запрос ведущего устройства Modbus) 1100 байтов (ответ на ведущее устройство).

Формат заголовка запроса и ответа

Заголовок запроса и ответа специальных функций Modbus HIMA создается следующим образом:

Код	Запрос	Ответ
100 (0x64)	1 байт Код функции 0x64 2 байта Начальный адрес 2 байта Число флагов 1...8800(0x2260)	1 байт Код функции 0x64 2 байта Число байтов = N N байтов Данные флага (8 флагов запаковываются в один байт)
101 (0x65)	1 байт Код функции 0x65 2 байта Начальный адрес 2 байта Число флагов 1...8800(0x226)	1 байт Код функции 0x65 2 байта Число байтов = N N байтов Данные флага (8 флагов запаковываются в один байт)
102 (0x66)	1 байт Код функции 0x66 2 байта Начальный адрес 2 байта Количество регистров 1...550(0x226)	1 байт Код функции 0x66 2 байта Число байтов = N N байтов Данные регистра
103 (0x67)	1 байт Код функции 0x67 2 байта Начальный адрес 2 байта Количество регистров 1...550(0x226)	1 байт Код функции 0x67 2 байта Число байтов = N N байтов Данные регистра
104 (0x68)	1 байт Код функции 0x68 2 байта Начальный адрес 2 байта Число флагов 1...8800(0x2260) 2 байта Число байтов = N N байтов Данные флага	1 байт Код функции 0x66 2 байта Начальный адрес 2 байта Число флагов 1...8800(0x2260)
105 (0x69)	1 байт Код функции 0x69 2 байта Начальный адрес 2 байта Количество регистров 1...550(0x226) 2 байта Число байтов = N N байтов Данные регистра	1 байт Код функции 0x69 2 байта Начальный адрес 2 байта Количество регистров 1...550(0x226)
106 (0x6A)	1 байт Код функции 0x6a 2 байта Начальный адрес для считывания 2 байта Количество регистров для чтения 1...550(0x226) 2 байта Начальный адрес для записи 2 байта Количество регистров для записи 1...550(0x226) 2 байта Число байтов для записи = N N байтов Данные регистра	1 байт Код функции 0x6a 2 байта Число байтов = N N байтов Данные регистра

6.9.8 Режим адресации Modbus (режим совместимости)

Данный режим адресации совместим с адресацией Modbus операционных систем COM HiMatrix до версии 11.

Режим совместимости гарантирует целостность переменных при доступе. Таким образом предотвращается считывание либо запись «полуцелых» чисел. Переменные типа BOOL не пакуются, т.е. каждая переменная типа BOOL сохраняется как BYTE.

Пример сохранения (формат Big Endian) переменных в режиме совместимости:

Register Address	0	Var1 (INT)	
	1	Var2 (SINT)	Var3 (SINT)
	2	Var4 (SINT)	Var5 (REAL)
	3	Var5 (REAL)	
	4	Var5 (REAL)	Var6 (BOOL)
	5	Var7 (BOOL)	Var8 (BOOL)

Рисунок 52: Сохранение переменных в режиме совместимости Modbus

Информация по назначению переменных передачи/приема, см. главу 6.9.5.

Доступ к переменным в режиме совместимости

- Переменные должны быть сортированы по столбцу Byte.Bit.
- Адрес переменной рассчитывается в режиме совместимости из индекса переменной (адрес переменной = индекс – 1).
- Общий размер запрошенных флаговых переменных указывается в BYTE. Функции Modbus Read/Write (multiple) Флаги 1, 2, 5, 15.
- Общий размер запрашиваемых регистровых переменных указывается в WORD. Функции Modbus Read/Write Registers 3, 4, 6, 16, 23.
- Столбцы «Byte.Bit, Register.Bit» и «Bit» в режиме совместимости не действительны.

Register Inputs							
	Name	Data type	Global Variable	Size [Bit]	Byte.Bit	Register.Bit	Bit
1	Variable_1	INT	Variable_1	16	0.0	0.0	0
2	Variable_2	SINT	Variable_2	8	0.0	0.8	0
3	Variable_3	SINT	Variable_3	8	0.8	0.8	0
4	Variable_4	SINT	Variable_4	8	0.0	0.8	0
5	Variable_5	REAL	Variable_5	32	0.0	0.0	0
6	Variable_6	BOOL	Variable_6	1	0.0	0.8	0
7	Variable_7	BOOL	Variable_7	1	0.8	0.8	0
8	Variable_8	BOOL	Variable_8	1	0.0	0.8	0

Рисунок 53: Индекс переменных

Заголовок телеграммы	Начальный адрес (индекс-1)	Число переменных	CRC
Slave Address Function 01	5	3	Check sum

Таблица 159: Считывание Var6 до Var8 посредством функции 01 Modbus

Заголовок телеграммы	HWB	NWB	HWB	NWB	CRC
Slave Address Function 01	Var6 (BOOL)	Var7 (BOOL)	Var8 (BOOL)	пусто	Check sum

Таблица 160: Телеграмма-ответ функции 01

6.9.9 Режим адресации Modbus (адресация через бит и регистр)

Данный режим адресации соответствует стандартной адресации Modbus и распознает только 2 типа данных:

- Булевы переменные/флаг (1 бит)
- Регистровые переменные (16 бит)

Смещение и длина передаваемых данных составляется из кратного 1 бита (дискретный ввод/флаг) либо 16 бит (регистры).

Данный режим адресации не гарантирует целостность переменных, т. е. при таком типе доступа могут считываться/записываться любые части переменных.

Переменные типа BOOL сохраняются запакованными, т. е. каждая переменная типа BOOL сохраняется как число бит в байте.

Пример сохранения (формат Big Endian) переменных в стандартном режиме Modbus:

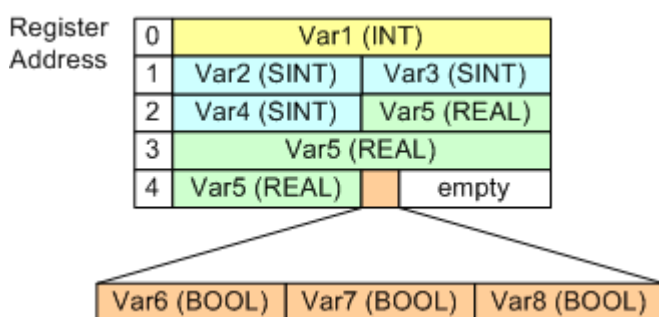


Рисунок 54: Сохранение переменных в стандартном режиме Modbus

Существует **Register Area** (входы и выходы) и **Bit Area** (входы и выходы). Обе области отделены друг от друга и могут принимать все допущенные типы данных. Различие состоит в разрешенных функциях Modbus для доступа к данным областям.

Register Area (регистровая область)

Переменные создаются во вкладке **Register Variables** в регистровой области. Более подробная информация по назначению переменных передачи/приема, см. главу 6.9.5.

Доступ к регистровой области осуществляется через функции Modbus Read/Write Registers 3, 4, 6, 16, 23. Для этого указывается адрес первой считываемой переменной и число регистров.

Чтобы посредством функций Modbus Read/Write (multiple) (флаги) 1, 2, 5, 15 иметь доступ к переменным, переменные должны зеркально отображаться в битовой области.

Адрес регистра	Адрес байта	Переменная	Размер [бит]	Байт.Бит	Регистр.Бит	Бит
0	0	Var1 (INT)	16	0.0	0.0	0
1	2 HWB	Var2 (SINT)	8	2.0	1.8	16
	3 NWB	Var3 (SINT)	8	3.0	1.0	24
2	4 HWB	Var4 (SINT)	8	4.0	2.8	32
	5 NWB	Var5 (REAL)	32	5.0	2.0	40
4	9	Var6 (BOOL)	1	9.0	4.0	72
		Var7 (BOOL)	1	9.1	4.1	73
		Var8 (BOOL)	1	9.2	4.2	74

Таблица 161: Сортированные переменные с адресацией

Заголовок телеграммы	Начальный адрес вкладки	Количество вкладок	CRC
Slave Address Function 03	2	3	Check sum

Таблица 162: Считывание переменных, начиная с Var4, с помощью функции 03

Заголовок телеграммы	HWB	NWB	HWB	NWB	HWB	NWB	CRC
Slave Address Function 03	Var4 (SINT)	Var5 (REAL)	Var5 (REAL)	Var5 (REAL)	Var5 (REAL)	Var6 до Var8 (BOOL)	Check sum

Таблица 163: Телеграмма-ответ функции 03

Битовая область

Переменные создаются во вкладке **Bit Variables** в битовой области.

Более подробная информация по назначению переменных передачи/приема, см. главу 6.9.5.

Доступ к битовой области осуществляется через функции Modbus Read/Write (multiple) (флаги) 1, 2, 5, 15. Для этого указывается адрес первой считываемой переменной и число бит.

Чтобы получить доступ с функцией Modbus Read/Write Registers 3, 4, 6, 16, 23 к переменным, битовые переменные зеркально отображаться в регистровой области.

Адрес регистра	Адрес байта	Тип данных	Размер [бит]	Байт.Бит	Регистр.Бит	Бит
0	0 HWB	Var10 (BOOL)	1	0.0	0.8	0
		Var11 (BOOL)	1	0.1	0.9	1
		Var12 (BOOL)	1	0.2	0.10	2
		Var13 (BOOL)	1	0.3	0.11	3
		Var14 (BOOL)	1	0.4	0.12	4
		Var15 (BOOL)	1	0.5	0.13	5
		Var16 (BOOL)	1	0.6	0.14	6
		Var17 (BOOL)	1	0.7	0.15	7
	1 NBW	Var18 (BOOL)	1	1.0	0.0	8
		Var19 (BOOL)	1	1.1	0.1	9
		Var20 (BOOL)	1	1.2	0.2	10
		Var21 (BOOL)	1	1.3	0.3	11
1	2	Var22 (WORD)	16	2.0	1.0	16
2	4	Var23 (BYTE)	8	4.0	2.8	32

Таблица 164: Битовые переменные с адресацией

Считывание значений Var14 до Var21 с помощью функции 01 Modbus:

Заголовок телеграммы	Начальный адрес бита	Число бит	CRC
Slave Address Function 01	4	8	Check sum

Таблица 165: Считывание переменных Var14 до Var21 с помощью функции 01

Телеграмма-ответ на функцию 01 Modbus:

Заголовок телеграммы	1 байт	CRC
Slave Address Function 01	Var14...Var21	Check sum

Таблица 166: Телеграмма-ответ функции 01

6.9.10 Смещения для альтернативной адресации Modbus

Чтобы получить доступ с функциями регистра к переменным в **Bit Area** и с функциями бита к переменным в **Register Area**, переменные должны зеркально отображаться в соответствующей другой области.

Во вкладку **Properties/Offsets** можно вносить смещения зеркально отображаемых переменных.

Альтернативный доступ можно активировать и деактивировать с помощью кнопки-флажка *Use alternative register/bit addressing*.

Элемент	Описание/диапазон значений
Использование альтернативной адресации регистров/бит	Активировано: Использовать альтернативную адресацию Деактивировано: Не использовать альтернативную адресацию Значение по умолчанию: Деактивировано
Регистровая область Смещение/ Битовые входы	0...65535
Регистровая область Смещение/ Битовые выходы	0...65535
Битовая область Смещение/ Регистровые входы	0...65535
Битовая область Смещение/ Регистровые выходы	0...65535

Таблица 167: Вкладка Смещение для ведомого устройства Modbus HIMA

i

Зеркально отображаемые переменные в битовой/регистровой области и имеющиеся переменные в битовой/регистровой области в отношении адресов Modbus не должны перекрывать друг на друга.

Доступ к отображаемым регистровым переменным в битовой области:

Доступ к отображаемым регистровым переменным возможен с помощью функций Modbus Read/Write (multiple) (флаги) 1, 2, 5, 15. Для этого необходимо указать адрес первой считываемой переменной и число бит.

Здесь находятся отображаемые регистровые переменные в битовой области, начиная с адреса байта 1000.

Адрес регистра	Адрес байта	Переменная	Размер [бит]	Байт.Бит	Регистр.Бит	Бит
500	1000	Var1 (INT)	16	1000.0	500.8	8000
501	1002 HWB	Var2 (SINT)	8	1002.0	501.8	8016
	1003 NWB	Var3 (SINT)	8	1003.0	501.0	8024
502	1004 HWB	Var4 (SINT)	8	1004.0	502.8	8032
	1005 NBW	Var5 (REAL)	32	1005.0	502.0	8040
504	1009	Var6 (BOOL)	1	1009.0	504.0	8072
		Var7 (BOOL)	1	1009.1	504.1	8073
		Var8 (BOOL)	1	1009.2	504.2	8074

Таблица 168: Регистровые переменные в битовой области с адресацией

Считывание Var6 до Var8 с помощью функции 01 Modbus:

Заголовок телеграммы	Начальный адрес бита	Число бит	CRC
Slave Address Function 01	8072	3	Check sum

Таблица 169: Считывание переменных Var6 до Var8 с помощью функции 01

Телеграмма-ответ на функцию 01 Modbus:

Заголовок телеграммы	1 байт	CRC
Slave Address Function 01	Var6...Var8 (BOOL)	Check sum

Таблица 170: Телеграмма-ответ на функцию 01

Доступ к отображаемым битовым переменным в регистровой области:

Доступ к отображаемой области битовых переменных возможен теперь посредством функций Modbus Read/Write Registers 3, 4, 6, 16, 23. Для этого необходимо указать адрес первой считываемой переменной и число регистров.

Здесь находятся отображаемые битовые переменные в регистровой области, начиная с адреса регистра 1000.

Адрес регистра	Адрес байта	Тип данных	Размер [бит]	Байт.Бит	Регистр.Бит	Бит
1000	2000 HWB	Var10 (BOOL)	1	2000.0	1000.8	16000
		Var11 (BOOL)	1	2000.1	1000.9	16001
		Var12 (BOOL)	1	2000.2	1000.10	16002
		Var13 (BOOL)	1	2000.3	1000.11	16003
		Var14 (BOOL)	1	2000.4	1000.12	16004
		Var15 (BOOL)	1	2000.5	1000.13	16005
		Var16 (BOOL)	1	2000.6	1000.14	16006
		Var17 (BOOL)	1	2000.7	1000.15	16007
	2001 NWB	Var18 (BOOL)	1	2001.0	1000.0	16008
		Var19 (BOOL)	1	2001.1	1000.1	16009
		Var20 (BOOL)	1	2001.2	1000.2	16010
		Var21 (BOOL)	1	2001.3	1000.3	16011
1001	2002	Var22 (WORD)	16	2002.0	1001.0	16016
1002	2004	Var23 (BYTE)	8	2004.0	1002.8	16032

Таблица 171: Отображаемые битовые переменные в регистровой области

Считывание Var22 до Var23 с помощью функции 03 Modbus:

Заголовок телеграммы	Начальный адрес Вкладка	Количество вкладок	CRC
Slave Address Function 03	1001	2	Check sum

Таблица 172: Считывание переменных Var22 до Var23 с помощью функции 03

Телеграмма-ответ функции 03 Modbus

Заголовок телеграммы	HWB	NWB	HWB	NWB	CRC
Slave Address Function 03	Var22 (WORD)	Var22 (WORD)	Var23 (BYTE)	пусто	Check sum

Таблица 173: Телеграмма-ответ на функцию 03

6.10 Панель управления (ведомое устройство Modbus)

В панели управления пользователь может проверять и регулировать настройки соединения ведомого устройства Modbus. Для этого отображается актуальная информация по состоянию (напр., состояние ведущего устройства и т.д.) ведомого устройства.

Чтобы открыть панель управления для контроля ведомого устройства Modbus, необходимо:

1. В структуре выбрать **Resource**.
2. В строке меню выбрать Online.
3. В **System Log-in** ввести данные доступа, чтобы открыть панель управления ресурса.
4. В структуре панели управления выбрать **Modbus Slave**.

6.10.1 Контекстное меню (ведомое устройство Modbus)

В контекстном меню выбранного ведомого устройства Modbus можно выбрать следующие команды:

Сброс статистических значений

Сбрасывает статистические данные (напр., время цикла мин., макс., и т.д.) на нуль.

6.10.2 Поле индикации (ведомое устройство Modbus)

В поле индикации отображаются следующие значения выбранного ведомого устройства Modbus.

Элемент	Описание
Name	Имя ведомого устройства Modbus
CPU Load (planned) [%]	См. главу 6.9.1
CPU Load (actual) [%]	

Таблица 174: Поле индикации ведомого устройства Modbus

6.10.3 Поле индикации (данные ведущего устройства)

В поле индикации «Данные ведущего устройства» отображаются следующие значения.

Элемент	Описание
Name	Имя данных ведущего устройства.
Interface	Совместимый с Modbus интерфейс модуля COM (RS485, Ethernet UDP/TCP).
Requests	Количество всех запросов ведущего устройства с момента последнего сброса счетчиков.
Valid requests	Количество действительных запросов ведущего устройства с момента последнего сброса счетчиков.
Invalid requests	Количество недействительных запросов ведущего устройства с момента последнего сброса счетчиков.
Timeout [ms]	Время ожидания, в течение которого ведомое устройство должно получить от своего ведущего устройства минимум один запрос. Если в течение времени ожидания ведомое устройство не получает запрос, то состояние <i>Master Connection</i> устанавливается на not connected.
Connection State	0 = не контролируется (время ожидания ведущего устройства равно нулю) 1 = соединение отсутствует 2 = соединение установлено

Таблица 175: Поле индикации данных ведущего устройства

6.11 Функция светодиода FBx у ведомого устройства Modbus

Модуль COM указывает состояние локального протокола ведомого устройства Modbus с помощью одного из светодиодов, присвоенных соответствующему интерфейсу полевой шины. Состояния данных светодиодов указаны в таблице ниже:

Светодиод FBx	Цвет	Описание
ВЫКЛ	Желтый	Протокол ведомого устройства Modbus не активен! То есть система управления находится в состоянии STOP или ведущее устройство Modbus не конфигурировано.
Мигает	Желтый	Протокол ведомого устройства Modbus активен и находится в состоянии обмена данными с ведущим устройством Modbus.
ВЫКЛ	Красный	Протокол ведомого устройства Modbus не имеет сбоев.
Мигает	Красный	Протокол ведомого устройства Modbus имеет сбой. Следующие события ведут к сбою. <ul style="list-style-type: none"> ▪ Принят неизвестный код функции ▪ Принят запрос с неверной адресацией ▪ Превышение бюджета времени вычислений <p>Если через 5 секунд не возникает событие ошибки, то индикация переходит в состояние <i>Protocol not disturbed</i>.</p>

Таблица 176: Светодиод FBx у ведомого устройства Modbus

6.11.1 Коды ошибок соединения TCP/IP Modbus

Коды ошибок соединения TCP/IP Modbus отображаются в диалоговом окне **Diagnose**.

Код ошибки	Описание
35	Действие заблокировано.
48	Порт уже используется.
50	Сеть не работает.
53	Соединение прервано программой.
54	Соединение прервано участником коммуникации.
55	Недостаточно буферной памяти.
60	Возникло время ожидания Соединение закрыто.
61	Соединение отклонено (участником коммуникации).
65	Отсутствует запись маршрута к участнику коммуникации.

Таблица 177: Коды ошибок TCP/IP Modbus

7 Send & Receive TCP

S&R TCP представляет собой независимый от изготовителя стандартный протокол для циклического и ациклического обмена данными и кроме TCP/IP не использует других специальных протоколов.

С помощью протокола S&R-TCP система управления HIMax поддерживает практически любую стороннюю систему, а также ПК с имеющимся интерфейсом сокетов (напр., Winsock.dll) для TCP/IP.

S&R TCP совместим с интерфейсом SEND/RECEIVE Siemens и позволяет осуществлять коммуникацию с системами управления Siemens через TCP/IP. Передача данных осуществляется через функциональные блоки S7 AG_SEND (FC5) и AG_RECV (FC6).

7.1 Требования к системе

Необходимое оснащение и требования к системе:

Элемент	Описание
Controller	HIMax с модулем COM
Activation	Активация осуществляется с помощью кода активации программы, см. главу 3.5.2.

Таблица 178: Требования к системе и оснащение S&R TCP

Свойства протокола S&R-TCP:

Элемент	Описание
S&R-TCP protocol	Для каждого модуля COM можно конфигурировать один протокол S&R-TCP.
Safety-related	Нет
Interface	Ethernet 10/100/1000BaseT
Data Exchange	Циклический и ациклический обмен данными через TCP/IP.
Function blocks	Функциональные блоки S&R-TCP должны использоваться при ациклическом обмене данными.
TCP Connections	В системе управления можно конфигурировать до 32 соединений TCP, если максимальный размер передаваемых и принимаемых данных не превышен.
Max. size of process data	<p>Всего можно передавать 128 кБ данных и принимать 128 кБ данных.</p> <p>1 Чтобы определить максимальное значение полезной нагрузки, необходимо все переменные состояния используемых соединений TCP и функциональных блоков TCP/SR отнять от максимального числа передаваемых данных (128 кБ). Распределение по отдельным соединениям TCP осуществляется произвольно.</p>

Таблица 179: Свойства S&R TCP

7.1.1 Создание протокола S&R-TCP

Чтобы создать новый протокол S&R-TCP, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols**.
2. В контекстном меню протокола выбрать **New, Send/Receive over TCP**, чтобы добавить новый протокол S&R-TCP.
3. В контекстном меню протокола Send/Receive-over-TCP **Properties, General** выбрать **COM-Module**.

7.2 Пример: Конфигурация S&R TCP

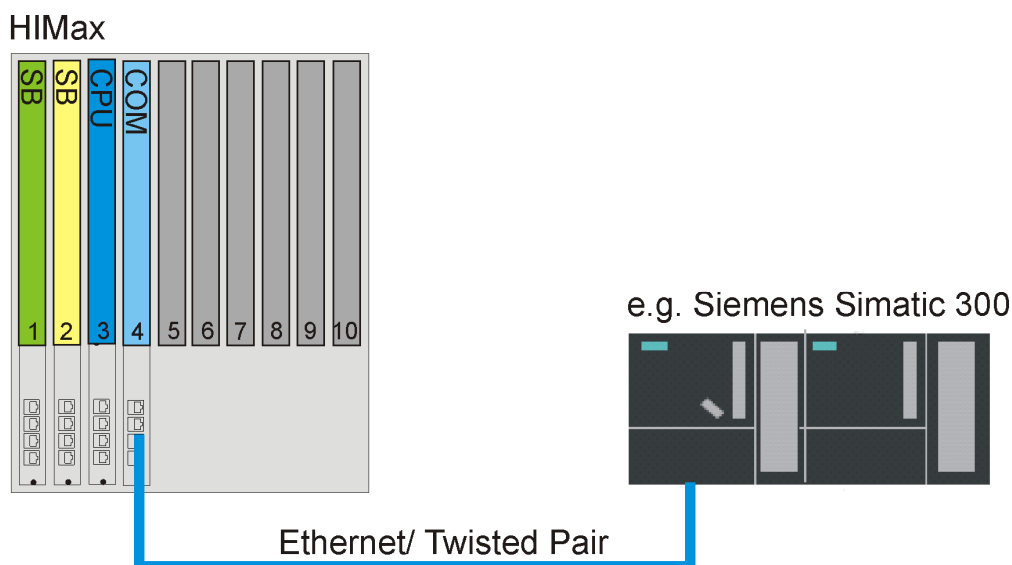


Рисунок 55: Соединение H1Max и системы управления Siemens

На данном примере протокол Send/Receive over TCP установлен в систему управления H1Max. H1Max должна осуществлять циклическую коммуникацию через S&R TCP с системой управления Siemens (напр., SIMATIC 300).

При этом H1Max (клиент) является активной станцией, которая создает соединение TCP с пассивным Siemens SIMATIC 300 (сервер). После установки соединения обе системы управления становятся равноправными и могут в любое время передавать и принимать данные.

При соединении H1Max и Siemens SIMATIC 300 необходимо учитывать следующее:

Для H1Max действуют описанные в разделе 7.1 (Требования к системе) требования.

H1Max и Siemens SIMATIC 300 соединяются друг с другом через интерфейсы Ethernet.

H1Max и Siemens SIMATIC 300 должны находиться в одной и той же подсети или при использовании маршрутизатора иметь соответствующие записи маршрута.

На данном примере система управления H1Max должна передать на Siemens SIMATIC 300 два BYTES и одно WORD. Переменные принимаются в SIMATIC 300 от блока AG_RECV (FC 6) и внутри передаются на блок AG_SEND (FC 5). Через блок AG_SEND (FC 5) система SIMATIC 300 передает переменные обратно в неизменном виде в систему управления H1Max.

После конфигурации пользователь может проверить передачу данных с помощью редактора инициализации HIMA.

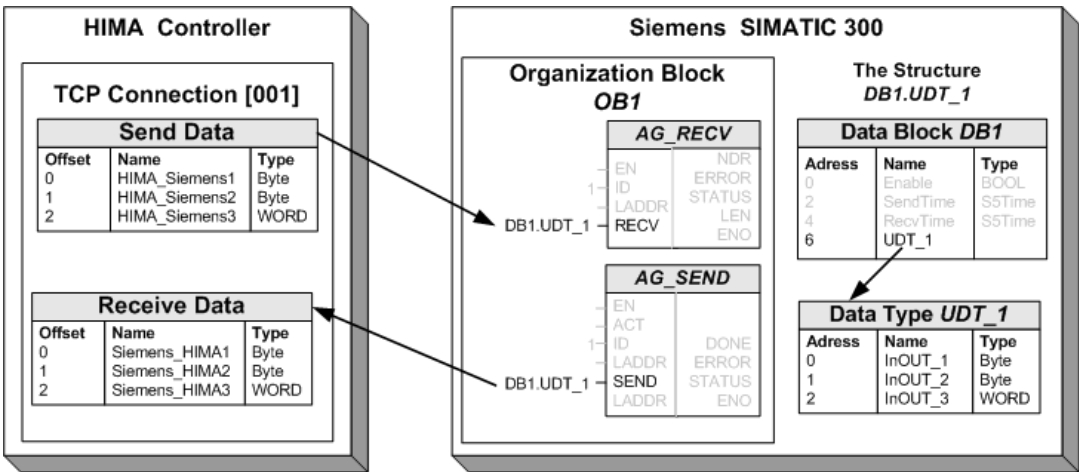


Рисунок 56: Передача данных между HIMaх и системой управления Siemens

Описание конфигурации системы управления HIMaх:

Элемент	Описание
TCP connection [001]	В данном диалоговом окне находятся все параметры, которые необходимы для связи с участником коммуникации (Siemens SIMATIC 300).
Send data	Смещения и типы переменных в системе управления должны совпадать с адресом и типами переменных в типе данных <i>UDT_1</i> системы SIMATIC 300.
Receive data	Смещения и типы переменных в системе управления HIMaх должны совпадать с адресом и типами переменных в типе данных <i>UDT_1</i> системы SIMATIC 300.

Таблица 180: Конфигурация системы управления HIMaх

Описание конфигурации Siemens SIMATIC 300:

Элемент	Описание
Organization block <i>OB1</i>	Функциональные блоки <i>AG_RECV</i> (FC6) и <i>AG_SEND</i> (FC 5) должны создаваться и конфигурироваться в организационном блоке <i>OB1</i> .
<i>AG_RECV</i> (FC 6)	Функциональный блок <i>AG_RECV</i> (FC 6) сохраняет принятые от участника коммуникации данные в тип данных <i>DB1.UDT_1</i> . <i>ID</i> входов и <i>LADDR</i> должны быть конфигурированы соответствующим образом для связи с участником коммуникации.
<i>AG_SEND</i> (FC 5)	Функциональный блок <i>AG_SEND</i> (FC 5) передает данные из типа данных <i>DB1.UDT_1</i> участнику коммуникации. <i>ID</i> входов и <i>LADDR</i> должны быть конфигурированы соответствующим образом для связи с участником коммуникации.
Data block <i>DB1</i>	Тип данных <i>UDT_1</i> определяется в блоке данных <i>DB1</i> .
Data type <i>UDT_1</i>	Адреса и типы переменных в SIMATIC 300 должны совпадать со смещениями и типами системы управления. Тип данных <i>UDT_1</i> получает принятую полезную нагрузку и сохраняет ее для передачи участнику коммуникации.

Таблица 181: Конфигурация Siemens SIMATIC 300

7.2.1 Конфигурация S&R TCP системы управления SIMATIC 300 Siemens

i

Следующие указания по конфигурированию системы управления Siemens могут быть неполными.

Все данные предоставляются без гарантии, при разработке проектов системы управления Siemens следует пользоваться документацией компании Siemens.

Чтобы создать сервер S&R-TCP в проекте системы SIMATIC 300, необходимо:

1. Запустить устройство управления SIMATIC Manager.
2. В SIMATIC Manager открыть проект системы управления SIMATIC 300.
3. В данном проекте создать и конфигурировать *Industrial Ethernet* и *MPI*.

Чтобы создать тип данных UDT1 с помощью следующих переменных, необходимо:

1. Открыть *Order Function Blocks* в Siemens SIMATIC Manager.
2. В главном меню открыть **Add, S7 Block, Data Type** и создать тип данных.
3. Присвоить типу данных имя **UDT1**.
4. Присвоить типу данных символическое имя **UDT_1**.
5. В типе данных *UDT_1* создать три переменных **InOut_x**, как показано на рисунки ниже.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	InOut_1	BYTE	B#16#0	
+1.0	InOut_2	BYTE	B#16#0	
+2.0	InOut_3	WORD	W#16#0	
=4.0		END_STRUCT		

Рисунок 57: Список переменных в блоке UDT1 Siemens

i

При циклическом и ациклическом обмене данными следует обращать внимание на то, что некоторые системы управления (напр., SIMATIC 300) вводят так называемые *pad bytes* (холостые байты). Таким образом обеспечивается, что все типы данных, размер которых больше одного байта, всегда начинаются с четного смещения и чтобы общая длина всех определенных переменных тоже всегда была четной.

В таких случаях с системе управления HiMax в соответствующих местах необходимо ввести фиктивные байты (см. главу 7.6, Сторонние системы с холостыми байтами).

Чтобы создать блок данных DB1 для функциональных блоков FC 5 и FC 6, необходимо:

1. В главном меню выбрать **Add, S7 Block, Data Block** и создать блок данных.
2. Присвоить блоку данных имя **DB1**.
3. Присвоить блоку данных символическое имя **DB1**.
4. Присвоить блоку данных DB1 тип данных **UDT_1**.
5. В блоке данных DB1 параметризовать типы данных, как показано на рисунке ниже.

Adresse	Name	Typ	Anfangswert
0.0		STRUCT	
+0.0	Enable	BOOL	FALSE
+2.0	SendTime	S5TIME	S5T#100MS
+4.0	RecvTime	S5TIME	S5T#10MS
+6.0	UDT_1	"UDT_1"	
=10.0		END_STRUCT	

Рисунок 58: Список переменных в блоке DB1 Siemens

Чтобы создать в редакторе символов следующие символы

1. Открыть диалоговое окно *KOP/AWL/FUP*, щелкнув правой кнопкой мыши на организационный блок **OB1**.
2. В главном меню **Extras, Symbol Table** открыть редуктор символов.
3. Дополнить *Symbol Editor* переменными **M 1.0 ... MW 5**, как показано на рисунке ниже.

S7-Programm(1) (Symbole) -- S7_Pro5\SIMATIC 300-Stat				
	Status	Symbol ▲	Adresse	Datentyp
1		Cycle Execution	OB 1	OB 1
2		DB1	DB 1	DB 1
3		UDT_1	UDT 1	UDT 1
4		RecDone	M 1.0	BOOL
5		RecError	M 1.1	BOOL
6		SendDone	M 1.2	BOOL
7		SendError	M 1.3	BOOL
8		RecStatus	MW 1	WORD
9		RecLen	MW 3	INT
10		SendStatus	MW 5	WORD
11				

Рисунок 59: Редактор символов SIMATIC

Чтобы создать функциональный блок **AG_RECV** (FC 6), необходимо:

1. Открыть диалоговое окно *KOP/AWL/FUP*.
2. Выбрать следующие функциональные блоки по порядку из структуры в левой части устройства управления Symatic-Manager:
one OR gate
one S_VIMP
one AG_RECV (FC 6).
3. Перетащить данные *Function Blocks* с помощью Drag&Drop в организационный блок *OB1*.
4. Соединить и конфигурировать *Function Blocks*, как показано на рисунке ниже.
5. Щелкнуть правой кнопкой мыши на функциональный блок **AG_RECV** (FC 6) и выбрать **Properties**.
6. Деактивировать **Active Connection Setup** и конфигурировать порты.
7. Записать параметры блока *LADDR* и внести их в структурную схему в блоке *AG_RECV* (FC 6).

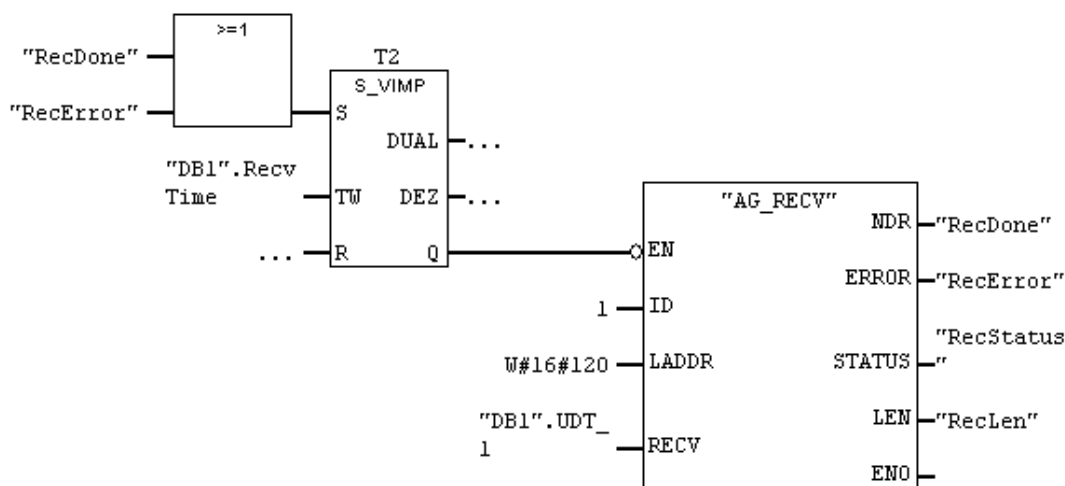


Рисунок 60: Структурная схема для приема

Чтобы создать функциональный блок **AG_SEND (FC 5)**, необходимо:

1. Открыть диалоговое окно *KOP/AWL/FUP*.
2. Выбрать следующие функциональные блоки по порядку из структуры в левой части устройства управления Symatic-Manager:
one OR gate
one S_VIMP
one AG_SEND (FC 5).
3. Перетащить данные *Function Blocks* с помощью Drag&Drop в организационный блок *OB1*.
4. Соединить и конфигурировать *Function Blocks*, как показано на рисунке ниже.
5. Щелкнуть правой кнопкой мыши на функциональный блок **AG_SEND (FC 5)** и выбрать **Properties**.
6. Деактивировать **Active Connection Setup** и конфигурировать порты.
7. Записать параметры блока *LADDR* и внести их в структурную схему в блоке *AG_SEND (FC 5)*.

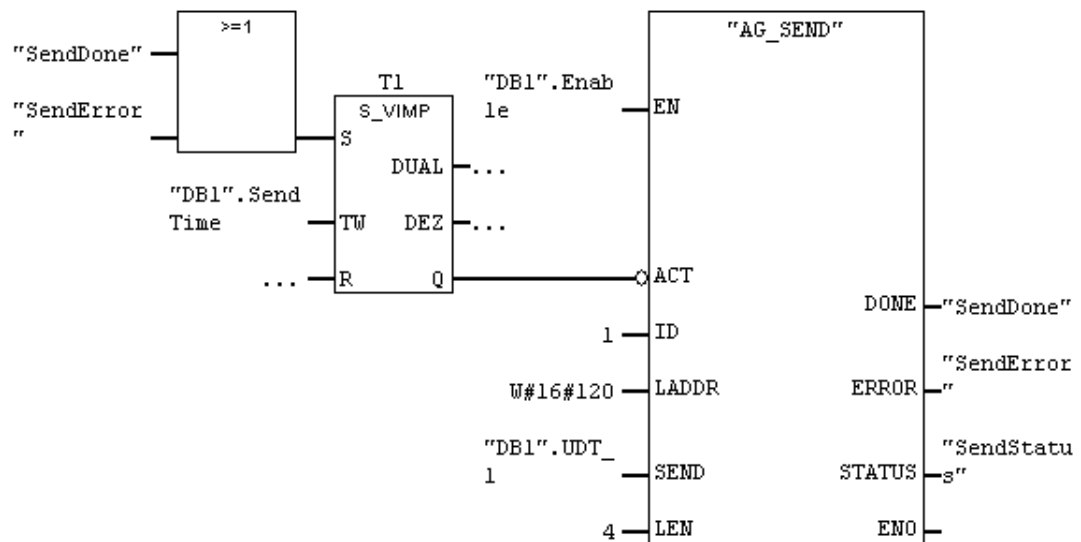


Рисунок 61: Структурная схема для передачи

Чтобы загрузить код в систему управления **SIMATIC 300**, необходимо:

1. Запустить **Code Generator** для программы.
2. Убедиться, что коды сгенерированы без ошибок.
3. Загрузить код в систему управления **SIMATIC 300**.

7.2.2 S&R TCP Конфигурирование системы управления HIMAх

Для конфигурирования систем управления HIMAх и использования средств программирования SILworX рекомендуется воспользоваться руководством «Первые шаги SILworX».

Чтобы создать следующие глобальные переменные в редакторе переменных, необходимо:

1. В структуре выбрать **Configuration, Global Variables**.
2. Щелкнуть правой кнопкой мыши на **Global Variables** и выбрать **Edit**.
3. Создать глобальные переменные, как показано в таблице, см. таблица 182.

Название	Тип
Siemens_HIMA1	Byte
Siemens_HIMA2	Byte
Siemens_HIMA3	WORD
HIMA_Siemens1	Byte
HIMA_Siemens2	Byte
HIMA_Siemens3	WORD

Таблица 182: Глобальные переменные

Чтобы создать протокол S&R-TCP в ресурсе, необходимо:

1. В структуре открыть **Configuration, Resource**.
2. Щелкнуть правой кнопкой мыши на **Protocols** и выбрать в контекстном меню **New**.
3. Выбрать **Send/Receive over TCP** и ввести имя для протокола.
4. Подтвердить с помощью OK, чтобы создать новый протокол.
5. Щелкнуть правой кнопкой мыши на **Send/Receive over TCP** и выбрать в контекстном меню **Properties**.
6. Выбрать **COM Modul**. Остальные параметры сохраняют значения по умолчанию.

Чтобы создать соединение TCP, необходимо:

1. Щелкнуть правой кнопкой мыши на **Send/Receive over TCP** и выбрать в контекстном меню **New**.
2. Щелкнуть правой кнопкой мыши на **TCP Connection** и выбрать в контекстном меню **Properties**.
3. Конфигурировать свойства, как показано на рисунке ниже.

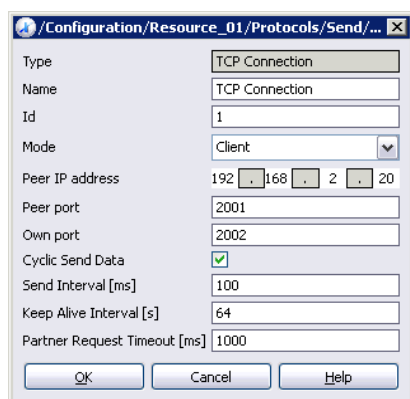


Рисунок 62: Свойства соединения TCP в SILworX

i

Если необходимо параметризовать циклический обмен данными между двумя системами управления, то в диалоговом окне *Properties* соединения TCP должна быть активирована опция *Cyclic Data Transfer*.

Чтобы конфигурировать принимаемые данные системы управления HIMAх, необходимо:

1. Щелкнуть правой кнопкой мыши на **TCP Connection** и выбрать в контекстном меню **Edit**.
2. Выбрать вкладку **Process Variables**.
3. В **Object Panel** выбрать следующие глобальные переменные и перетащить их с помощью Drag&Drop в область **Input Signals**.

Глобальная переменная	Тип
Siemens_HIMA1	Byte
Siemens_HIMA2	Byte
Siemens_HIMA3	WORD

Таблица 183: Переменные для принимаемых данных

4. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Register Inputs**, и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

i

Обращайте внимание на то, что смещения переменных в системе управления HIMAх должны совпадать с адресами переменных в типе данных *UDT_1* системы SIMATIC 300.

Чтобы конфигурировать передаваемые данные системы управления HIMAх, необходимо:

1. Щелкнуть правой кнопкой мыши на **TCP Connection** и выбрать в контекстном меню **Edit**.
2. Выбрать вкладку **Process Variables**.
3. В **Object Panel** выбрать следующие глобальные переменные и перетащить их с помощью Drag&Drop в область **Input Signals**.

Глобальная переменная	Тип
HIMA_Siemens1	Byte
HIMA_Siemens2	Byte
HIMA_Siemens3	WORD

Таблица 184: Переменные для передаваемых данных

4. Открыть контекстное меню, щелкнув правой кнопкой мыши на пустое место в области **Register Inputs** и выбрать **New Offsets**, чтобы заново пронумеровать смещения переменных.

i

Обращайте внимание на то, что смещения переменных в системе управления HIMAх должны совпадать с адресами переменных в типе данных *UDT_1* системы SIMATIC 300.

Подтверждение конфигурации S&R TCP:

1. Открыть в структуре **Configuration, Resource, Protocols, Send/Receive over TCP**.
2. Щелкнуть на кнопку **Verification** в строке меню и подтвердить с помощью **OK**.
3. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.

7.3 Функции меню протокола S&R-TCP

7.3.1 Редактирование (Edit)

Диалоговое окно **Edit** протокола S&R-TCP содержит следующие вкладки.

Системные переменные

С помощью *System Variables* можно проанализировать состояние протокола TCP Send Receive в прикладной программе.

Элемент	Описание
Number of active connections [UDIN T]	Системная переменная, которая сообщает количество активных (исправных) соединений.
Number of disturbed connections [UDIN T]	Системная переменная, которая сообщает количество неисправных соединений. «Неисправный» означает, что соединение TCP было прервано в результате времени ожидания или ошибки.
Status [DWORD]	Функция отсутствует

Таблица 185: Системные переменные S&R TCP

7.3.2 Свойства

Обмен данными через соединение TCP осуществляется циклически либо ациклически. Для ациклического обмена данными требуются функциональные блоки S&R-TCP.

Циклический и ациклический обмен данными не может использоваться одновременно в одном соединении.

Общие положения

Название	Описание
Type	Send/Receive over TCP
Name	Имя протокола Send/Receive-over-TCP. Максимум 31 символ.
Module	Выбор модуля COM, на котором обрабатывается данный протокол.
Use Max CPU Load	Активировано: Сохранить предельное значение нагрузки ЦПУ из поля Max. CPU Load [%]. Деактивировано: Не использовать предельное значение нагрузки ЦПУ для данного протокола.
Max. CPU Load [%]	Максимальная нагрузка модуля ЦПУ, которая может создаваться при обработке протокола. Диапазон значений: 1...100% Значение по умолчанию: 30%

Таблица 186: Общие свойства S&R TCP

ЦПУ/СОМ

Заданные параметры обеспечивают высокоскоростной обмен данными S&R TCP между модулем СОМ и модулем ЦПУ в системе управления. Данные параметры необходимо изменять только в том случае, если снижение нагрузки СОМ и/или ЦПУ требуется для приложения, и это допустимо для процесса.

i

Изменение параметров рекомендуется доверять только опытному программисту. Увеличение времени обновления СОМ и ЦПУ означает также, что увеличится фактическое время обновления данных S&R TCP. Необходимо проверять временные требования устройства.

Название	Описание
Refresh Rate [ms]	Время обновления в миллисекундах, когда происходит обмен данными в протоколе S&R-TCP между СОМ и ЦПУ. Если значение Refresh Rate равняется нулю или меньше, чем время цикла системы управления, то обмен данными осуществляется так быстро, насколько это возможно. Диапазон значений: $0 \dots (2^{31} - 1)$ Значение по умолчанию: 0
Within one cycle	Активировано: Передача данных S&R TCP от ЦПУ на СОМ за один цикл ЦПУ. Деактивировано: Передача данных S&R TCP (максимум 900 байтов для одного направления) от ЦПУ на СОМ за несколько циклов ЦПУ.

Таблица 187: Параметры СОМ/ЦПУ

7.4 Функции меню соединения TCP

7.4.1 Редактирование (Edit)

С помощью функции меню **Edit** Вы можете попасть во вкладки для переменных процесса и системных переменных.

Переменные процесса

Input Signals

Переменные для циклического обмена данными, которые должны приниматься этой системой управления, вводятся в область *Input Signals*.

Во вкладке *Input Signals* можно создавать любые переменные. Смещения и типы переменных должны совпадать со смещениями и типами переменных (передаваемые данные) участника коммуникации.

Output Signals

Переменные для циклического обмена данными, которые должны передаваться этой системой управления, вводятся в область *Output Signals*.

Во вкладке *Output Signals* можно создавать любые переменные. Смещения и типы переменных должны совпадать со смещениями и типами переменных (получаемые данные) участника коммуникации.

7.4.2 Системные переменные

С помощью переменных во вкладке *System Variables* можно проанализировать состояние соединения TCP в прикладной программе.

Название	Описание
Bytes received [UDINT]	Число принятых байтов
Bytes sent [UDINT]	Число отправленных байтов
Error code [UDINT]	Код ошибки соединения TCP. См. главу 7.8.4
Error code time stamp [ms] [UDINT]	Доля метки времени в миллисекундах. Время возникновения ошибки.
Error code time stamp [s] [UDINT]	Доля метки времени в секундах. Время возникновения ошибки.
Partner Request Timeout [UDINT]	При циклической передаче данных: Время ожидания, в течение которого после передачи данных минимум один пакет данных должен быть принят участником коммуникации. 0 = Выкл 1...2 ³² -1 [мс]
Partner Connection State [BYTE]	Если в течение времени ожидания не принят ни один пакет данных, то состояние соединения участника коммуникации устанавливается на <i>not connected</i> , и соединение выполняется заново. 0 = Отсутствует соединение 1 = Соединение ОК
Status [DWORD]	Состояние соединения TCP. (См. главу 7.8.6)

Таблица 188: Системные переменные

7.4.3 Свойства

Обмен данными через соединение TCP осуществляется циклически либо ациклически. Для ациклического обмена данными требуются функциональные блоки S&R-TCP. При циклической передаче данных эксплуатация функциональных блоков S&R-TCP невозможна.

Название	Описание
Type	Соединение TCP
Name	Любое однозначное имя для соединения TCP. Максимум 31 символ.
ID	Любой, но однозначный идентификационный номер для каждого соединения TCP. ID № требуется также в качестве ссылки в функциональных блоках S&R-TCP. Диапазон значений 0..255 Значение по умолчанию: 0
Mode	Сервер (значение по умолчанию): Данная станция работает в качестве сервера, т.е. в пассивном режиме. Создание соединения должно инициироваться участником коммуникации (клиент). После установки первого соединения обе станции являются равноправными и могут передавать данные в любое время. Требуется указание собственного порта.
	Сервер с определенным участником коммуникации: Данная станция работает в качестве сервера, т.е. в пассивном режиме. Создание соединения должно инициироваться участником коммуникации (клиент). После установки первого соединения обе станции являются равноправными и могут передавать данные в любое время. Если здесь введен IP-адрес и/или порт участника коммуникации, то только определенный участник коммуникации может принять соединение. Все другие станции игнорируются. Если один из параметров (IP-адрес или порт) сбрасывается на нуль, то проверка для данного параметра не осуществляется.
	Клиент: Данная станция работает в качестве клиента, т. е. станция инициирует создание соединения с участником коммуникации. Требуется указание IP-адреса и порта участника коммуникации. Опционально может указываться также собственный порт.
Partner IP address	IP-адрес участника коммуникации. 0.0.0.0: допускается любой IP-адрес. Действительный диапазон: 1.0.0.0...223.255.255.255, кроме: 127.x.x.x Значение по умолчанию: 0
Partner port	Порт участника коммуникации. Нуль: любой порт. Зарезервированные или уже занятые порты (1...1024) отклоняются COM-BS. Диапазон значений 0.65535 Значение по умолчанию: 0

Название	Описание
Own Port	Собственный порт. Нуль означает любой порт. Зарезервированные или уже занятые порты (1...1024) отклоняются COM-BS. Диапазон значений 0...65535 Значение по умолчанию: 0
Cyclic data transfer	Деактивировано (Значение по умолчанию): Циклическая передача данных деактивирована. Обмен данными через данное соединение TCP должен программироваться с помощью функциональных блоков. В этом соединении не должны быть определены циклические входные/выходные данные.
	Активировано: Циклическая передача данных активна. Данные определяются в диалоговом окне Process Variables соединения TCP. Должны быть определены принимаемые данные. В этом соединении не должны эксплуатироваться функциональные блоки.
Send Interval [ms]	Только редактирование при циклической передаче данных. Здесь настраивается интервал передачи. Диапазон значений 10...2147483647 мс (небольшие значения округляются до 10 мс) Значение по умолчанию: 0
Keep Alive Interval [s]	Время до того момента, пока установленный TCP контроль соединения не будет активен. Нуль деактивирует контроль соединения. Если в течение настроенного интервала сообщений KeepAlive не происходит обмен данными, то участнику коммуникации отправляются пробы KeepAlive. Если соединение еще присутствует, то пробы KeepAlive подтверждаются участником коммуникации. Если за время > 10 интервалов KeepAlive не происходит обмен данными между участниками коммуникации, то соединение закрывается. Если после передачи пакета данных не получен ответ, то в определенные интервалы пакет данных передается повторно. Соединение прерывается после 12 неудачных повторов (прибл. 7 минут). Диапазон значений 1... 65535 с Значение по умолчанию: 0 = деактивировано
Partner Request Timeout [ms]	При циклической передаче данных: Время ожидания, в течение которого после передачи данных минимум один пакет данных должен быть принят участником коммуникации. Если в течение времени ожидания не принят ни один пакет данных, то <i>Partner connection</i> устанавливается на <i>not connected</i> , и соединение выполняется заново. После закрытия соединения в результате наступления времени ожидания или другой ошибки активная сторона заново создает соединение с задержкой 10 x PartnerRequestTimeout либо 10 секунд, если PartnerRequestTimeout = 0. Пассивная сторона открывает порт уже по прошествии половины времени. 0 = Выкл Диапазон значений 1...2 ³² -1 [мс] Значение по умолчанию: 0

Таблица 189: Свойства соединения S&R TCP

7.5 Обмен данными

S&R TCP работает по принципу клиент-сервер. Создание соединения должно инициироваться участником коммуникации, который конфигурирован как клиент. После установки первого соединения оба участника коммуникации являются равноправными и могут передавать данные в любое время.

S&R TCP не имеет собственного протокола для надежной передачи данных, а использует для этого протокол TCP/IP. Поскольку TCP передает данные в «поток данных», то необходимо убедиться, что смещения и типа передаваемых переменных принимаемой и передаваемой стороны идентичны.

S&R TCP совместим с интерфейсом SEND/RECEIVE Siemens и позволяет осуществлять циклический обмен данными с функциональными блоками Siemens S7 AG_SEND (FC5) и AG_RECV (FC6) (см. главу 7.2, пример Конфигурация S&R TCP).

Кроме того, компания HIMA предлагает пять функциональных блоков S&R-TCP, посредством которых в прикладной программе коммуникацией можно управлять и подстраивать ее индивидуально. С помощью функциональных блоков S&R-TCP можно передавать и принимать различные протоколы (напр., Modbus), которые передаются через TCP.

7.5.1 Соединения TCP

Для каждого соединения с участником коммуникации через S&R TCP необходимо создать минимум одно соединение TCP в системе управления HIMAх.

В *Properties* соединения TCP должен быть занесен идентификационный номер соединения TCP и адреса/порты собственной системы управления и участника коммуникации.

В системе управления HIMAх можно создать максимум 32 соединения TCP.

Созданные соединения TCP должны иметь различные идентификационные номера и различные адреса/порты.

Чтобы создать новое соединение TCP, необходимо:

1. В структуре открыть **Configuration, Resource**.
2. Щелкнуть правой кнопкой мыши на **Protocols** и выбрать в контекстном меню **New**.
3. Выбрать **Send/Receive over TCP** и ввести имя для протокола.
4. Подтвердить с помощью ОК, чтобы создать новый протокол.
5. Щелкнуть правой кнопкой мыши на **Send/Receive over TCP** и выбрать в контекстном меню **Properties**.
6. Выбрать **COM Modul**. Остальные параметры сохраняют значения по умолчанию.

СОВЕТ Система управления HIMAх и сторонняя система должны находиться в одной и той же субсети или при использовании маршрутизатора иметь соответствующие записи маршрута.

7.5.2 Циклический обмен данными

Если используется циклический обмен данными, то в системе управления NI-Max и у участника коммуникации должен быть определен интервал передачи.

Интервал передачи представляет собой циклический интервал, в течение которого участник коммуникации, передающий данные, передает свои переменные участнику коммуникации, принимающему данные.

- Чтобы обеспечить непрерывный обмен данными, у обоих участников коммуникации должен быть настроен приблизительно одинаковый интервал передачи (см. главу 7.5.5, контроль потока).
- Опция *Cyclic Data Transfer* должна быть активирована в используемом соединении TCP для циклического обмена данными.
- В соединении TCP, в котором активирована опция *Cyclic Send Data*, не должны использоваться функциональные блоки.
- Передаваемые и принимаемые переменные присваиваются соединению TCP в диалоговом окне *Process Variables*. Принимаемые данные должны присутствовать **обязательно**, наличие передаваемых данных опционально.

i

Одинаковые переменные (одинаковые смещения и типы), которые в одной станции определены как передаваемые данные, в другой станции должны быть определены как принимаемые данные.

7.5.3 Ациклический обмен данными с помощью функциональных блоков

Ациклический обмен данными в системе управления NI-Max контролируется прикладной программой с помощью функциональных блоков S&R-TCP.

Таким образом, существует возможность осуществлять контроль обменом данных с помощью таймера или механического переключателя на физическом входе системы управления NI-Max.

Опция *Cyclic Data Transfer* в используемом соединении TCP должна быть деактивирована.

Для каждого направления передачи в соединении TCP всегда может использоваться только один **S&R TCP Function Block** (одновременно).

Передаваемые или получаемые переменные назначаются в диалоговом окне *Process Variables* функциональных блоков S&R-TCP (все кроме сброса).

i

Одинаковые переменные (одинаковые смещения и типы), которые в одной станции определены как передаваемые данные, в другой станции должны быть определены как принимаемые данные.

7.5.4 Одновременный циклический и ациклический обмен данными

Для этого одно соединение TCP необходимо конфигурировать для циклических данных и второе для ациклических данных. Оба соединения TCP должны использовать разные *IP-адреса* и *порты* для участников коммуникации.

Одно соединение TCP не может использоваться сразу для циклического и ациклического обмена данными.

7.5.5 Контроль потока

Контроль потока является частью TCP и контролирует непрерывный обмен данными между двумя участниками коммуникации.

При циклической передаче данных максимум после 3–5 отправленных пакетов минимум один пакет должен быть принят, в противном случае передача блокируется, пока пакет не будет принят, или контроль соединения прервет соединение.

Число (3...5) возможных передач без получения пакета зависит от размера передаваемых пакетов.

Число=5 для небольших пакетов < 4 кБ.

Число=3 для больших пакетов ≥ 4 кБ.

- При создании проекта следует обращать внимание на то, что ни одна из станций не будет отправлять данных больше, чем другая может обработать синхронно.
- Для циклического обмена данными у обоих участников коммуникации должен быть настроен примерно одинаковый интервал передачи.

7.6 Сторонние системы с холостыми байтами

При циклическом и ациклическом обмене данными следует обращать внимание на то, что некоторые системы управления (напр., SIMATIC 300) вводят так называемые *pad bytes* (холостые байты). Таким образом обеспечивается, что все типы данных, размер которых больше одного байта, всегда начинаются с четного смещения и чтобы общая длина пакетов (в байтах) тоже всегда была четной.

В системе управления HiMax для *холостых байтов* в соответствующих местах необходимо ввести *фиктивные байты*.

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	InOut_1	BYTE	B#16#0
+2.0	InOut_3	WORD	W#16#0
=4.0		END_STRUCT	

Рисунок 63: Список переменных Siemens

В системе управления Siemens вводится *холостой байт* (невидимый), чтобы переменная *InOut_3* начиналась на четном смещении.

Output Signals				
F	Name	Data type	Offset	Global Variable
1	InOut_1	BYTE	0	InOut_1
2	Dummy	BYTE	1	Dummy
3	InOut_3	WORD	2	InOut_3

Рисунок 64: Список переменных HiMax

В системе управления HiMax необходимо ввести *фиктивный байт*, чтобы переменная *InOut_3* имела такое же смещение, как и в системе управления Siemens.

7.7 Функциональные блоки S&R-TCP

Если циклическая передача данных не достаточно удобна, то данные могут передаваться и приниматься также посредством функциональных блоков S&R-TCP. Опция *Cyclic data transfer* в используемом соединении TCP должна быть деактивирована.

С помощью функциональных блоков S&R-TCP пользователь может оптимально адаптировать передачу данных через TCP/IP в соответствии с требованиями своего проекта.

Функциональные блоки параметрируются в прикладной программе. Таким образом, функции (передача, прием, сброс) системы управления HIMax могут устанавливаться и анализироваться в прикладной программе.

Функциональные блоки S&R-TCP требуются только для ациклического обмена данными. Для циклического обмена данными между сервером и клиентом данные функциональные блоки не требуются!

i

Конфигурирование функциональных блоков S&R-TCP описывается в главе 10.

Доступны следующие функциональные блоки:

Функциональный блок	Описание функции
TCP_Reset (см. главу 7.7.1)	Сброс соединения TCP
TCP_Send (см. главу 7.7.2)	Передача данных
TCP_Receive (см. главу 7.7.3)	Прием пакетов данных фиксированной длины
TCP_ReceiveLine (см. главу 7.7.4)	Прием строки ASCII
TCP_ReceiveVar (см. главу 7.7.5)	Прием пакетов данных различной длины (с полем длины)
LATCH	Используется только в других функциональных блоках
PIG	Используется только в других функциональных блоках
PIGII	Используется только в других функциональных блоках

Таблица 190: Функциональные блоки для соединений S&R TCP

7.7.1 TCP_Reset

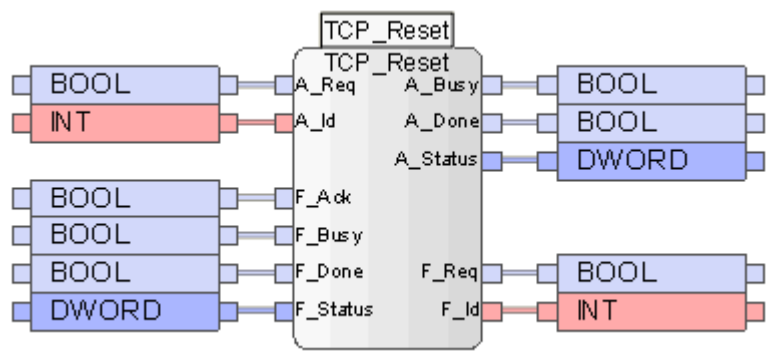


Рисунок 65: Функциональный блок TCP_Reset

С помощью функционального блока **TCP Reset** можно восстановить нарушенное соединение, если функциональный блок передачи или приема сообщает об ошибке TIMEOUT (16#8A).

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом А:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «А» означает «Application» (приложение).

А-входы	Описание	Тип
A_Req	Передний фронт запускает блок	BOOL
A_Id	Идентификационный номер <i>ID</i> нарушенного соединения TCP, который необходимо сбросить.	INT

Таблица 191: А-входы Функциональный блок TCP_Reset

А-выходы	Описание	Тип
A_Busy	TRUE: Сброс функционального блока еще не завершен.	BOOL
A_Done	TRUE: Процесс передачи завершен без ошибок	BOOL
A_Status	На выходе <i>A_Status</i> выдается состояние и код ошибки функционального блока и соединения TCP.	DWORD

Таблица 192: А-выходы Функциональный блок TCP_Reset

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы создают соединение для функционального блока **Reset** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **Reset** в структуре (в папке Функциональные блоки) с функциональным блоком **TCP_Reset** (в прикладной программе) осуществляется посредством общих переменных. Они должны быть созданы пользователем заранее в редакторе переменных.

Соедините *F-Inputs* функционального блока **TCP_Reset** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **Reset** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Busy	BOOL
F_Done	BOOL
F_Status	DWORD

Таблица 193: F-входы Функциональный блок TCP_Reset

Соедините *F-Outputs* функционального блока **TCP_Reset** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **Reset** в структуре.

F-выходы	Тип
F_Req	BOOL
F_Id	DWORD

Таблица 194: F-выходы Функциональный блок TCP_Reset

Чтобы создать соответствующий функциональный блок Reset в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New**.
2. Выбрать функциональный блок **Reset**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **Reset** и выбрать **Edit**
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **Reset** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **TCP_Reset** в прикладной программе.

Входы	Тип
ID	DWORD
REQ	BOOL

Таблица 195: Входные системные переменные

Соедините следующие выходы функционального блока **Reset** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **TCP_Reset** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Таблица 196: Выходные системные переменные

Для использования функционального блока TCP_Reset необходимо:

1. В прикладной программе на входе *A_ID* установить идентификационный номер нарушенного соединения TCP.
2. В прикладной программе установить вход *A_Req* на TRUE.

i

Функциональный блок реагирует на передний фронт на *A_Req*.

Выход *A_Busy* является TRUE, пока сброс не будет опрашен на определенное соединение TCP. Затем выход *A_Busy* меняется на FALSE и выход *A_Done* на TRUE.

7.7.2 TCP_Send

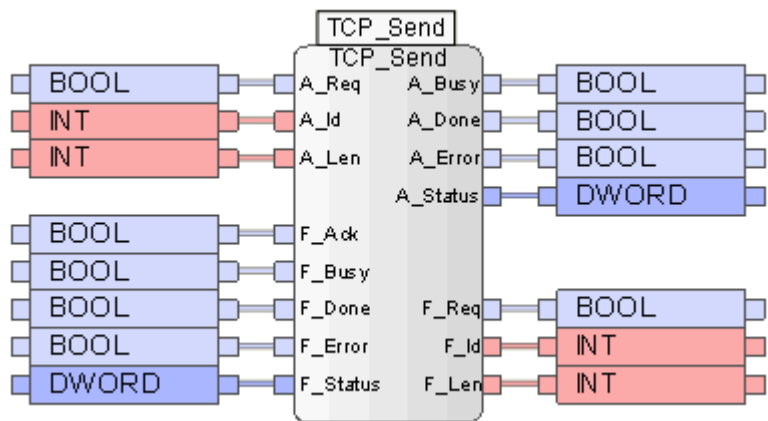


Рисунок 66: Функциональный блок TCP_Send

Функциональный блок **TCP_Send** служит для ациклической передачи переменных участнику коммуникации. У участника коммуникации функциональный блок, напр., *Receive* должен быть конфигурирован с такими же переменными и смещениями.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом A:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	Передний фронт запускает блок.	BOOL
A_Id	Идентификационный номер конфигурированного соединения TCP для участника коммуникации, которому должны передаваться данные.	INT
A_Len	Число передаваемых переменных в байтах. A_Len должна быть больше нуля и не должна заканчиваться внутри переменной.	INT

Таблица 197: A-входы Функциональный блок TCP_Send

A-выходы	Описание	Тип
A_Busy	TRUE: Процесс передачи еще не завершен.	BOOL
A_Done	TRUE: Процесс передачи завершен без ошибок	BOOL
A_Error	TRUE: Возникла ошибка FALSE: Нет ошибки	BOOL
A_Status	На выходе A_Status выдается состояние и код ошибки функционального блока и соединения TCP.	DWORD

Таблица 198: A-выходы Функциональный блок TCP_Send

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение с функциональным блоком **Send** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **Send** в структуре (в папке Функциональные блоки) с функциональным блоком **TCP_Send** (в прикладной программе) осуществляется посредством общих переменных. Они должны быть созданы пользователем заранее в редакторе переменных.

Соедините *F-Inputs* функционального блока **TCP_Send** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **Send** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Busy	BOOL
F_Done	BOOL
F_Error	BOOL
F_Status	DWORD

Таблица 199: F-входы Функциональный блок TCP_Send

Соедините *F-Outputs* функционального блока **TCP_Send** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **Send** в структуре.

F-выходы	Тип
F_Id	DWORD
F_Len	INT
F_Req	BOOL

Таблица 200: F-выходы Функциональный блок TCP_Send

Чтобы создать соответствующий функциональный блок Send в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New**.
2. Выбрать функциональный блок **Send**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **Send** и выбрать **Edit**
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **Send** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **TCP_Send** в прикладной программе.

Входы	Тип
ID	DWORD
LEN	INT
REQ	BOOL

Таблица 201: Входные системные переменные

Соедините следующие выходы функционального блока **Send** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **TCP_Send** в прикладной программе.

Выходы	Тип
Ack	BOOL
Busy	BOOL
Done	BOOL
ERROR	BOOL
STATUS	DWORD

Таблица 202: Выходные системные переменные

Данные	Описание
Переданные данные	Во вкладке <i>Process Variables</i> можно создавать любые переменные. Смещения и типы переменных должны совпадать со смещениями и типами переменных участника коммуникации.

Таблица 203: Переданные данные

Для использования функционального блока TCP_Send необходимо:

·

Передаваемые переменные должны создаваться во вкладке *Process Variables* диалогового окна *Send*. Смещения и типы переменных должны совпадать со смещениями и типами переменных участника коммуникации.

1. В прикладной программе указать ID соединения TCP на входе *A_ID*.
2. В прикладной программе указать длину передаваемых переменных на входе *A_Len* в байтах.
3. В прикладной программе установить вход *A_Req* на TRUE.

·

Функциональный блок реагирует на передний фронт на *A_Req*.

Выход *A_Busy* установлен на TRUE до тех пор, пока переменные не будут переданы. Затем выходы *A_Busy* меняются на FALSE и выходы *A_Done* на TRUE.

Если процесс передачи выполнен неудачно, то выход *A_Error* переходит на TRUE, и на выходе *A_Status* выдается код ошибки.

7.7.3 TCP_Receive

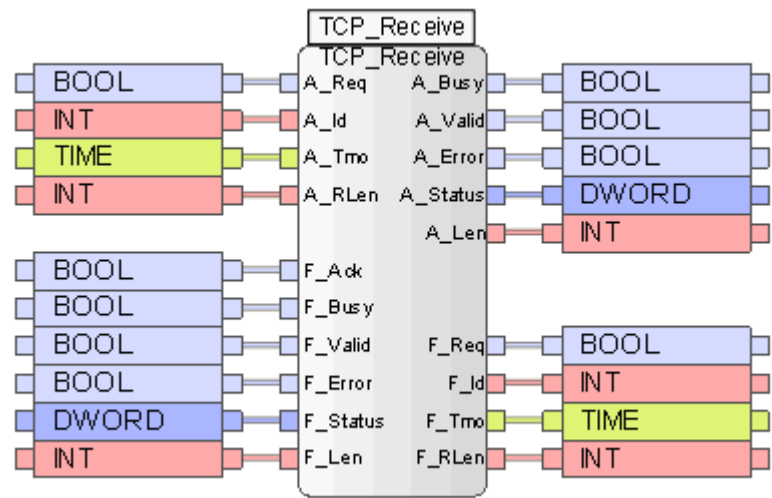


Рисунок 67: Функциональный блок TCP_Receive

С помощью функционального блока **TCP_Receive** могут приниматься определенные переменные от участника коммуникации.

У участника коммуникации функциональный блок, напр., TCP_Send должен быть конфигурирован с такими же переменными и смещениями.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом A:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	Передний фронт запускает функциональный блок.	BOOL
A_Id	Идентификационный номер конфигурированного соединения TCP для участника коммуникации, от которого должны быть получены данные.	INT
A_Tmo	Время ожидания приема. Если в течение этого времени не будут получены данные, то блок завершает работу и выдает сообщение об ошибке. Если вход A_Tmo не занят или установлен на нуль, то время ожидания деактивировано.	TIME
A_RLen	A_RLen – это ожидаемая длина принимаемых переменных в байтах. A_RLen должна быть больше нуля и не должна заканчиваться внутри переменной.	INT

Таблица 204: A-входы Функциональный блок TCP_Receive

A-выходы	Описание	Тип
A_Busy	TRUE: Прием данных еще не завершен.	BOOL
A_Valid	TRUE: Прием данных завершен без ошибок.	BOOL
A_Error	TRUE: Возникла ошибка FALSE: Нет ошибки	BOOL
A_Status	На выходе A_Status выдается состояние и код ошибки функционального блока и соединения TCP.	DWORD
A_Len	Число принятых байтов.	INT

Таблица 205: A-выходы Функциональный блок TCP_Receive

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы создают соединение с функциональным блоком **Receive** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **Receive** в структуре (в папке Функциональные блоки) с функциональным блоком **TCP_Receive** (в прикладной программе) осуществляется посредством общих переменных. Они должны быть созданы пользователем заранее в редакторе переменных.

Соедините *F-Inputs* функционального блока **TCP_Receive** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **Receive** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Таблица 206: A-входы Функциональный блок TCP_Receive

Соедините *F-Outputs* функционального блока **TCP_Receive** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **Receive** в структуре.

F-выходы	Тип
F_Req	BOOL
F_Id	DWORD
F_Tmo	INT
F_RLen	INT

Таблица 207: F-выходы Функциональный блок TCP_Receive

Чтобы создать соответствующий функциональный блок **Receive** в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New**.
2. Выбрать функциональный блок **Receive**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **Receive** и выбрать **Edit**
☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **Receive** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **TCP_Receive** в прикладной программе.

Входы	Тип
ID	INT
REQ	BOOL
RLEN	INT
TIMEOUT	TIME

Таблица 208: Входные системные переменные

Соедините следующие выходы функционального блока **Receive** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **TCP_Receive** в прикладной программе.

Выходы	Тип
Ack	BOOL
Busy	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Таблица 209: Выходные системные переменные

Данные	Описание
Переменные приема	Во вкладке <i>Process Variables</i> можно создавать любые переменные. Смещения и типы переменных должны совпадать со смещениями и типами переменных участника коммуникации.

Таблица 210: Переменные приема

Для использования функционального блока TCP_Receive необходимо:

i

Переменные приема должны создаваться во вкладке *Process Variables* диалогового окна *Receive*. Смещения и типы переменных приема должны совпадать со смещениями и типами переменных передачи участника коммуникации.

1. В прикладной программе указать идентификационный номер соединения TCP на входе *A_ID*.
 2. В прикладной программе установить время ожидания приема на входе *A_Tmo*.
 3. В прикладной программе установить ожидаемую длину принимаемых переменных на входе *A_RLen*.
 4. В прикладной программе установить вход *A_Req* на TRUE.
-

i

Функциональный блок запускает с помощью переднего фронта на *A_Req*.

Выход *A_Busy* является TRUE, пока переменные не будут приняты или не истечет время ожидания приема. Затем выход *A_Busy* переходит на FALSE и выходы *A_Valid* или *A_Error* на TRUE.

Если прием переменных осуществлен без ошибок, то выход *A_Valid* переходит на TRUE. Переменные, определенные во вкладке *Data*, можно анализировать.

Если прием переменных осуществлен с ошибками, то выход *A_Error* переходит на TRUE, и на выходе *A_Status* выдается код ошибки.

7.7.4 TCP_ReceiveLine

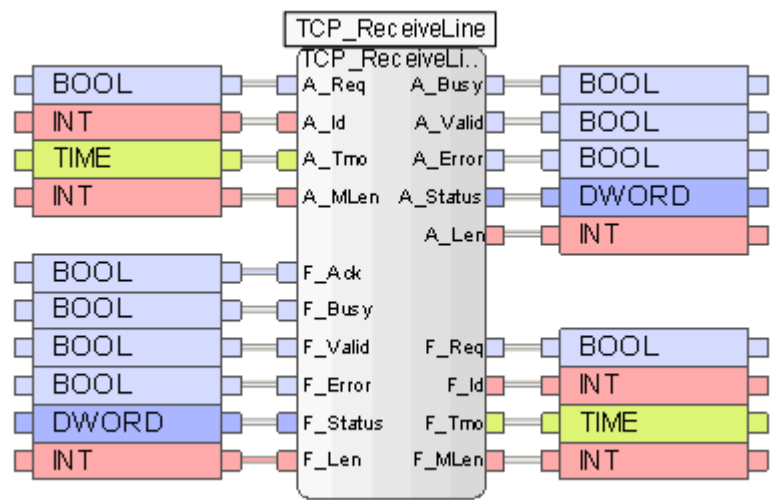


Рисунок 68: Функциональный блок TCP_ReceiveLine

Функциональный блок **TCP_ReceiveLine** служит для приема цепочки символов ASCII, включая LineFeed (16#0A) участника коммуникации.

i Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Входы и выходы функционального блока с префиксом A:

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	Передний фронт запускает блок и подготавливает соединение к приему	BOOL
A_Id	Идентификационный номер конфигурированного соединения TCP для участника коммуникации, от которого должны быть получены данные.	INT
A_Tmo	Время ожидания приема Если в течение этого времени не будут получены данные, то блок завершает работу и выдает сообщение об ошибке. Если вход оставлен открытым или установлен нуль, то время ожидания выключено.	TIME
A_MLen	<i>A_Mlen</i> – максимальная длина принимаемой строки в байтах. Переменные приема должны создаваться во вкладке <i>Data</i> в функциональном блоке <i>Com</i> . Передаваемые байты = Мин (<i>A_MLen</i> , длина строки, длина области данных).	INT

Таблица 211: A-входы Функциональный блок TCP_ReceiveLine

A-выходы	Описание	Тип
A_Busy	TRUE: Прием данных еще не завершен.	BOOL
A_Valid	TRUE: Прием данных завершен без ошибок.	BOOL
A_Error	TRUE: Возникла ошибка FALSE: Нет ошибки	BOOL
A_Status	На выходе A_Status выдается состояние и код ошибки функционального блока и соединения TCP.	DWORD
A_Len	Число принятых байтов.	INT

Таблица 212: A-выходы Функциональный блок TCP_ReceiveLine

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение для функционального блока **ReceiveLine** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **ReceiveLine** в структуре (в папке Функциональные блоки) с функциональным блоком **TCP_ReceiveLine** (в прикладной программе) осуществляется посредством общих переменных. Они должны быть созданы пользователем заранее в редакторе переменных.

Соедините *F-Inputs* функционального блока **TCP_ReceiveLine** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **ReceiveLine** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Таблица 213: F-входы Функциональный блок TCP_ReceiveLine

Соедините *F-Outputs* функционального блока **TCP_ReceiveLine** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **ReceiveLine** в структуре.

F-выходы	Тип
A_Req	BOOL
A_Id	INT
A_Tmo	TIME
A_MLen	INT

Таблица 214: F-выходы Функциональный блок TCP_ReceiveLine

Чтобы создать соответствующий функциональный блок **ReceiveLine** в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New**.
2. Выбрать функциональный блок **ReceiveLine**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **ReceiveLine** и выбрать **Edit**.
 - ☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **ReceiveLine** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **TCP_ReceiveLine** в прикладной программе.

Входы	Тип
ID	INT
MLEN	INT
REQ	BOOL
TIMEOUT	TIME

Таблица 215: Входные системные переменные

Соедините следующие выходы функционального блока **ReceiveLine** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **TCP_ReceiveLine** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Таблица 216: Выходные системные переменные

Данные	Описание
Переменные приема	Во вкладке <i>Process Variables</i> должны создаваться только переменные типа BYTE. Смещения переменных должны совпадать со смещениями переменных участника коммуникации.

Таблица 217: Переменные приема

Для использования функционального блока TCP_ReceiveLine необходимо:

-
- i** Переменные приема байтового типа должны создаваться во вкладке *Process Variables* диалогового окна *ReceiveLine*. Смещения переменных приема должны совпадать со смещениями переменных передачи участника коммуникации.
-
1. В прикладной программе указать идентификационный номер соединения TCP на входе *A_ID*.
 2. В прикладной программе установить время ожидания приема на входе *A_Tmo*.
 3. В прикладной программе установить ожидаемую длину принимаемой строки на входе *A_MLen*.
-

- i** *A_MLen* должна быть больше нуля и определяет размер буфера приема в байтах. Если буфер приема заполнен, а конец строки еще не появился, то процесс считывания завершается без сообщения об ошибке. На выходе *A_Len* доступно число принятых байтов: Принятые байты = Мин (*A_MLen*, длина строки, длина области данных).
-

4. В прикладной программе установить вход *A_Req* на TRUE.
-

- i** Функциональный блок реагирует на передний фронт на *A_Req*.
-

Выход *A_Busy* является TRUE, пока буфер приема не будет заполнен или не будет принят конец строки *LineFeed* или не истечет время ожидания приема. Затем выходы *A_Busy* переходят на FALSE и *A_Valid* или *A_Error* на TRUE.

Если прием переменных осуществлен без ошибок, то выход *A_Valid* переходит на TRUE. Переменные, определенные во вкладке *Data*, можно анализировать.

Если прием переменных осуществлен с ошибками, то выход *A_Error* переходит на TRUE, и на выходе *A_Status* выдается код ошибки.

7.7.5 TCP_ReceiveVar

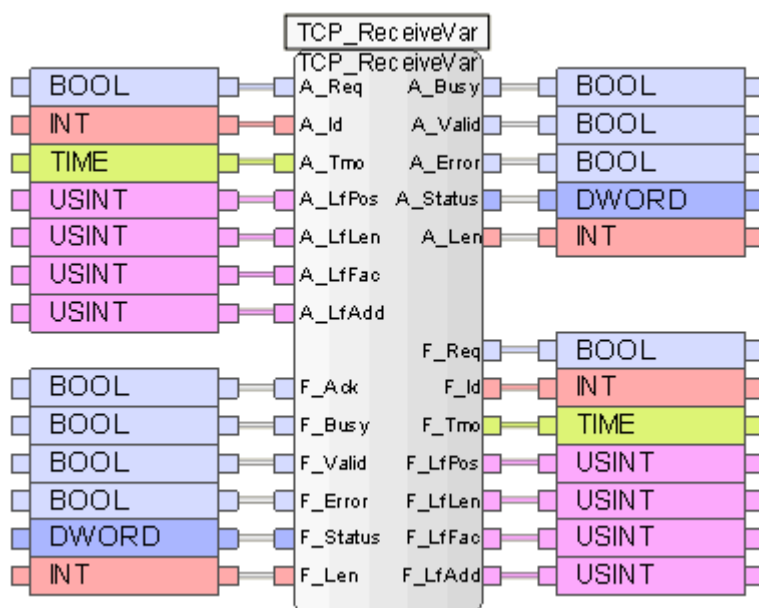


Рисунок 69: Функциональный блок TCP_ReceiveVar

С помощью функционального блока **TCP_ReceiveVar** можно анализировать пакеты данных любой длины, которые имеют поле длины.

i

Для конфигурирования перетащите функциональный блок с помощью Drag&Drop из библиотеки блоков в прикладную программу (см. также главу 10).

Описание функции

Принятые пакеты данных должны иметь изображенную на рисунке ниже структуру (напр., протокол Modbus). Подгонка под любой формат протокола осуществляется путем настройки входных параметров *A_LfPos*, *A_LfLen*, *A_LfFac*, *A_LfLen*.

Принятый пакет данных состоит из области заголовка и полезной нагрузки. Область заголовка содержит такие данные, как адрес участника коммуникации, функция телеграммы, поле длины и т.д., которые требуются для коммуникационного соединения. Чтобы проанализировать область полезной нагрузки необходимо отделить область заголовка и считать поле длины.

Размер области заголовка заносится в параметр *A_LfAdd*.

Длина области полезной нагрузки должна считываться из поля длины текущего считанного пакета данных. Позиция поля длины заносится в параметр *A_LfPos*. Размер поля длины заносится в *LfLen* в байтах. Если длина указана не в байтах, то для этого в *A_LfFac* должен быть внесен переводный коэффициент (напр., 2 для Word или 4 для Double Word).

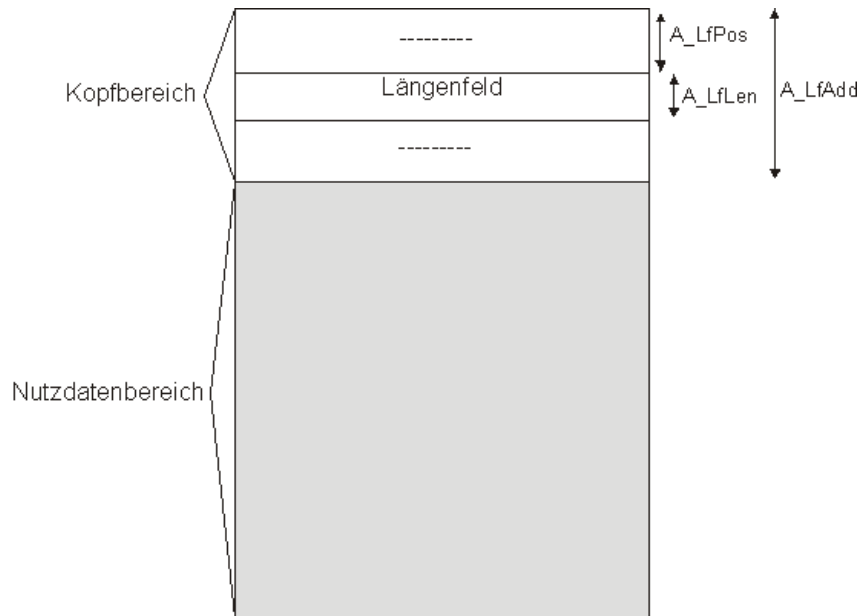


Рисунок 70: Структура пакета данных

Входы и выходы функционального блока с префиксом A

Через эти входы и выходы Вы можете контролировать и анализировать функциональный блок с помощью прикладной программы. Префикс «A» означает «Application» (приложение).

A-входы	Описание	Тип
A_Req	С помощью переднего фронта запускается функциональный блок ЦПУ	BOOL
A_Id	Идентификационный номер <i>ID</i> конфигурированного соединения TCP для участника коммуникации, от которого должен быть получен пакет данных.	DWORD
A_Tmo	Время ожидания приема Если в течение этого времени не будут получены данные, то блок завершает работу и выдает сообщение об ошибке. Если вход оставлен открытым или установлен нуль, то время ожидания выключено.	INT
A_LfPos	Начальная позиция поля длины в пакете данных; нумерация начинается с нуля (измерено в байтах).	USINT
A_LfLen	Размер поля длины <i>A_LfLen</i> в байтах. Разрешены 1, 2 или 4 байта.	USINT
A_LfFac	Переводный коэффициент в байтах, если запись в поле длины сделана не в байтах. Если вход оставлен открытым или установлен нуль, то в качестве значения по умолчанию берется 1.	USINT
A_LfAdd	Размер поля заголовка в байтах	USINT

Таблица 218: A-входы Функциональный блок TCP_ReceiveVar

A-выходы	Описание	Тип
A_Busy	TRUE: Прием данных еще не завершен.	BOOL
A_Valid	TRUE: Прием данных завершен без ошибок.	BOOL
A_Error	TRUE: При считывании возникла ошибка FALSE: Нет ошибки	BOOL
A_Status	На выходе <i>A_Status</i> выдается состояние и код ошибки функционального блока и соединения TCP.	DWORD
A_Len	Число принятых байтов.	INT

Таблица 219: A-выходы Функциональный блок TCP_ReceiveVar

Входы и выходы функционального блока с префиксом F:

Данные входы и выходы функционального блока создают соединение для функционального блока **Variabel Receive** в структуре. Префикс «F» означает «Field» (поле).

i

Соединение функционального блока **Variabel Receive** в структуре (в папке Функциональные блоки) с функциональным блоком **TCP_ReceiveVar** (в прикладной программе) осуществляется посредством общих переменных. Они должны быть созданы пользователем заранее в редакторе переменных.

Соедините *F-Inputs* функционального блока **TCP_ReceiveVar** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините выходы функционального блока **Variabel Receive** в структуре.

F-входы	Тип
F_Ack	BOOL
F_Busy	BOOL
F_Valid	BOOL
F_Error	BOOL
A_Status	DWORD
A_Len	INT

Таблица 220: F-входы Функциональный блок TCP_ReceiveVar

Соедините *F-Outputs* функционального блока **TCP_ReceiveVar** в прикладной программе с одинаковыми переменными, с которыми Вы позже соедините входы функционального блока **Variabel Receive** в структуре.

F-выходы	Тип
F_Req	BOOL
F_Id	INT
F_Tmo	TIME
F_LfPos	USINT
A_LfLen	USINT
A_LfFac	USINT
A_LfAdd	USINT

Таблица 221: F-выходы Функциональный блок TCP_ReceiveVar

Чтобы создать соответствующий функциональный блок **Variabel Receive** в структуре, необходимо:

1. Открыть в структуре **Configuration, Resource, Protocols, Send Receive over TCP, Function Blocks, New**.
2. Выбрать функциональный блок **Variabel Receive**.
3. Щелкнуть правой кнопкой мыши на функциональный блок **Variabel Receive** и выбрать **Edit**.
 - ☒ Открывается присвоение переменных функциональному блоку.

Соедините входы функционального блока **Variabel Receive** в структуре с теми переменными, с которыми Вы уже соединили *F-Outputs* функционального блока **TCP_ReceiveVar** в прикладной программе.

Входы	Тип
ID	INT
Lf Add	USINT
Lf Fac	USINT
Lf Len	USINT
Lf Pos	USINT
REQ	BOOL
TIMEOUT	TIME

Таблица 222: Входные системные переменные

Соедините следующие выходы функционального блока **Variabel Receive** в структуре с теми переменными, с которыми Вы уже соединили *F-Inputs* функционального блока **TCP_ReceiveVar** в прикладной программе.

Выходы	Тип
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Таблица 223: Выходные системные переменные

Данные	Описание
Переменные приема	Во вкладке <i>Process Variables</i> можно создавать любые переменные. Смещения и типы переменных должны совпадать со смещениями и типами переменных участника коммуникации.

Таблица 224: Переменные приема

Для использования функционального блока TCP_ReceiveVar необходимо:

i

Переменные приема должны создаваться во вкладке *Process Variables* диалогового окна *Variable Receive*. Смещения и типы переменных приема должны совпадать со смещениями и типами переменных передачи участника коммуникации.

1. В прикладной программе указать идентификационный номер соединения TCP на входе *A_ID*.
 2. В прикладной программе установить время ожидания приема на входе *A_Tmo*.
 3. В прикладной программе установить параметры *A_LfPos*, *A_LfLen*, *A_LfFac* и *A_LfAdd*.
 4. В прикладной программе установить вход *A_Req* на TRUE.
-

i

Функциональный блок запускает с помощью переднего фронта на *A_Req*.

Выход *A_Busy* является TRUE, пока переменные не будут приняты или не истечет время ожидания приема. Затем выход *A_Busy* переходит на FALSE и выходы *A_Valid* или *A_Error* на TRUE.

Если прием переменных осуществлен без ошибок, то выход *A_Valid* переходит на TRUE. Переменные, определенные во вкладке *Data*, можно анализировать. Выход *A_Len* содержит число байт, которые были фактически считаны.

Если прием переменных осуществлен с ошибками, то выход *A_Error* переходит на TRUE, и на выходе *A_Status* выдается код ошибки.

7.8 Панель управления (Send/Receive over TCP)

В панели управления пользователь может проверять и регулировать настройки протокола приема/передачи. Кроме того, отображается актуальная информация о состоянии (напр., нарушенное соединение и т.д.) протокола приема/передачи.

Чтобы открыть панель управления для контроля протокола приема/передачи, необходимо:

1. В структуре выбрать **Resource**.
2. В **строке меню** выбрать Online.
3. В **System Log-in** ввести данные доступа, чтобы открыть панель управления ресурса.
4. В структуре панели управления выбрать **Send/Receive Protocol**.

7.8.1 Контекстное меню (Send/Receive Protocol)

В контекстном меню выбранного протокола приема/передачи Вы можете выбрать следующие команды:

Reset:

Сбрасывает статистические данные (напр., время цикла мин., макс., и т.д.) на нуль.

7.8.2 Поле индикации (Send/Receive Protocol)

В поле индикации отображаются следующие значения выбранного протокола приема/передачи.

Элемент	Описание
Name	Протокол TCP SR
CPU Load (planned) [%]	См. главу 6.3.2.
CPU Load (actual) [%]	
Undisturbed connections	Число соединений со сбоями
Disturbed connections	Число соединений со сбоями

Таблица 225: Поле индикации протокола S&R

7.8.3 Поле индикации (Send/Receive Server)

В поле индикации отображаются следующие значения выбранного клиента Modbus.

Элемент	Описание
Name	Имя ведомого устройства Modbus.
Partner Timeout [ms]	Время ожидания, в течение которого после передачи данных минимум один пакет данных был принят участником коммуникации.
Connection State	Текущее состояние данного соединения 0x00: Соединение ОК 0x01: Соединение закрыто 0x02: Сервер ожидает установки соединения 0x04: Клиент пытается установить соединение 0x08: Соединение заблокировано
Peer Address	IP-адрес участника коммуникации.
Peer Port	Порт участника коммуникации.
Own Port	Порт данной системы управления.
Error code	Код ошибки (см. главу 7.8.4).

Таблица 226: Поле индикации ведомого устройства Modbus

7.8.4 Код ошибки соединения TCP

Коды ошибок можно считывать из переменных *Errorcode*.

На каждое конфигурированное соединение: Состояние соединения выводится из состояния соединения и кода ошибки последнего действия.

Код ошибки Десятичный	Код ошибки Шестнадцатеричный	Описание
0	16#00	ОК
4	16#04	Прерванный системный вызов
5	16#05	Ошибка ввода/вывода
6	16#06	Неизвестное устройство
9	16#09	Недействительный дескриптор сокета
12	16#0C	Нет свободной памяти
13	16#0D	В доступе отказано
14	16#0E	Недействительный адрес
16	16#10	Устройство повреждено
22	16#16	Недействительное значение (напр., в поле длины)
23	16#17	Дескрипторная таблица заполнена
32	16#20	Соединение прервано
35	16#23	Действие заблокировано
36	16#24	Действие выполняется в данный момент
37	16#25	Действие уже выполняется
38	16#27	Требуется целевой адрес
39	16#28	Сообщение слишком длинное
40	16#29	Неверный тип протокола для сокета
42	16#2A	Протокол не доступен
43	16#2B	Протокол не поддерживается
45	16#2D	Действие для сокета не поддерживается
47	16#2F	Адрес протокола не поддерживается
48	16#30	Адрес уже используется
49	16#31	Адрес не может быть присвоен
50	16#32	Сеть не работает
53	16#35	Программа прервала соединение
54	16#36	Соединение сброшено участником коммуникации
55	16#37	Недостаточно буферной памяти
56	16#38	Сокет уже соединен
57	16#39	Сокет не соединен
58	16#3A	Сокет закрыт
60	16#3C	Время для выполнения действия истекло
61	16#3D	Соединение отклонено (участником коммуникации)
65	16#41	Отсутствует запись маршрута к участнику коммуникации
78	16#4E	Функция отсутствует
254	16#FE	Возникло время ожидания
255	16#FF	Соединение закрыто участником коммуникации

Таблица 227: Коды ошибок соединения TCP

7.8.5 Таблица дополнительных кодов ошибок функциональных блоков

Коды ошибок функциональных блоков (напр., 16#8x) выдаются только на A_Status функциональных блоков S&R-TCP.

Код ошибки Десятичный	Код ошибки Шестнадцатеричный	Описание
129	16#81	Неизвестный Id соединения
130	16#82	Недопустимая длина
131	16#83	В этом соединении разрешены только циклические данные
132	16#84	Соединение в настоящее время не доступно
133	16#85	Время ожидания слишком большое
134	16#86	Внутренняя ошибка программы
135	16#87	Ошибка конфигурации
136	16#88	Передаваемые данные не подходят к конфигурированной структуре данных.
137	16#89	Функциональный блок остановлен
138	16#8A	Возникло время ожидания или передача заблокирована
139	16#8B	Функциональный блок данного вида уже активен в этом соединении

Таблица 228: Дополнительные коды ошибок

7.8.6 Состояние соединения

Код ошибки Десятичный	Код ошибки Шестнадцатеричный	Описание
0	16#00	Соединение ОК
1	16#01	Соединение закрыто
2	16#02	Сервер ожидает установки соединения
4	16#04	Клиент пытается установить соединение
8	16#08	Соединение заблокировано

Таблица 229: Состояние соединения

7.8.7 Состояние соединения участника коммуникации

Состояние протокола Десятичный	Описание
0	Отсутствует соединение
1	Соединение ОК

Таблица 230: Состояние соединения участника коммуникации

8 SNTP-Protocol (протокол SNTP)

(Simple Network Time Protocol, простой сетевой протокол времени)

С помощью протокола SNTP через Ethernet синхронизируется время клиентов SNTP посредством сервера SNTP.

Системы управления можно конфигурировать и использовать в качестве **сервера SNTP** и/или в качестве **клиента SNTP**. Действует стандарт SNTP согласно RFC 2030 (версия 4 SNTP) с тем ограничением, что поддерживается только режим Unicast.

Необходимое оснащение и требования к системе:

Элемент	Описание
Система управления	HiMax с модулем COM или только модулем ЦПУ
Активация	Данная функция по умолчанию разблокирована во всех системах HiMax.
Интерфейсы	Ethernet 10/100/1000BaseT

Таблица 231: Требования к системе и оснащение S&R TCP

8.1 SNTP Client (клиент SNTP)

Клиент SNTP всегда использует для синхронизации времени только доступный сервер SNTP с самым высоким приоритетом.

В каждом ресурсе можно конфигурировать клиента SNTP для синхронизации времени.

Чтобы создать нового клиента SNTP, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols**.
2. Щелкнуть правой кнопкой мыши на **Protocols** и выбрать в контекстном меню **New, SNTP Client**.
☒ Добавляется новый клиент SNTP.
3. В контекстном меню клиента SNTP **Properties** выбрать **COM-Modul**.

Диалоговое окно клиента SNTP содержит следующие параметры.

Элемент	Описание
Type	Клиент SNTP
Name	Имя для клиента SNTP максимум 32 символа.
Module	Выбор модуля ЦПУ или COM, на котором обрабатывается данный протокол.
Use Max CPU Load	Активировано: Сохранить предельное значение нагрузки ЦПУ из поля Max. CPU Load [%]. Деактивировано: Не использовать предельное значение нагрузки ЦПУ для данного протокола.
Max. CPU Load [%]	Максимальная нагрузка модуля ЦПУ, которая может создаваться при обработке протокола. Диапазон значений: 1...100% Значение по умолчанию: 30%
Description	Любое однозначное описание для SNTP.
Current SNTP Version	Индикация актуальной версии SNTP.

Элемент	Описание
Reference Stratum	<p>Уровень клиента SNTP определяет точность его локального времени. Чем ниже уровень, тем точнее его локальное время. Нуль означает неспецифицированный или недоступный уровень (недействительный). Актуальным используемым сервером SNTP клиента SNTP является тот сервер, который доступен и имеет самый высокий приоритет.</p> <p>Если уровень актуального сервера SNTP меньше, чем уровень клиента SNTP, то ресурс перенимает время актуального сервера SNTP.</p>
Reference Stratum (continuation)	<p>Если уровень актуального сервера SNTP больше, чем уровень клиента SNTP, то ресурс не перенимает время актуального сервера SNTP.</p> <p>Если уровень актуального сервера SNTP такой же, как уровень клиента SNTP, то необходимо различать два случая:</p> <ul style="list-style-type: none"> ▪ Если клиент SNTP (ресурс) работает исключительно в качестве клиента SNTP, то ресурс перенимает время актуального сервера SNTP. ▪ Если клиент SNTP (ресурс) работает одновременно и в качестве сервера SNTP, то на каждый запрос клиента SNTP перенимается половина разности времени для актуального сервера SNTP на ресурсе (время медленно приводится в соответствие). <p>Диапазон значений: 1...16 Значение по умолчанию: 15</p>
Client Time Request Interval [s]	<p>Интервал времени, за который осуществляется синхронизация времени актуальным сервером SNTP.</p> <p>Интервал запроса времени клиента в клиенте SNTP должен быть больше, чем время ожидания в сервере SNTP.</p> <p>Диапазон значений: 16 с...16384 с Значение по умолчанию: 16</p>

Таблица 232: Свойства клиента SNTP

8.2 Клиент SNTP (информация по серверу)

В SNTP-Server Info конфигурируется соединение с сервером SNTP.

Внутри одного клиента SNTP можно конфигурировать от 1 до 4 SNTP-Server Info.

Чтобы создать новый SNTP- Server Info, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols, SNTP Client**.
2. Щелкнуть правой кнопкой мыши на **Protocols** и выбрать в контекстном меню **New, SNTP Server Info**.
☒ Добавляется новый SNTP-Server Info.
3. В контекстном меню SNTP-Server Info **Properties** выбрать **COM-Modul**.

Диалоговое окно SNTP-Server Info содержит следующие параметры.

Элемент	Описание
Type	Информация по серверу SNTP
Name	Имя для SNTP-Server-Info. Максимум 32 символ.
Description	Описание сервера SNTP. Максимум 31 символ.
IP Address	IP-адрес ресурса или компьютера, на котором конфигурировать сервер SNTP. Значение по умолчанию: 0.0.0.0
SNTP Server Priority	Приоритет, с которым клиент SNTP обрабатывает данный сервер SNTP. Конфигурированные для клиента SNTP серверы SNTP должны иметь различные приоритеты. Диапазон значений: 0 (самый низкий приоритет) до 4294967295 (самый высокий приоритет.) Значение по умолчанию: 1
SNTP Server Timeout [s]	Время ожидания в сервере SNTP должно быть настроено на меньшее значение, чем <i>Time Request Interval</i> в клиенте SNTP. Диапазон значений: 1 с...16384 с Значение по умолчанию: 1

Таблица 233: Свойства SNTP-Server-Info

8.3 SNTP Server (сервер SNTP)

Сервер SNTP принимает запрос от клиента SNTP и передает свое актуальное время клиенту SNTP.

Чтобы создать новый сервер SNTP, необходимо:

1. В структуре открыть **Configuration, Resource, Protocols**.
2. Щелкнуть правой кнопкой мыши на **Protocols** и в контекстном меню выбрать **New, SNTP Server**.
☒ Добавляется новый сервер SNTP.
3. В контекстном меню сервера SNTP **Properties** выбрать **COM-Modul**.

Диалоговое окно сервера SNTP содержит следующие параметры.

Элемент	Описание
Type	Сервер SNTP
Name	Имя для сервера SNTP максимум 31 символ.
Module	Выбор модуля ЦПУ или COM, на котором обрабатывается данный протокол.
Use Max CPU Load	Активировано: Сохранить предельное значение нагрузки ЦПУ из поля Max. CPU Load [%]. Деактивировано: Не использовать предельное значение нагрузки ЦПУ для данного протокола.
Max. CPU Load [%]	Максимальная нагрузка модуля ЦПУ, которая может создаваться при обработке протокола. Диапазон значений: 1...100% Значение по умолчанию: 30%
Description	Описание SNTP
Current SNTP Version	Индикация актуального SNTP.
Stratum of Timeserver	Уровень сервера SNTP определяет точность его локального времени. Чем ниже уровень, тем точнее локальное время. Нуль означает неспецифицированный или недоступный уровень (недействительный). Уровень сервера SNTP должен быть ниже или равен уровню делающего запрос клиента SNTP. В противном случае время сервера SNTP не перенимается клиентом SNTP. Диапазон значений: 1...15 Значение по умолчанию: 14

Таблица 234: Свойства сервера SNTP

9 Сервер X-OPC

Сервер HIMA X-OPC выступает в качестве интерфейса передачи данных между системами управления HIMAх и сторонними системами, имеющими интерфейс OPC.

OPC означает Openess, Productivity & Collaboration и основывается на разработанной компанией Microsoft технологии (COM/DCOM). С его помощью можно соединять друг с другом систему управления процессом, системы визуализации и системы управления различных источников для обмена данными, см. также www.opcfoundation.org.

Сервер HIMA X-OPC после установки на компьютер реализуется в качестве службы Windows.

Общее конфигурирование и обслуживание сервера X-OPC проводится в SILworX. В панели управления SILworX Вы можете загрузить, запустить и остановить сервер X-OPC как систему управления.

Сервер X-OPC поддерживает следующие спецификации:

- **Доступ к данным, версии 1.0, 2.05a и 3.0**
Доступ к данным используется для передачи данных процесса от системы управления HIMAх клиенту OPC.
Вы можете передать любую глобальную переменную системы управления HIMAх клиенту OPC.
- **Аварийные сигналы и события, версия 1.10**
Используется для передачи аварийных сигналов и событий от системы управления HIMAх клиенту OPC. Любую глобальную переменную системы управления HIMAх Вы можете контролировать с регистрацией события.
События – это изменения состояния переменной, вызванные установкой или системой управления и имеющие метку времени.
Аварийные сигналы – это события, сигнализирующие о повышенной опасности.
Различают булевы и скалярные события, см. главу 9.2.3.

Необходимое оснащение и требования к системе

Элемент	Описание
Activation	Следующие лицензии могут быть активированы дополнительно: Сервер доступа к данным Сервер аварийных сигналов и событий
PC Operating System	Сервер X-OPC может работать на компьютере на базе x86 процессора со следующими операционными системами: <ul style="list-style-type: none"> ▪ Windows XP Professional (мин. пакет обновления 2) (32 бита) ▪ Windows Server 2003 (32 бита) ▪ Windows Vista Ultimate (32 бита) ▪ Windows Vista Business (32 бита)
Requirements to the host PC	Минимальные требования к хост-компьютеру: <ul style="list-style-type: none"> ▪ Pentium 4 ▪ 1 Гбайт (XP) либо 2,5 Гбайт RAM (Vista) ▪ Сетевая карта 1 Гбит/с <p>1 Минимальные требования действуют только при эксплуатации сервера X-OPC, если на хост-компьютере не работают другие приложения (напр., SILworX, Word и т.д.).</p>

Таблица 235: Требования к системе и оснащение сервера X-OPC

Свойства сервера X-OPC

Элемент	Описание
OPC server	Сервер X-OPC поддерживает функции <ul style="list-style-type: none"> ▪ OPC Data Access Custom Interface, версии 1.0, 2.05a и 3.0. ▪ OPC Alarm & Event Interfaces 1.10
Safety-related	Сервер X-OPC работает на компьютере и не является безопасным.
Interface	Рекомендация: Ethernet 1 Гбит/с
Data Exchange	Обмен данными через safeethernet .
Ethernet Network	Составляющая основу для сервера X-OPC, систем управления и клиентов OPC сеть должна иметь пропускную способность минимум 1 Гбит/с.
Global Variables	Можно использовать глобальные переменные из контекста конфигурации.
Permissible Types of Variables	Разрешены все типы данных, которые можно создавать в SILworX.
Non-permissible ASCII characters	Следующие символы зарезервированы и не могут использоваться (напр., для глобальных переменных): ! « # ',\` :
HiMax Controllers	Максимум 255 систем управления HiMax могут поддерживаться сервером X-OPC.
safeethernet connection	Сервер X-OPC может обмениваться данными по 128 кБ для каждого соединения safeethernet .
X OPC Server	На одном хост-компьютере могут эксплуатироваться 10 серверов X-OPC.
X-OPC Clients	Один сервер X-OPC может поддерживать 10 клиентов OPC.
Data Access Tags	Сервер доступа к данным поддерживает максимум 100000 тегов передачи данных. Определение: Теги: подготовленные сервером X-OPC данные. Теги соответствуют определенным глобальным переменным. Элементы: запрошенные клиентом OPC данные.
Alarm & Event Event definitions	Сервер аварийных сигналов и событий X-OPC поддерживает максимально 100000 определений событий.

Таблица 236: Свойства сервера X-OPC

Свойства системы управления HIMax

Элемент	Описание
safeethernet connection	HIMax может выполнять обмен данными общим размером 128 кБ для одного соединения safeethernet с сервером X-OPC. За один цикл HIMax на сервер X-OPC передается только один вид. (Вид – это фрагмент из 1100 кБ).
Max. number of event definitions	В одной системе управления HIMax можно определить максимум 20000 системных событий и 6000 событий входа/выхода.
Event memory size	Энергонезависимый буфер событий процессорного модуля HIMax регистрирует максимум 5000 событий. Если буфер заполнен, то новые события больше не сохраняются, пока запись событий не будет считана минимум одним сервером аварийных сигналов и событий X-OPC и тем самым маркирована для перезаписи.
Alarm & Event Timestamp	Для каждого события можно выбрать источник события. Определенные как ЦПУ события создаются на процессорном модуле. Он создает события полностью в каждом своем цикле. Таким образом, значение каждой глобальной переменной может регистрироваться и оцениваться как событие. Определенные как I/O Event события могут создаваться только на модулях ввода/вывода SOE (напр., AI 32 02 или DI 32 04). Он создает события полностью в каждом своем цикле.
Max. number of X-OPC A&E Server	Доступ в систему управления HIMax могут получить максимум 4 сервера аварийных сигналов и событий X-OPC, они могут одновременно считывать события из буфера событий процессорного модуля.

9.1 Конфигурирование соединения сервера OPC

На этом примере сконфигурировано избыточное соединение сервера X-OPC с системой управления HIMax.

Серверы X-OPC предоставляют переменные процесса и значения событий системы управления HIMax для клиентов OPC. Клиенты OPC имеют доступ к предоставленным переменным процесса и значениям событий и отображают их на своем пользовательском интерфейсе.

9.1.1 Необходимое программное обеспечение:

- SILworX
- X-OPC-Server
- OPC-Client

9.1.2 Условия для работы сервера X-OPC:

- Сеть Ethernet должна иметь пропускную способность 1 Гбит/с.
- Время системы компьютера/сервера должно быть синхронизировано, напр., с помощью SNTP.
- Убедитесь, что наборы данных для Data Access (доступ к данным) и Alarm & Events (аварийные сигналы и события) в системе управления, на серверах X-OPC и в клиентах OPC подходят друг к другу.

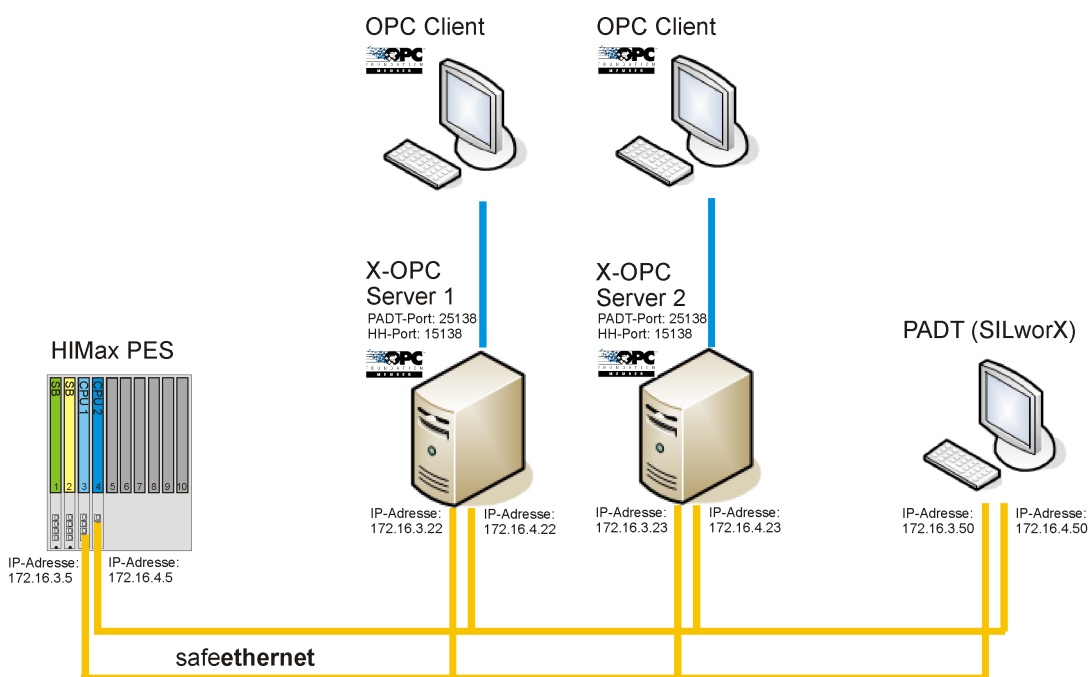


Рисунок 71: Избыточная эксплуатация X-OPC

9.1.3 Установка сервера OPC

Сервер X-OPC должен устанавливаться на соответствующем хост-компьютере. Запустите файл *X-OPC.exe* на соответствующем хост-компьютере и следуйте указаниям программы по установке.

i

Запомните ID системы и номер порта PADT. Они потребуются для генерирования лицензионного ключа!

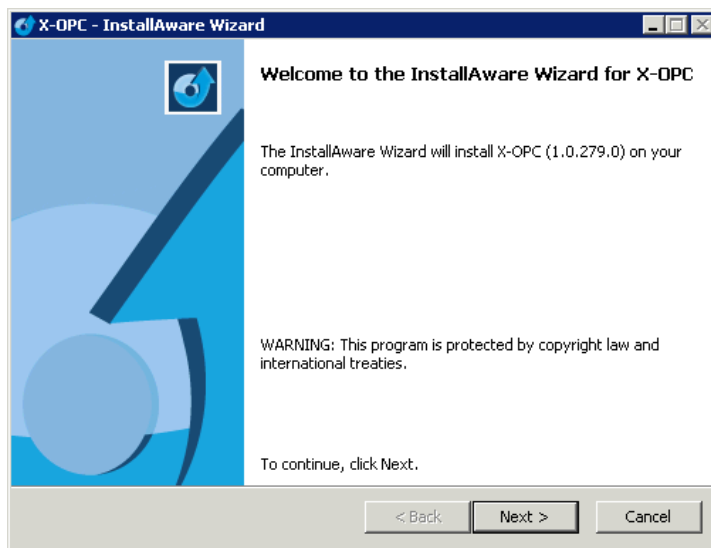


Рисунок 72: Программа по установке сервера X-OPC

Чтобы конфигурировать сервер X-OPC на первом хост-компьютере, необходимо:

Следуйте указаниям программы по установке.

1. Ввести следующие данные для сервера X-OPC:
 - ID системы: 100
 - Порт PADT: 25138
 - Произвольное имя сервера X-OPC (отображается в клиенте OPC).
2. Для установки сервера X-OPC щелкните на **Next>**.

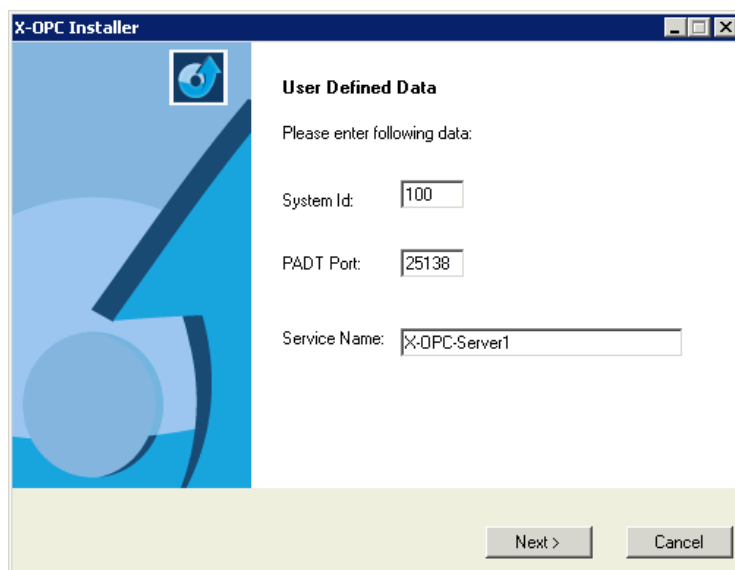


Рисунок 73: Программа по установке сервера X-OPC

Чтобы конфигурировать сервер X-OPC на втором хост-компьютере, необходимо:

Следуйте указаниям программы по установке.

1. Ввести следующие данные для сервера X-OPC:
 - ID системы: 110
 - Порт RAPT: 25138

i

Порт RAPT и порт HH второго сервера X-OPC могут совпадать с первым, если серверы X-OPC эксплуатируются на различных компьютерах.

- Произвольное имя сервера X-OPC (отображается в клиенте OPC).
2. Для установки сервера X-OPC щелкните на **Next>**.

9.1.4 Конфигурирование интерфейса DCOM

Если клиент OPC и сервер OPC работают не на одном компьютере, то Вы должны адаптировать интерфейс DCOM.

Для этого необходимо выполнить шаги, описанные в руководстве OPC Foundation. *Использование OPC посредством DCOM с Microsoft Windows XP пакет обновления 2 версия 1.10* (см. www.opcfoundation.org).

Автоматический запуск сервера X-OPC после перезапуска компьютера:

1. В Windows открыть **Start, Settings, Control Panel, Administration, Services** и выбрать из списка **X-OPC Server**.
2. В контекстном меню сервера X-OPC выбрать **Properties**.
3. Во вкладке **General** выбрать тип запуска **Automatic**.

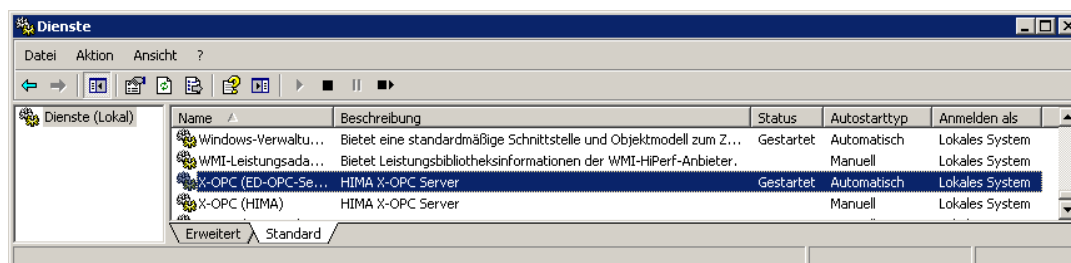


Рисунок 74: Автоматический запуск настройки для сервера X-OPC

Чтобы проверить, работает ли сервер X-OPC на компьютере, необходимо:

1. Открыть *Windows Task Manager* и выбрать вкладку *Processes*.
2. Проверить, выполняется ли процесс *X-OPC.exe* на компьютере.

9.1.5 Создание сервера OPC в SILworX

Чтобы создать новый набор серверов OPC, необходимо:

1. В структуре открыть **Configuration**.
2. Щелкнуть правой кнопкой мыши на **Configuration** и выбрать в контекстном меню **New, OPC Server Set**.
☒ Добавляется новый набор серверов OPC.
3. В контекстном меню набора серверов OPC выбрать **Properties** и сохранить значения по умолчанию.

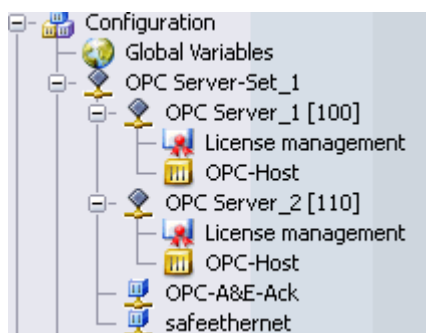


Рисунок 75: Избыточная эксплуатация X-OPC

Чтобы создать первый сервер OPC, необходимо:

1. В структуре выбрать **Configuration, OPC Server Set**.
2. Щелкнуть правой кнопкой мыши на **OPC Server Set** и выбрать в контекстном меню **New, OPC Server**.
☒ Добавляется новый сервер OPC.
3. Щелкнуть правой кнопкой мыши на **OPC Server** и выбрать в контекстном меню **Properties**.
☒ Ввести ID системы [SRS] (напр., 100).
3. Щелкнуть правой кнопкой мыши на **OPC Host** и выбрать в контекстном меню **Edit**.
☒ Открывается диалоговое окно OPC-Host для конфигурирования интерфейсов IP
4. Щелкнуть правой кнопкой мыши на пустое место в диалоговом окне OPC-Host и выбрать в контекстном меню **New IP Device**.
 - настроить порт PADT (напр., 25138).
 - IP-адрес компьютера, на котором установлен сервер X-OPC (напр., 172.16.3.22).
 - IP-адрес компьютера, на котором установлен сервер X-OPC (напр., 172.16.4.22).
 - Отметить как интерфейс по умолчанию.
 - настроить порт HH (напр., 15138).

В том же самом наборе серверов OPC создается избыточный сервер OPC.

Чтобы создать избыточный сервер OPC, необходимо:

1. В структуре выбрать **Configuration, OPC Server Set**.
2. Щелкнуть правой кнопкой мыши на **OPC Server Set** и выбрать в контекстном меню **New, OPC Server**.
 - ☒ Добавляется новый сервер OPC.
3. Щелкнуть правой кнопкой мыши на **OPC Server** и выбрать в контекстном меню **Properties**.
 - ☒ Ввести ID системы [SRS] (напр., 110).
5. Щелкнуть правой кнопкой мыши на **OPC Host** и выбрать в контекстном меню **Edit**.
 - ☒ Открывается диалоговое окно OPC-Host для конфигурирования интерфейсов IP
6. Щелкнуть правой кнопкой мыши на пустое место в диалоговом окне OPC-Host и выбрать в контекстном меню **New IP Device**.
 - настроить порт PADT (напр., 25138).
 - IP-адрес компьютера, на котором установлен сервер X-OPC (напр., 172.16.3.123).
 - IP-адрес компьютера, на котором установлен сервер X-OPC (напр., 172.16.4.123).
 - Отметить как интерфейс по умолчанию.
 - настроить порт HH (напр., 15138).

i

Если на компьютере установлено сетевое устройство защиты, то в конфигурации сетевого устройства защиты в качестве исключения должны быть занесены порты TCP/UDP PADT и HH сервера X-OPC.

9.1.6 Настройки сервера OPC в редакторе safeethernet

Чтобы создать соединение между сервером OPC и ресурсом (системой управления HlMax), необходимо:

1. Открыть в наборе серверов OPC **safeethernet Editor**
2. В выборе объекта щелкнуть на **Resource** и с помощью Drag&Drop перетащить на свободное место в рабочей области редактора **safeethernet**.
3. Для Alarm & Events (аварийные сигналы и события) по умолчанию активирован параметр **Activate SER**.

	IF CH 1 (local)	IF CH 2 (local)	IF CH 1 (target)	IF CH 2 (target)	Profile	Sync/Async
1					Fast & Noisy	ASYN
2	110.x.x (122.16.4.23:15138)	110.x.x (122.16.4.23:15138)	2.0.15 (172.16.4.5:6010)	2.0.15 (172.16.4.5:6010)		
3	100.x.x (172.16.3.22:15138)	100.x.x (172.16.3.22:15138)	2.0.5 (172.16.3.5:6010)	2.0.5 (172.16.3.5:6010)		

Рисунок 76: Избыточная эксплуатация X-OPC

i

Используемые интерфейсы Ethernet компьютера отображены в столбце **IF Ch1 (local)**. Интерфейсы Ethernet системы управления HlMax должны выбираться в столбце **IF Ch1 (target)**.

Значения по умолчанию параметров **safeethernet** для связи с сервером X-OPC рассчитаны на максимальную возможность доступа.

Receive Timeout = 1000 мс, Response Time = 500 мс и т.д.

Более подробная информация по параметрам **safeethernet**, см. главу 4.6.1.

9.1.7 Конфигурация в качестве сервера доступа к данным X-OPC

Чтобы открыть детальный вид соединения safeethernet, необходимо:

Условие: Редактор safeethernet сервера OPC должен быть открыт.

1. Щелкнуть правой кнопкой мыши на строку **Resource**, чтобы открыть контекстное меню ресурса.
2. В контекстном меню выбрать **Detail View**, чтобы открыть детальный вид соединения safeethernet.
3. Щелкнуть на вкладку **View definitions**.
4. Щелкнуть правой кнопкой мыши на свободное место в рабочей области и выбрать **New View Definition**.
В столбце Приоритет выполняется настройка, как часто передается данный вид по отношению к другим видам (вид представляет собой фрагмент размером 1100 кБ).
5. Щелкнуть на вкладку **OPC Server Set<->Resource**.

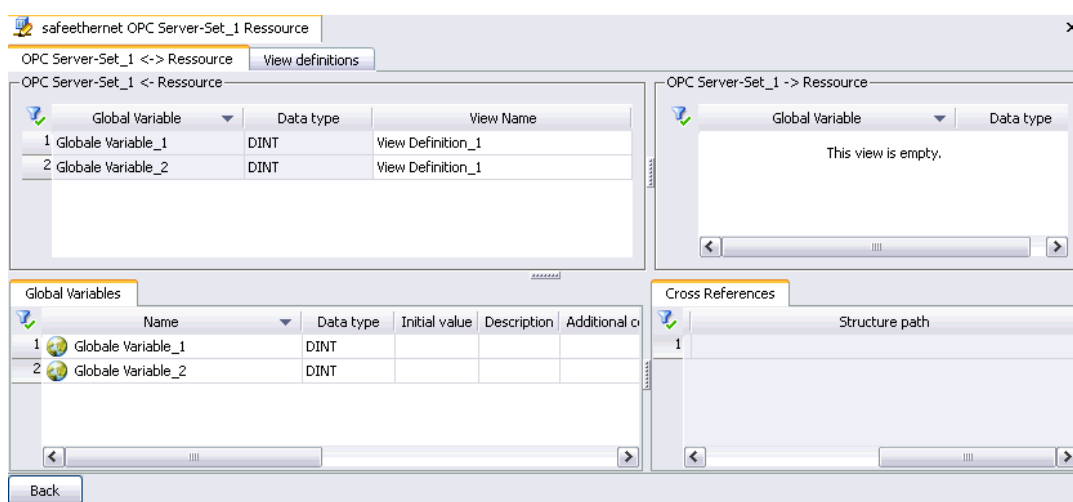


Рисунок 77: Детальный вид соединения safeethernet

Чтобы добавить переменные приема OPC, необходимо:

Переменные приема OPC передаются от ресурса на сервер OPC

1. В выборе объекта выбрать **Global Variable** и с помощью Drag&Drop перетащить в столбец **OPC Server Set <-Resource-**.
2. Щелкнуть два раза на столбец **View Name** и выбрать созданное ранее **View Definition**.
3. Повторить данное действие для других переменных приема OPC.

Чтобы добавить переменные передачи OPC, необходимо:

Переменные передачи OPC передаются от сервера OPC ресурсу

1. В выборе объекта выбрать **Global Variable** и с помощью Drag&Drop перетащить в столбец **OPC Server Set->Resource**.
2. Щелкнуть два раза на столбец **View Name** и выбрать созданное ранее **View Definition**.
3. Повторить данное действие для других переменных передачи OPC.

i

Переменные приема и передачи OPC должны создаваться в наборе серверов только один раз. Они автоматически используются обоими серверами X-OPC в наборе серверов OPC.

Генерирование кода и подтверждение набора серверов OPC:

1. В структуре выбрать **Configuration, OPC Server Set**.
2. Щелкнуть на **Code Generation** в строке меню и подтвердить с помощью **OK**.
3. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.

Загрузить сгенерированный код на сервер X-OPC:

1. Щелкнуть правой кнопкой мыши на **OPC Server** и выбрать в контекстном меню **Online** для **System Log-in** (входа в систему).
2. Ввод данных доступа:
 - IP-адрес компьютера, на котором установлен сервер X-OPC (напр., 172.16.3.123).
 - User name: Administrator
 - Password: отсутствует
 - Rights: Administrator
3. Щелкнуть на **Login**, чтобы открыть панель управления.
4. В строке меню SILworX щелкнуть на символ **Resource Download**.
☒ Выполняется загрузка кода на сервер X-OPC.
5. Щелкните в строке меню SILworX на символ **Resource Cold Start**.
☒ Сервер X-OPC работает.

Откройте клиента OPC:

Отображаемое в клиенте OPC имя сервера X-OPC состоит из:

HIMA (изготовитель).**Service name (имя службы)** (см. главу 9.1.3)-**DA** (Data Access: доступ к данным).

Установите соединение с сервером X-OPC. Конфигурированные данные Data Access должны теперь передаваться клиенту OPC.

9.1.8 Конфигурация в качестве сервера аварийных сигналов и событий X-OPC

На этом примере конфигурируется сервер аварийных сигналов и событий X-OPC для системы управления HIMax.

Сервер аварийных сигналов и событий X-OPC регистрирует события через **safeethernet** от системы управления HIMax и предоставляет их клиенту OPC. Клиент OPC имеет доступ к предоставленным переменным событий и размещает их на своем пользовательском интерфейсе.

Чтобы создать Alarm&Events (аварийные сигналы и события), необходимо:

1. В структуре открыть **Configuration, Resource**.
2. Щелкнуть правой кнопкой мыши на **Resource** и выбрать в контекстном меню **New, Alarm & Events**.
☒ Добавлен новый объект Alarm & Events (аварийные сигналы и события).
3. Щелкнуть правой кнопкой мыши на **Alarm & Events** и выбрать **Edit**.
4. Выбрать вкладку **Scalar Event Definition** для скалярных событий.
5. В выборе объекта щелкнуть на **Global Variable** и с помощью Drag&Drop перетащить на свободное место в рабочей области редактора аварийных сигналов и событий.
6. Конфигурирование событий, см. таблица

Столбец	Описание	Диапазон значений
---------	----------	-------------------

и таблица Откройте в Windows St.

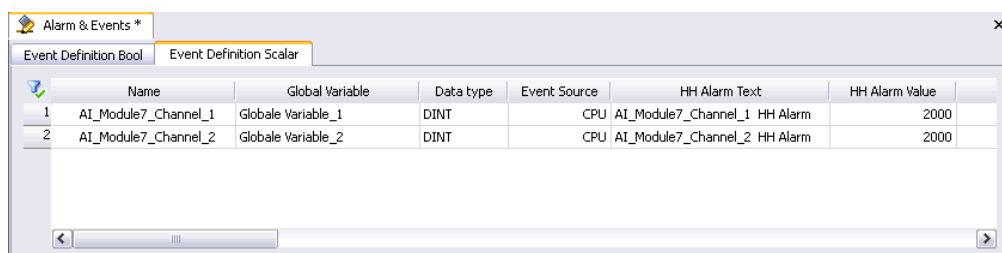


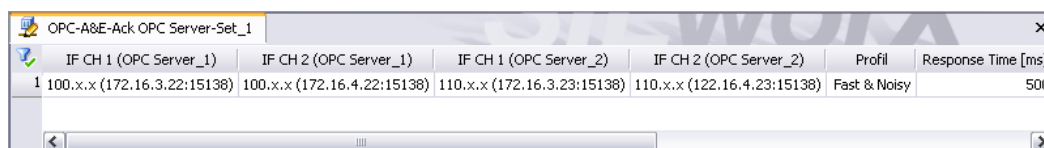
Рисунок 78: Редактор аварийных сигналов и событий

Чтобы создать соединение подтверждения между обоими серверами аварийных сигналов и событий X-OPC, необходимо:

i

Если два сервера аварийных сигналов и событий X-OPC эксплуатируются с избыточностью, то подтверждения приема для подтверждения (квитирования) аварийных сигналов на обоих серверах X-OPC могут синхронизироваться. Для этого создается соединение подтверждения.

1. В структуре выбрать **Configuration, OPC Server Set, New**.
2. Щелкнуть правой кнопкой мыши на **OPC Server Set** и выбрать в контекстном меню **New, OPC-A&E-Ack**.
3. В диалоговом окне OPC-A&E-Ack выбрать следующие IP-соединения.
 - IF CH1 (сервер OPC 1, напр., 172.16.3.122).
 - IF CH2 (сервер OPC 1, напр., 172.16.4.122).
 - IF CH1 (сервер OPC 2, напр., 172.16.3.123).
 - IF CH2 (сервер OPC 2, напр., 172.16.4.123).



	IF CH 1 (OPC Server_1)	IF CH 2 (OPC Server_1)	IF CH 1 (OPC Server_2)	IF CH 2 (OPC Server_2)	Profil	Response Time [ms]
1	100.x.x (172.16.3.22:15138)	100.x.x (172.16.4.22:15138)	110.x.x (172.16.3.23:15138)	110.x.x (122.16.4.23:15138)	Fast & Noisy	500

Рисунок 79: Избыточная эксплуатация X-OPC

Генерирование кода и подтверждение набора серверов OPC:

1. В структуре выбрать **Configuration, OPC Server Set**.
2. Щелкнуть на **Code Generation** в строке меню и подтвердить с помощью **OK**.
4. Тщательно проверить записи в индикации состояний, при необходимости исправить ошибки.

Загрузить сгенерированный код на сервер X-OPC:

1. Щелкнуть правой кнопкой мыши на **OPC Server** и выбрать в контекстном меню **Online** для входа в систему.
2. Ввод данных доступа:
 - IP-адрес компьютера, на котором установлен сервер X-OPC (напр., 172.16.3.123).
 - Имя пользователя: Администратор
 - Пароль: отсутствует
 - Права: Администратор
3. Щелкнуть на **Login**, чтобы открыть панель управления.
5. В строке меню SILworX щелкнуть на символ **Resource Download**.
 - ☒ Выполняется загрузка кода на сервер OPC.
6. Щелкните в строке меню SILworX на символ **Resource Cold Start**.
 - ☒ Сервер OPC работает.

Откройте клиента OPC:

Отображаемое в клиенте OPC имя сервера X-OPC состоит из:

HIMA (изготовитель).**Service name (имя службы)** (см. главу 9.1.3)-**AE (Alarm&Event: аварийные сигналы и события)**

Установите соединение с сервером X-OPC. Конфигурированные Alarme & Events (аварийные сигналы и события) должны теперь передаваться клиенту OPC.

i

Если система управления и сервер аварийных сигналов и событий X-OPC соединены, то сервер аварийных сигналов и событий X-OPC необходимо сначала синхронизировать. Для этого сервер аварийных сигналов и событий X-OPC считывает со всех переменных, которые определены как событие, текущее состояние и передает имеющиеся аварийные сигналы клиенту OPC. В клиенте OPC появляется отображение состояния системы управления. Считывание событий начинается только с этого момента.

После синхронизации сервера X-OPC с системой управления в клиенте OPC обновляются все события. Записи событий с более ранними метками времени переписываются на текущие считанные состояния переменных событий.

9.2 Параметрирование

9.2.1 Набор серверов OPC

Набор серверов OPC выступает в качестве общей основы для параметрирования одного или двух серверов OPC.

Переменные приема и передачи OPC создаются в редакторе safeethernet набора серверов OPC пользователем один раз и автоматически становятся идентичными для обоих избыточных серверов X-OPC.

Чтобы создать новый набор серверов OPC, необходимо:

1. Откройте в структуре **Configuration**.
2. Выберите в контекстном меню конфигурации **New, OPC Server Set**, чтобы добавить новый набор серверов OPC.
3. Сохраните значения по умолчанию в контекстном меню набора серверов OPC **Properties**.

Диалоговое окно набора серверов OPC содержит следующие параметры.

Элемент	Описание
Name	Имя набора серверов OPC. Максимум 31 символ.
Safety time [ms]	<p>Безопасное время – это время в миллисекундах, в течение которого сервер X-OPC должен среагировать на ошибку.</p> <p>Условие: Безопасное время $\geq 2 \times$ время сторожевого устройства</p> <p>Диапазон значений: 2000...400000 мс</p> <p>Значение по умолчанию: 20 000 мс</p>
Watchdog time [ms]	<p>Время сторожевого устройства – это максимальное время в миллисекундах, которое может потребоваться серверу X-OPC для выполнения цикла программы. Если заданное время сторожевого устройства превышает (выполнение цикла программы длится слишком долго), то сервер X-OPC останавливается.</p> <p>Условие: Время сторожевого устройства ≥ 1000 мс и $\leq 0,5 \times$ безопасное время</p> <p>Диапазон значений: 1000...200000 мс</p> <p>Значение по умолчанию: 10000 мс</p>

Элемент	Описание
Main Enable	<p>Настройка кнопки-флажка OPC «Главная деблокировка» влияет на функцию других кнопок-флажков OPC.</p> <p>Если «Главная деблокировка» выключена, то настройки других кнопок-флажков OPC не могут быть изменены в то время, пока прикладная программа выполняет обработку (система управления в состоянии RUN).</p> <p>Значение по умолчанию: Activated</p>
Autostart	<p>Автоматический запуск определяет, могут ли автоматически запускаться конфигурации OPC после включения или после начальной загрузки сервера OPC посредством холодного пуска, теплого пуска или запуск не может быть осуществлен (выкл).</p> <p>Если автоматический запуск деактивирован, то после начальной загрузки сервер X-OPC переходит в состояние STOP/VALID CONFIGURATION.</p> <p>Значение по умолчанию: Deactivated</p>
Start allowed	<p>Только если активирован <i>Start Allowed</i>, сервер X-OPC может запуститься с программирующего устройства.</p> <p>Если <i>Start Allowed</i> деактивирован, Вы не можете запустить сервер X-OPC с программирующего устройства. Вы можете запустить сервер X-OPC только в том случае, если <i>Autostart</i> активирован и Вы включите хост-компьютер и выполните начальную загрузку.</p> <p>Если не активированы ни <i>Autostart</i>, ни <i>Start Allowed</i>, сервер X-OPC больше не запуститься. Это может потребоваться при выполнении работ по техобслуживанию, чтобы предотвратить запуск устройства.</p> <p>Значение по умолчанию: Activated</p>
Loading allowed	<p>Если функция <i>Load Allowed</i> деактивирована, то в систему управления невозможно загрузить (новую) конфигурацию OPC.</p> <p>Деактивируйте <i>Load Allowed</i>, если Вы не хотите, чтобы в сервере X-OPC могла быть переписана загруженная конфигурация OPC.</p> <p>Значение по умолчанию: Activated</p>
Reload allowed	Функция отсутствует!
Global Forcing Allowed	<p>Только если активирована функция <i>Global Forcing Allowed</i>, Вы можете запустить <i>Start Global Forcing</i>.</p> <hr/> <p>i Редактор инициализации может быть вызван для индикации содержимого переменных даже в том случае, если деактивирована функция <i>Global Forcing Allowed</i>.</p> <hr/> <p>Значение по умолчанию: Deactivated</p>

Элемент	Описание
Global Forcing Timeout Reaction	<p>Если настроено <i>Global Force Timeout Reaction, Stop Resource</i>, то после завершения времени инициализации сервер X-OPC переходит в состояние STOP. Все выходы сервера X-OPC установлены на LOW.</p> <p>Если настроен <i>Global Force Timeout Reaction, Stop Forcing Only</i>, то сервер X-OPC продолжает выполнение конфигурации OPC и после того, как истечет время инициализации.</p> <hr/> <p>i Если инициализация разрешена, тщательно проверьте настройку для остановки во время ожидания инициализации. Соблюдайте также указания в руководстве по функциональной безопасности.</p> <hr/> <p>Значение по умолчанию: остановить ресурс</p>
Max.Com. Time Slice ASYNC [ms]	<p><i>Max. Com. Time Slice ASYNC [ms]</i> – это время в миллисекундах, которое резервируется для каждого цикла сервера X-OPC, чтобы обработать все имеющиеся коммуникационные задачи для одноранговой связи.</p> <p>Значение по умолчанию: 10 мс</p>
Period [ms]	<p>Период ЦПУ сервера X-OPC</p> <p>Значение по умолчанию: 50 мс</p>
Delimiter of name space	<p>Точка .</p> <p>Косая /</p> <p>Двоеточие :</p> <p>Обратная \</p> <p>косая</p> <p>Значение по умолчанию: Точка</p>
Namespace type	<p>В зависимости от требования клиента OPC можно настроить тип пространства имен:</p> <ul style="list-style-type: none"> Иерархическое пространство имен Плоское пространство имен <p>Значение по умолчанию: Иерархическое пространство имен</p>
Short tag names for DA	<p>Только если выбрано <i>Flat Namespace</i>, можно активировать данный параметр.</p> <p>Это опция, при которой данные и события предлагаются клиенту OPC без дополнительного контекста (имя пути).</p> <p>Значение по умолчанию: Deactivated</p>
Changeless update	<p>Настройка по требованию клиента OPC.</p> <p>Активировано: Если <i>Changeless Update</i> активировано, то сервер X-OPC по истечении «OPC Group-UpdateRate» всегда передает клиенту OPC все элементы.</p> <p>Деактивировано: Если <i>Changeless Update</i> деактивировано, клиенту OPC передаются только измененные значения (такой порядок действий соответствует спецификации OPC).</p>

Элемент	Описание
Cycle delay [ms]	Задержка цикла ограничивает нагрузку ЦПУ компьютера посредством сервера X-OPC, чтобы выполнялись также другие программы. Диапазон значений: 1...100 мс Значение по умолчанию: 5 мс
Simple Events for CPU/IO Events	Никогда Только при запуске Всегда
Short Tag Names for I&O	Только если выбрано <i>Flat Namespace</i> , можно активировать данный параметр. Это опция, при которой данные и события предлагаются клиенту OPC без дополнительного контекста (имя пути). Значение по умолчанию: Deactivated

Таблица 237: Свойства

9.2.2 Сервер OPC

Чтобы создать новый сервер OPC, необходимо:

1. Откройте в структуре **Configuration, OPC Server Set**.
2. Выберите в контекстном меню набора серверов OPC **New, OPC Server**, чтобы добавить новый сервер OPC.
3. Выберите в контекстном меню сервера OPC **Properties**.

Диалоговое окно сервера OPC содержит следующие параметры.

Элемент	Описание
Name	Имя сервера OPC. Максимум 31 символ.
System ID [SRS]	Значение по умолчанию: 60000

Таблица 238: Свойства

Чтобы открыть хост OPC, необходимо:

1. Откройте в структуре **Configuration, OPC Server Set, OPC Server**.
2. Выберите в контекстном меню хоста OPC **Edit**, чтобы открыть обзор интерфейсов IP.

Диалоговое окно хоста OPC содержит следующие параметры.

Элемент	Описание
PADT port	Значение по умолчанию: 25138
Name	Имя набора серверов OPC. Максимум 31 символ.
IP address	IP-адрес хост-компьютера. Значение по умолчанию: 192.168.0.1
Standard interface	Должно быть активировано, если хост-компьютер имеет больше одного порта Ethernet. Значение по умолчанию: Activated
HN port	Значение по умолчанию: 15138

Таблица 239: Редактирование (Edit)

9.2.3 Alarm & Events (аварийные сигналы и события)

Редактор аварийных сигналов и событий предназначен для параметрирования аварийных сигналов и событий системы управления HIMax.

HIMax различает булевы и скалярные события.

Булевы события:

- Изменения булевых переменных, например, для цифровых входов.
- Аварийное и обычное состояние, Вы можете присваивать их состояниям переменных по вашему желанию

Скалярные события:

- Переходы за предельные значения, которые определены для скалярной переменной, например, для аналогового входа.
- Возможно два верхних и два нижних предела.
Для предельных значений должно действовать следующее:
самый верхний предел \geq верхний предел \geq обычный диапазон \geq нижний предел \geq самый нижний предел.
Гистерезис будет действовать в следующих случаях:
 - При понижении значения ниже верхнего предела
 - При превышении значения нижнего предела
 При указании гистерезиса Вы можете избежать отображения ненужного большого количества событий, если глобальная переменная сильно отличается от предельного значения.

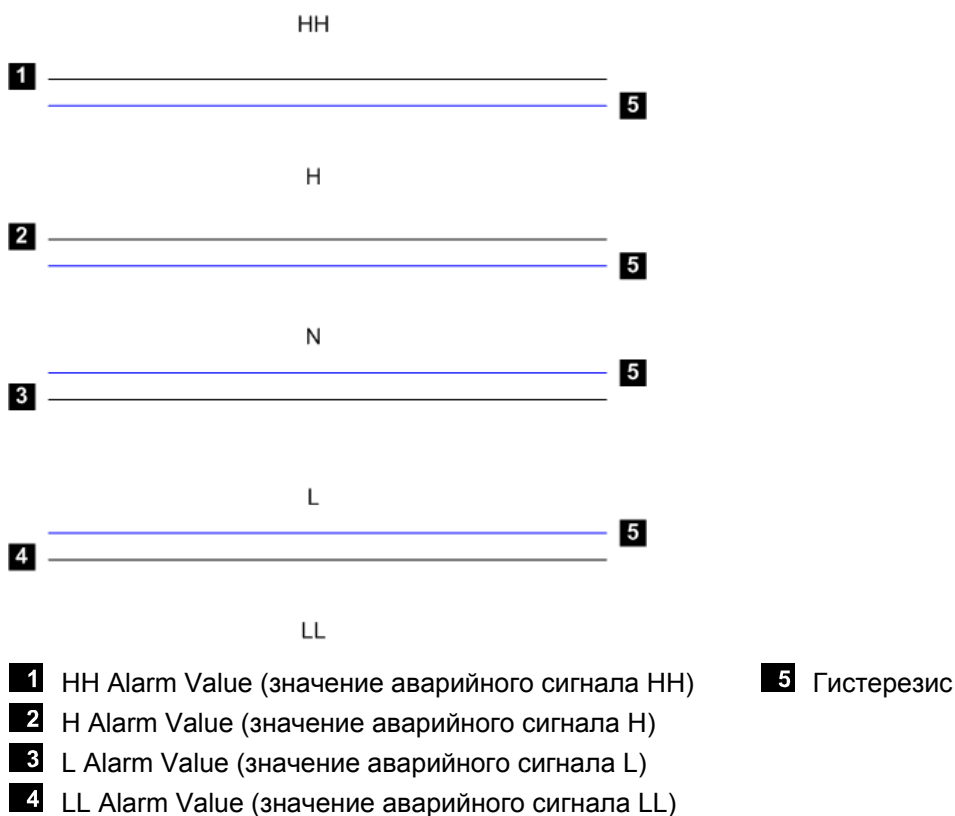


Рисунок 80: Пять областей скалярного события

Чтобы создать **Alarm&Events** (аварийные сигналы и события), необходимо:

1. В структуре открыть **Configuration, Resource**.
2. Щелкнуть правой кнопкой мыши на **Resource** и выбрать в контекстном меню **New, Alarm & Events**.
☒ Добавлен новый объект Alarm & Events (аварийные сигналы и события).
3. Щелкнуть правой кнопкой мыши на **Alarm & Events** и выбрать **Edit**.
4. Выбрать вкладку **Scalar Event Definition** для скалярных событий.
5. В выборе объекта щелкнуть на **Global Variable** и с помощью Drag&Drop перетащить на свободное место в рабочей области редактора аварийных сигналов и событий.
6. Конфигурирование событий, см. таблица 240 и таблица 241.

Параметры булевых событий внесите в таблицу, содержащую следующие столбцы:

Столбец	Описание	Диапазон значений
Name	Имя определения события	Текст, макс. 31 знака
Global variable	Имя присвоенной глобальной переменной (введено, например, с помощью Drag&Drop)	
Data type	Тип данных глобальной переменной, не изменяется	BOOL
Event source	CPU Event Метка времени создается на процессорном модуле. Он создает события полностью в каждом своем цикле. IO Event Метка времени создается на подходящем модуле ввода/вывода (напр., DI 32 04). Auto Event Создается событие ЦПУ и, при наличии, события ввода/вывода модулей ввода/вывода. Значение по умолчанию: Событие ЦПУ	Событие ЦПУ, событие ввода/вывода, событие Auto
Alarm when FALSE	Activated Изменение значения TRUE->FALSE глобальных переменных инициирует событие Deactivated Изменение значения FALSE->TRUE глобальных переменных инициирует событие Значение по умолчанию: деактивировано	Кнопка-флажок активирована, деактивирована
Alarm Text	Текст, описывающий аварийное состояние	Текст
Alarm Severity	Приоритет аварийного состояния Значение по умолчанию: 1	1...1000
Alarm Ack Required	Activated Требуется подтверждение аварийного состояния оператором (квитирование) Deactivated Подтверждение аварийного состояния оператором не требуется Значение по умолчанию: деактивировано	Кнопка-флажок активирована, деактивирована
Return to Normal Text	Текст, описывающий аварийное состояние	Текст
Return to Normal Severity	Приоритет нормального состояния	1...1000
Return to Ack Required	Требуется подтверждение нормального состояния оператором (квитирование) Значение по умолчанию: деактивировано	Кнопка-флажок активирована, деактивирована

Таблица 240: Параметры для булевых событий

Параметры скалярных событий внесите в таблицу, содержащую следующие столбцы:

Столбец	Описание	Диапазон значений
Name	Имя определения события	Текст, макс. 31 знака
Global variable	Имя назначенной глобальной переменной (введено, например, с помощью Drag&Drop)	
Data type	Тип данных глобальной переменной, не изменяется.	Зависит от типа глобальных переменных
Event source	<p>CPU Event Метка времени создается на процессорном модуле. Он создает события полностью в каждом своем цикле.</p> <p>IO Event Метка времени создается на подходящем модуле ввода/вывода (напр., AI 32 02).</p> <p>Auto Event Создается событие ЦПУ и, при наличии, события ввода/вывода модулей ввода/вывода.</p> <p>Значение по умолчанию: Событие ЦПУ</p>	Событие ЦПУ, событие ввода/вывода, событие Auto
HH Alarm Text	Текст, описывающий аварийное состояние максимального верхнего предельного значения	Текст
HH Alarm Value	Максимальное верхнее предельное значение, которое инициировало событие. Условие: $(HH \text{ Alarm Value} - Hysteresis) > H \text{ Alarm Value}$ или $HH \text{ Alarm Value} = H \text{ Alarm Value}$	Зависит от типа глобальных переменных
HH Alarm Severity	Приоритет максимального верхнего предельного значения, значение по умолчанию: 1	1...1000
HH Ack Required	<p>Activated Оператор должен подтвердить превышение максимального верхнего предельного значения (квитирование).</p> <p>Deactivated Оператор не должен подтверждать превышение максимального верхнего предельного значения.</p> <p>Значение по умолчанию: деактивировано</p>	Кнопка-флажок активирована, деактивирована
H Alarm Text	Текст, описывающий аварийное состояние верхнего предельного значения	Текст
H Alarm Value	Верхнее предельное значение, которое инициировало событие. Условие: $(H \text{ Alarm Value} - Hysteresis) > (L \text{ Alarm Value} + Hysteresis)$ или $H \text{ Alarm Value} = L \text{ Alarm Value}$	зависит от типа глобальных переменных
H Alarm Severity	Приоритет верхнего предельного значения, значение по умолчанию: 1	1...1000
H Ack Required	<p>Activated Оператор должен подтвердить превышение верхнего предельного значения (квитирование).</p> <p>Deactivated Оператор не должен подтверждать превышение верхнего предельного значения.</p> <p>Значение по умолчанию: деактивировано</p>	Кнопка-флажок активирована, деактивирована
Return to Normal Text	Текст, описывающий аварийное состояние	Текст
Return to Normal Severity	Приоритет нормального состояния, значение по умолчанию: 1	1...1000
Return to Normal Ack Required	Требуется подтверждение нормального состояния оператором (квитирование), значение по умолчанию: деактивировано	Кнопка-флажок активирована, деактивирована
L Alarm Text	Текст, описывающий аварийное состояние нижнего предельного значения	Текст

Столбец	Описание	Диапазон значений
L Alarm Value	Нижнее предельное значение, которое инициировало событие. Условие: $(L \text{ Alarm Value} + \text{Hysteresis}) < (H \text{ Alarm Value} - \text{Hysteresis})$ или $L \text{ Alarm Value} = H \text{ Alarm Value}$	зависит от типа глобальных переменных
L Alarm Severity	Приоритет нижнего предельного значения, значение по умолчанию: 1	1...1000
L Ack Required	Activated Оператор должен подтвердить падение ниже нижнего предельного значения (квитирование). Deactivated Оператор не должен подтверждать падение ниже нижнего предельного значения. Значение по умолчанию: деактивировано	Кнопка-флажок активирована, деактивирована
LL Alarm Text	Текст, описывающий аварийное состояние максимального нижнего предельного значения	Текст
LL Alarm Value	Максимальное нижнее предельное значение, которое инициировало событие. Условие: $(LL \text{ Alarm Value} + \text{Hysteresis}) < (L \text{ Alarm Value})$ или $LL \text{ Alarm Value} = L \text{ Alarm Value}$	зависит от типа глобальных переменных
LL Alarm Severity	Приоритет максимального нижнего предельного значения, значение по умолчанию: 1	1...1000
LL Ack Required	Activated Оператор должен подтвердить падение ниже максимального нижнего предельного значения (квитирование). Deactivated Оператор не должен подтверждать падение ниже максимального нижнего предельного значения. Значение по умолчанию: деактивировано	Кнопка-флажок активирована, деактивирована
Alarm Hysteresis	Гистерезис предотвращает постоянное создание большого количества событий, если значение процесса часто колеблется у предельного значения.	зависит от типа глобальных переменных

Таблица 241: Параметры для скалярных событий

9.3 Удаление сервера X-OPC

Чтобы удалить сервер X-OPC, необходимо:

1. Откройте в Windows **Start, Settings, Control Panel, Software**.
2. Выберите в списке сервер X-OPC, который Вы хотите удалить, и щелкните на **Remove**.
3. Следуйте указаниям программы для удаления.

10 Конфигурирование функциональных блоков

Протоколы полевой шины и соответствующие функциональные блоки работают на модуле COM системы управления HIMA. Поэтому эти функциональные блоки должны быть созданы в структуре SILworX в **Configuration, Resource, Protocols...**

Чтобы управлять данными функциональными блоками на модуле COM, в прикладной программе SILworX могут создаваться функциональные блоки (см. главу 10.1), которые можно использовать в качестве функциональных блоков по умолчанию.

Соединение функциональных блоков в прикладной программе SILworX с соответствующими функциональными блоками в структуре SILworX осуществляется через общие переменные. Они должны быть созданы пользователем заранее в редакторе переменных.

10.1 Создание библиотек функциональных блоков

Библиотеки функциональных блоков для PROFIBUS DP и TCP Send/Receive добавляются в проект с помощью функции *Restore...* (контекстное меню проекта).

Библиотеку функциональных блоков Вы можете получить по запросу от службы поддержки HIMA.

Тел.: +49-(0)6202-709 -185

-259

-261

Эл. почта: support@hima.com

10.2 Конфигурирование функциональных блоков в прикладной программе

Необходимые функциональные блоки могут копироваться с помощью Drag&Drop в прикладную программу. Конфигурируйте входы и выходы, используя описание соответствующего функционального блока.

Верхняя часть функционального блока

Верхняя часть функционального блока соответствует пользовательскому интерфейсу, с помощью которого функциональный блок управляется прикладной программой.

Здесь соединяются переменные, которые используются в прикладной программе. Префикс «А» означает «Application» (приложение).

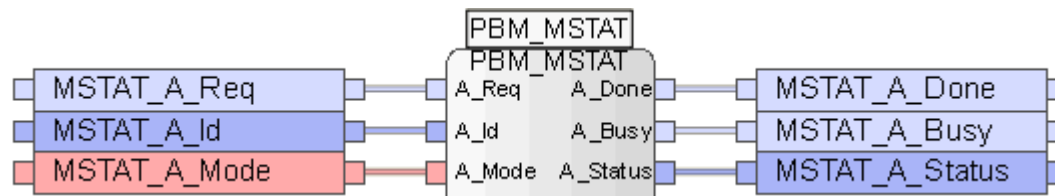


Рисунок 81: Функциональный блок PNM_MSTST (верхняя часть)

Нижняя часть функционального блока

Нижняя часть функционального блока отображает соединение с функциональным блоком (в структуре SILworX).

Здесь соединяются переменные, которые должны быть соединены с функциональным блоком в структуре SILworX. Префикс «F» означает «Field» (поле).

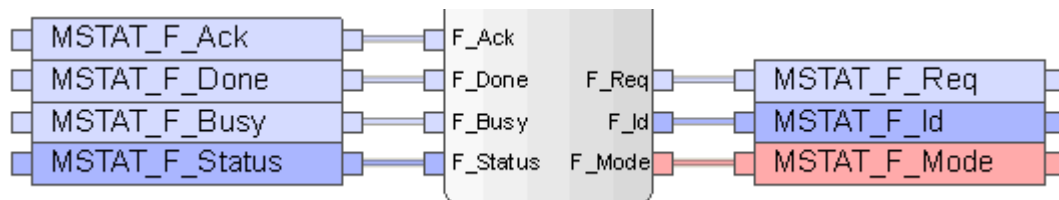


Рисунок 82: Функциональный блок PNM_MSTST (нижняя часть)

10.3 Конфигурирование функциональных блоков в структуре SILworX

Чтобы конфигурировать функциональный блок в структуре SILworX, необходимо:

1. Выбрать в структуре **Configuration, Resource, Protocols**, напр., **PROFIBUS Master**.
2. Щелкните правой кнопкой мыши на **Function Blocks** и выберите **New**.
3. Выберите подходящий функциональный блок (в структуре SILworX).

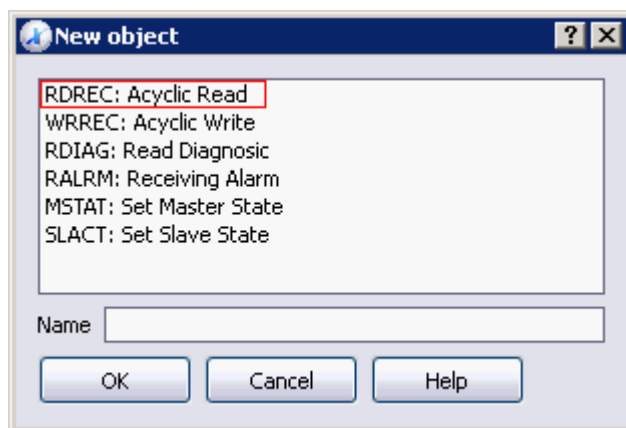


Рисунок 83: Выбор функциональных блоков

Входы функционального блока (галочка в столбце «Входная переменная») должны быть связаны с теми переменными, которые связаны с *F-Outputs* функционального блока в прикладной программе.

Выходы функционального блока (галочка в столбце «Входная переменная» отсутствует) должны быть связаны с теми переменными, которые связаны с *F-Inputs* функционального блока в прикладной программе.

System Variables				
F	Name	Data type	Input variable	Global Variable
1	ACK	BOOL	<input checked="" type="checkbox"/>	MSTAT_F_Ack
2	BUSY	BOOL	<input checked="" type="checkbox"/>	MSTAT_F_Busy
3	DONE	BOOL	<input checked="" type="checkbox"/>	MSTAT_F_Done
4	M_ID	DWORD	<input type="checkbox"/>	MSTAT_F_Id
5	MODE	INT	<input type="checkbox"/>	MSTAT_F_Mode
6	REQ	BOOL	<input type="checkbox"/>	MSTAT_F_Req
7	STATUS	DWORD	<input checked="" type="checkbox"/>	MSTAT_F_Status

Рисунок 84: Системные переменные функционального блока MSTAT

Приложение

Глоссарий

Понятие	Описание
ARP	Address Resolution Protocol: сетевой протокол для присвоения сетевых адресов аппаратным адресам
AI	Analog Input, аналоговый вход
без реактивного воздействия	Если к одному и тому же источнику (напр., трансмиттеру) подключены два входных контура. В этом случае входной контур обозначается как контур «без реактивного воздействия», если он не искажает сигналы другого входного контура.
COM	Модуль связи
CRC	Cyclic Redundancy Check, контрольная сумма
DI	Digital Input, цифровой вход
DO	Digital Output, цифровой выход
ЭМС	Электромагнитная совместимость, ЭМС
EN	Европейские стандарты
ESD	ElectroStatic Discharge, электростатический разряд
FB	Полевая шина
FBS	Язык функциональных блоковых диаграмм (FBD)
FTA	Field Termination Assembly
FTZ	Время отказоустойчивости
ICMP	Internet Control Message Protocol: сетевой протокол для сообщений о состоянии и сбоях
МЭК	Международные стандарты по электрооборудованию
MAC-адрес	Аппаратный адрес сетевого подключения (Media Access Control)
PADT	Инструмент программирования и отладки (согл. МЭК 61131-3), ПК с SILworX
PE	Защитное заземление
PELV	Protective Extra Low Voltage: низкое напряжение с безопасным размыканием (ЗСНН)
PES	Программируемая электронная система, ПЭС
PFD	Probability of Failure on Demand: вероятность ошибки при запросе функции безопасности
PFH	Probability of Failure per Hour: вероятность сбоя в течение одного часа
Плата сопряжения	Плата сопряжения для модуля HIMax
R	Read (чтение)
Rack-ID	Идентификация несущего каркаса (номер)
R/W	Read/Write (чтение/запись)
SB	Системная шина (модуль)
SELV	Safety Extra Low Voltage: безопасное сверхнизкое напряжение (БСНН)
SFF	Safe Failure Fraction, доля безопасно устраняемых неполадок
SIL	Safety Integrity Level, уровень совокупности безопасности (согл. МЭК 61508)
SILworX	Программа для программирования для HIMax
SNTP	Simple Network Time Protocol, простой сетевой протокол времени (RFC 1769)
SRS	System.Rack.Slot
SW	Программное обеспечение
TMO	Время ожидания
W	Запись
WD	Watchdog, сторожевое устройство
WDZ	Время сторожевого устройства

Список изображений

Рисунок 1: Диалоговое окно конфигурации для модуля ЦПУ и COM в SILworX	21
Рисунок 2: Системные структуры	31
Рисунок 3: Создание конфигурации избыточного соединения.....	32
Рисунок 4: Структура ресурса.....	32
Рисунок 5: Значения параметров избыточного соединения safeethernet.....	33
Рисунок 6: Детальный вид редактора safeethernet.....	33
Рисунок 7: Избыточное соединение между двумя системами управления HIMax	41
Рисунок 8: Конфигурированное как избыточное соединение двух систем управления HIMax через провод.....	41
Рисунок 9: Соединение safeethernet двух систем управления HIMax	45
Рисунок 10: Соединение safeethernet системы HIMax с системой управления HIMatrix	46
Рисунок 11: Соединение safeethernet с удаленными устройствами ввода/вывода	46
Рисунок 12: Соединение safeethernet двух систем HIMax с одной системой управления HIMatrix	47
Рисунок 13: Соединение safeethernet между ресурсом A1 в проекте A и ресурсом B1 в проекте B	52
Рисунок 14: Вариант с проектом A в качестве локального проекта	53
Рисунок 15: Вариант с проектом B в качестве локального проекта	53
Рисунок 16: Установка связи между SILworX и ELOP II Factory	54
Рисунок 17: Ресурс прокси HIMatrix	55
Рисунок 18: Параметры соединения safeethernet для ресурса прокси.....	56
Рисунок 19: Экспорт соединения Safeethernet.....	57
Рисунок 20: Импорт соединений в ELOP II Factory.....	58
Рисунок 21: Редактор P2P в ELOP II Factory.....	58
Рисунок 22: Присвоение сигналов передачи в ELOP II Factory.....	59
Рисунок 23: Присвоение сигналов приема в ELOP II Factory	59
Рисунок 24: Панель управления для контроля соединения	60
Рисунок 25: Связь через PROFIBUS DP	64
Рисунок 26: Ведомое устройство HIMax PROFIBUS DP с модулями.....	67
Рисунок 27: Поле данных пользователя.....	70
Рисунок 28: Диалоговое окно Verification.....	70
Рисунок 29: Свойства ведущего устройства PROFIBUS DP.....	71
Рисунок 30: Свойства ведомого устройства PROFIBUS DP	72
Рисунок 31: Изохронный цикл PROFIBUS DP	81
Рисунок 32: Диалоговое окно Edit User Parameters	92
Рисунок 33: Функциональный блок MSTAT	94
Рисунок 34: Функциональный блок RALRM.....	97
Рисунок 35: Функциональный блок RDIAG	101
Рисунок 36: Функциональный блок RDREC.....	105
Рисунок 37: Функциональный блок SLACT.....	108
Рисунок 38: Функциональный блок WRREC.....	111
Рисунок 39: Вспомогательный функциональный блок ACTIVE	114
Рисунок 40: Вспомогательный функциональный блок ALARM	115
Рисунок 41: Вспомогательный функциональный блок DEID	116
Рисунок 42: Вспомогательный функциональный блок ID	117
Рисунок 43: Вспомогательный функциональный блок NSLOT	118
Рисунок 44: Вспомогательный функциональный блок SLOT.....	118
Рисунок 45: Вспомогательный функциональный блок STDDIAG.....	119
Рисунок 46: Связь через Modbus TCP	134
Рисунок 47: Массив переменных в стандартном режиме Modbus	134
Рисунок 48: Сеть Modbus	147
Рисунок 49: Шлюз Modbus.....	150
Рисунок 50: Последовательный Modbus	153
Рисунок 51: Телеграмма Modbus.....	153
Рисунок 52: Сохранение переменных в режиме совместимости Modbus	169
Рисунок 53: Индекс переменных	169
Рисунок 54: Сохранение переменных в стандартном режиме Modbus	170
Рисунок 55: Соединение HIMax и системы управления Siemens.....	178
Рисунок 56: Передача данных между HIMax и системой управления Siemens.....	179
Рисунок 57: Список переменных в блоке UDT1 Siemens	180
Рисунок 58: Список переменных в блоке DB1 Siemens	181

Рисунок 59: Редактор символов SIMATIC	181
Рисунок 60: Структурная схема для приема	182
Рисунок 61: Структурная схема для передачи.....	183
Рисунок 62: Свойства соединения TCP в SILworX	184
Рисунок 63: Список переменных Siemens	193
Рисунок 64: Список переменных HlMax.....	193
Рисунок 65: Функциональный блок TCP_Reset.....	195
Рисунок 66: Функциональный блок TCP_Send.....	198
Рисунок 67: Функциональный блок TCP_Receive	201
Рисунок 68: Функциональный блок TCP_ReceiveLine	205
Рисунок 69: Функциональный блок TCP_ReceiveVar	209
Рисунок 70: Структура пакета данных	210
Рисунок 71: Избыточная эксплуатация X-OPC	224
Рисунок 72: Программа по установке сервера X-OPC	225
Рисунок 73: Программа по установке сервера X-OPC	225
Рисунок 74: Автоматический запуск настройки для сервера X-OPC	226
Рисунок 75: Избыточная эксплуатация X-OPC	227
Рисунок 76: Избыточная эксплуатация X-OPC	228
Рисунок 77: Детальный вид соединения safeethernet	229
Рисунок 78: Редактор аварийных сигналов и событий	231
Рисунок 79: Избыточная эксплуатация X-OPC	232
Рисунок 80: Пять областей скалярного события	238
Рисунок 81: Функциональный блок PNM_MSTST (верхняя часть).....	242
Рисунок 82: Функциональный блок PNM_MSTST (нижняя часть).....	243
Рисунок 83: Выбор функциональных блоков	243
Рисунок 84: Системные переменные функционального блока MSTAT	244

Список таблиц

Таблица 1: Дополнительные руководства	9
Таблица 2: Нормы для ЭМС, климатических и экологических требований	12
Таблица 3: Общие условия	12
Таблица 4: Климатические условия	13
Таблица 5: Механические испытания	14
Таблица 6: Испытания на помехоустойчивость	14
Таблица 7: Испытания на помехоустойчивость	14
Таблица 8: Испытания на эмиссию помех	14
Таблица 9: Дополнительная проверка характеристик подачи постоянного напряжения	15
Таблица 10: Количество стандартных протоколов на модуль	18
Таблица 11: Протоколы на модуле COM 1	19
Таблица 12: Протоколы на модуле COM 3	19
Таблица 13: Свойства интерфейсов Ethernet	20
Таблица 14: Параметр конфигурации	23
Таблица 15: Параметр сетевого коммутатора Ethernet	24
Таблица 16: Параметры маршрута	24
Таблица 17: Интерфейсы полевой шины	25
Таблица 18: Назначение штырьковых выводов подключений D-SUB FB1 и FB2 для PROFIBUS DP	26
Таблица 19: Назначение штырьковых выводов подключений D-SUB FB1 и FB2 для Modbus	26
Таблица 20: Назначение штырьковых выводов подключений D-SUB FB1 и FB2 для RS232	26
Таблица 21: Протоколы связи	27
Таблица 22: Примеры для номеров изделия модуля COM	28
Таблица 23: Параметр Протокол safeethernet	36
Таблица 24: Вкладка Системные переменные в редакторе safeethernet	40
Таблица 25: Доступные интерфейсы Ethernet	40
Таблица 26: Комбинации для соединений safeethernet	41
Таблица 27: Сетевые профили safeethernet	48
Таблица 28: Поле индикации соединения safeethernet	61
Таблица 29: Необходимое оснащение и требования к системе	63
Таблица 30: Свойства ведущего устройства PROFIBUS DP	63
Таблица 31: Выходы ведомого устройства HIMA PROFIBUS DP	65
Таблица 32: Входы ведомого устройства HIMA PROFIBUS DP	65
Таблица 33: Переменные модуля ввода [000] DP Input/ELOP Export: 2 Bytes	68
Таблица 34: Переменные модуля ввода [001] DP Input/ELOP Export: 8 Bytes	68
Таблица 35: Переменные модуля ввода [002] DP Input/ELOP Export: 1 Byte	69
Таблица 36: Переменные модуля вывода [003] DP Output/ELOP Import: 2 Bytes	69
Таблица 37: Переменные модуля вывода [004] DP Output/ELOP Import: 1 Byte	69
Таблица 38: Системные переменные ведущего устройства PROFIBUS DP	74
Таблица 39: Общие свойства ведущего устройства PROFIBUS DP	74
Таблица 40: Вкладка Время в диалоговом окне свойств ведущего устройства PROFIBUS DP	76
Таблица 41: Вкладка ЦПУ/COM в диалоговом окне свойств ведущего устройства PROFIBUS DP	76
Таблица 42: Прочие свойства ведущего устройства PROFIBUS DP	77
Таблица 43: Ориентировочные значения HIMAх для контрольного времени обращения маркера при различной скорости передачи	78
Таблица 44: Время передачи для символа при различной скорости передачи	79
Таблица 45: Элементы для расчета контрольного времени обращения маркера	79
Таблица 46: Системные переменные ведомого устройства PROFIBUS DP	84
Таблица 47: Вкладка Параметры ведомого устройства PROFIBUS DP	85
Таблица 48: Вкладка Группы в диалоговом окне свойств ведомого устройства PROFIBUS DP	85
Таблица 49: Вкладка DP-V1 в диалоговом окне свойств ведомого устройства PROFIBUS DP	86
Таблица 50: Вкладка Аварийные сигналы в диалоговом окне свойств ведомого устройства PROFIBUS DP	86
Таблица 51: Вкладка Данные в диалоговом окне свойств ведомого устройства PROFIBUS DP	87
Таблица 52: Вкладка Модель в диалоговом окне свойств ведомого устройства PROFIBUS DP	87
Таблица 53: Вкладка Свойства в диалоговом окне свойств ведомого устройства PROFIBUS DP	88
Таблица 54: Вкладка Скорость передачи в диалоговом окне свойств ведомого устройства PROFIBUS DP	88

Таблица 55: Вкладка Ациклическая передача в диалоговом окне свойств ведомого устройства PROFIBUS DP	89
Таблица 56: Файл GSD ведомого устройства PROFIBUS DP от HIMax	90
Таблица 57: Пример: Блоки 1...4 поля данных пользователя	92
Таблица 58: Пример: Блоки 1...4 поля данных пользователя	92
Таблица 59: Обзор функциональных блоков PROFIBUS DP	93
Таблица 60: А-входы Функциональный блок MSTAT	94
Таблица 61: А-выходы Функциональный блок MSTAT	95
Таблица 62: F-входы Функциональный блок MSTAT	95
Таблица 63: F-выходы Функциональный блок MSTAT	95
Таблица 64: Входные системные переменные	96
Таблица 65: Выходные системные переменные	96
Таблица 66: А-входы Функциональный блок RALRM	98
Таблица 67: А-выходы Функциональный блок RALRM	98
Таблица 68: F-входы Функциональный блок RALRM	98
Таблица 69: F-выходы Функциональный блок RALRM	99
Таблица 70: Входные системные переменные	99
Таблица 71: Выходные системные переменные	99
Таблица 72: Аварийные данные	100
Таблица 73: А-входы Функциональный блок RALRM	101
Таблица 74: А-выходы Функциональный блок RALRM	101
Таблица 75: F-входы Функциональный блок RDIAG	102
Таблица 76: F-выходы Функциональный блок RDIAG	102
Таблица 77: Входные системные переменные	102
Таблица 78: Выходные системные переменные	103
Таблица 79: Диагностические данные	103
Таблица 80: А-входы Функциональный блок RDREC	105
Таблица 81: А-выходы Функциональный блок RDREC	106
Таблица 82: F-входы Функциональный блок RDREC	106
Таблица 83: F-выходы Функциональный блок RDREC	106
Таблица 84: Входные системные переменные	107
Таблица 85: Выходные системные переменные	107
Таблица 86: Данные	107
Таблица 87: А-входы Функциональный блок SLACT	108
Таблица 88: А-выходы Функциональный блок SLACT	109
Таблица 89: F-входы Функциональный блок SLACT	109
Таблица 90: F-выходы Функциональный блок SLACT	109
Таблица 91: Входные системные переменные	110
Таблица 92: Выходные системные переменные	110
Таблица 93: А-входы Функциональный блок WRREC	111
Таблица 94: А-выходы Функциональный блок WRREC	111
Таблица 95: F-входы Функциональный блок WRREC	112
Таблица 96: F-выходы Функциональный блок WRREC	112
Таблица 97: Входные системные переменные	112
Таблица 98: Выходные системные переменные	113
Таблица 99: Данные	113
Таблица 100: Обзор вспомогательных функциональных блоков	114
Таблица 101: Входы Вспомогательный функциональный блок ACTIVE	114
Таблица 102: Выходы Вспомогательный функциональный блок ACTIVE	114
Таблица 103: Входы Вспомогательный функциональный блок ALARM	115
Таблица 104: Выходы Вспомогательный функциональный блок ALARM	116
Таблица 105: Входы Вспомогательный функциональный блок DEID	116
Таблица 106: Выходы Вспомогательный функциональный блок DEID	116
Таблица 107: Входы Вспомогательный функциональный блок ID	117
Таблица 108: Выходы Вспомогательный функциональный блок ID	117
Таблица 109: Входы Вспомогательный блок NSLOT	118
Таблица 110: Выходы Вспомогательный блок NSLOT	118
Таблица 111: Входы Вспомогательный блок SLOT	119
Таблица 112: Выходы Вспомогательный блок SLOT	119
Таблица 113: Входы Вспомогательный блок STDDIAG	120
Таблица 114: Выходы Вспомогательный блок STDDIAG	120
Таблица 115: Коды ошибки функциональных блоков	121

Таблица 116: Поле индикации ведущего устройства PROFIBUS	123
Таблица 117: Состояние ведущего устройства PROFIBUS DP	124
Таблица 118: Поведение ведущего устройства PROFIBUS DP	124
Таблица 119: Светодиод FBx (ведущее устройство PROFIBUS DP)	125
Таблица 120: Оснащение и требования к системе ведомого устройства HIMA PROFIBUS DP	126
Таблица 121: Свойства ведомого устройства HIMA PROFIBUS DP	126
Таблица 122: Системные переменные ведомого устройства PROFIBUS DP	128
Таблица 123: Свойства ведомого устройства: Вкладка Общее	130
Таблица 124: Поле индикации (PROFIBUS DP Slave)	131
Таблица 125: Светодиод FBx (PROFIBUS DP Slave)	131
Таблица 126: Оснащение и требования к системе ведущего устройства Modbus	133
Таблица 127: Свойства ведущего устройства Modbus	133
Таблица 128: Системные переменные ведущего устройства Modbus	138
Таблица 129: Общие свойства ведущего устройства Modbus	139
Таблица 130: Параметры COM/ЦПУ	140
Таблица 131: Функции Modbus	141
Таблица 132: Телеграмма-запрос Read Coils (01)	143
Таблица 133: Телеграмма-запрос Read Discrete Inputs (02)	143
Таблица 134: Телеграмма-запрос Read Holding Registers	144
Таблица 135: Телеграмма-запрос Read Input Registers	144
Таблица 136: Вкладка Read Write Holding	145
Таблица 137: Телеграмма-запрос Write Multiple Coils (15)	146
Таблица 138: Телеграмма-запрос Write Multiple Registers (16)	146
Таблица 139: Телеграмма-запрос Write Single Coil (05)	146
Таблица 140: Телеграмма-запрос Write Single Register	146
Таблица 141: Системные переменные ведомого устройства TCP/UDP	148
Таблица 142: Параметр конфигурации	149
Таблица 143: Параметры соединения шлюза Modbus	152
Таблица 144: Переменные состояния ведомого устройства-шлюза	152
Таблица 145: Параметры соединения ведомого устройства-шлюза	152
Таблица 146: Параметры последовательного ведущего устройства Modbus	154
Таблица 147: Системные переменные ведомого устройства Modbus	154
Таблица 148: Параметры соединения ведущего устройства Modbus	155
Таблица 149: Поле индикации ведущего устройства Modbus	156
Таблица 150: Светодиод FBx у ведущего устройства Modbus	157
Таблица 151: Оснащение и требования к системе ведомого устройства Modbus HIMA	158
Таблица 152: Свойства ведомого устройства Modbus	158
Таблица 153: Вкладка «Общее» для ведомого устройства Modbus HIMA	160
Таблица 154: Вкладка Serial Interface для ведомого устройства Modbus HIMA	161
Таблица 155: Вкладка TCP and UDP Ports для ведомого устройства Modbus HIMA	161
Таблица 156: Вкладка CPU/COM для ведомого устройства Modbus HIMA	162
Таблица 157: Вкладка «Системные переменные» для ведомого устройства Modbus HIMA	163
Таблица 158: Функции Modbus ведомого устройства Modbus HIMA	165
Таблица 159: Считывание Var6 до Var8 посредством функции 01 Modbus	169
Таблица 160: Телеграмма-ответ функции 01	169
Таблица 161: Сортированные переменные с адресацией	171
Таблица 162: Считывание переменных, начиная с Var4, с помощью функции 03	171
Таблица 163: Телеграмма-ответ функции 03	171
Таблица 164: Битовые переменные с адресацией	171
Таблица 165: Считывание переменных Var14 до Var21 с помощью функции 01	172
Таблица 166: Телеграмма-ответ функции 01	172
Таблица 167: Вкладка Смещение для ведомого устройства Modbus HIMA	172
Таблица 168: Регистровые переменные в битовой области с адресацией	173
Таблица 169: Считывание переменных Var6 до Var8 с помощью функции 01	173
Таблица 170: Телеграмма-ответ на функцию 01	173
Таблица 171: Отображаемые битовые переменные в регистровой области	173
Таблица 172: Считывание переменных Var22 до Var23 с помощью функции 03	174
Таблица 173: Телеграмма-ответ на функцию 03	174
Таблица 174: Поле индикации ведомого устройства Modbus	175
Таблица 175: Поле индикации данных ведущего устройства	175
Таблица 176: Светодиод FBx у ведомого устройства Modbus	176
Таблица 177: Коды ошибок TCP/IP Modbus	176

Таблица 178: Требования к системе и оснащение S&R TCP	177
Таблица 179: Свойства S&R TCP	177
Таблица 180: Конфигурация системы управления HIMax	179
Таблица 181: Конфигурация Siemens SIMATIC 300	179
Таблица 182: Глобальные переменные	184
Таблица 183: Переменные для принимаемых данных	185
Таблица 184: Переменные для передаваемых данных	185
Таблица 185: Системные переменные S&R TCP	186
Таблица 186: Общие свойства S&R TCP	186
Таблица 187: Параметры COM/ЦПУ	187
Таблица 188: Системные переменные	188
Таблица 189: Свойства соединения S&R TCP	190
Таблица 190: Функциональные блоки для соединений S&R TCP	194
Таблица 191: А-входы Функциональный блок TCP_Reset	195
Таблица 192: А-выходы Функциональный блок TCP_Reset	195
Таблица 193: F-входы Функциональный блок TCP_Reset	196
Таблица 194: F-выходы Функциональный блок TCP_Reset	196
Таблица 195: Входные системные переменные	196
Таблица 196: Выходные системные переменные	197
Таблица 197: А-входы Функциональный блок TCP_Send	198
Таблица 198: А-выходы Функциональный блок TCP_Send	198
Таблица 199: F-входы Функциональный блок TCP_Send	199
Таблица 200: F-выходы Функциональный блок TCP_Send	199
Таблица 201: Входные системные переменные	199
Таблица 202: Выходные системные переменные	200
Таблица 203: Переданные данные	200
Таблица 204: А-входы Функциональный блок TCP_Receive	201
Таблица 205: А-выходы Функциональный блок TCP_Receive	202
Таблица 206: А-входы Функциональный блок TCP_Receive	202
Таблица 207: F-выходы Функциональный блок TCP_Receive	202
Таблица 208: Входные системные переменные	203
Таблица 209: Выходные системные переменные	203
Таблица 210: Переменные приема	203
Таблица 211: А-входы Функциональный блок TCP_ReceiveLine	205
Таблица 212: А-выходы Функциональный блок TCP_ReceiveLine	206
Таблица 213: F-входы Функциональный блок TCP_ReceiveLine	206
Таблица 214: F-выходы Функциональный блок TCP_ReceiveLine	206
Таблица 215: Входные системные переменные	207
Таблица 216: Выходные системные переменные	207
Таблица 217: Переменные приема	207
Таблица 218: А-входы Функциональный блок TCP_ReceiveVar	210
Таблица 219: А-выходы Функциональный блок TCP_ReceiveVar	211
Таблица 220: F-входы Функциональный блок TCP_ReceiveVar	211
Таблица 221: F-выходы Функциональный блок TCP_ReceiveVar	211
Таблица 222: Входные системные переменные	212
Таблица 223: Выходные системные переменные	212
Таблица 224: Переменные приема	212
Таблица 225: Поле индикации протокола S&R	214
Таблица 226: Поле индикации ведомого устройства Modbus	214
Таблица 227: Коды ошибок соединения TCP	215
Таблица 228: Дополнительные коды ошибок	216
Таблица 229: Состояние соединения	216
Таблица 230: Состояние соединения участника коммуникации	216
Таблица 231: Требования к системе и оснащение S&R TCP	217
Таблица 232: Свойства клиента SNTP	218
Таблица 233: Свойства SNTP-Server-Info	219
Таблица 234: Свойства сервера SNTP	220
Таблица 235: Требования к системе и оснащение сервера X-OPC	221
Таблица 236: Свойства сервера X-OPC	222
Таблица 237: Свойства	236
Таблица 238: Свойства	237
Таблица 239: Редактирование (Edit)	237

Приложение

По связи

Таблица 240:	Параметры для булевых событий	239
Таблица 241:	Параметры для скалярных событий	241

Алфавитный указатель

Условия использования	
Защита от воздействия	
электростатического разряда	17
климатические	14

механические	16
Подача напряжения	16
ЭМС	16

HI 801 062 R

© 2009 HIMA Paul Hildebrandt GmbH

HIMax и SILworX являются зарегистрированными товарными знаками фирмы:

HIMA Paul Hildebrandt GmbH

Albert-Bassermann-Str. 28

68782 Brühl, Германия

Тел: +49 6202 709-0

Факс: +49 6202 709-107

HIMax-info@hima.com

www.hima.com



SAFETY
NONSTOP