

PBM-ETH-3.0

Руководство пользователя

V1.0



Оглавление

Глава 1. Обзор продукции	5
1. Назначение PBM-ETH-3.0	5
2. Особенности PBM-ETH-3.0	6
3. Технические характеристики PBM-ETH-3.0	6
Глава 2. Внешний вид, индикация и разъёмы	8
2.1 Расположение элементов устройства и индикация	8
2.2 Конструкция и габариты	13
2.3 Назначение DIP-переключателей	13
2.4 Режимы работы портов Ethernet	15
Глава 3. Установка и подключение	16
3.1 Установка на DIN-рейку	16
3.2 Разъём интерфейса PROFIBUS и его установка	16
3.3 Подключение питания	17
3.4 Подключение кабелей интерфейсов Ethernet и PROFIBUS	18
Глава 4. Принцип работы PBM-ETH-3.0	19
Глава 5. Начало работы с PBM-ETH-3.0	20
Глава 6. Настройка с помощью конфигурационного ПО PBConfi	21
6.1 Настройка с помощью PBConfi в режиме офлайн	22
6.1.1 Создание нового проекта	22
6.1.2 Обновление каталога устройств	23
6.1.4 Добавление Ведомого устройства (Slave)	25
6.1.4.1 Настройка параметров PROFIBUS DPV1	26
6.1.4.2 Настройка сторожевого таймера (Watchdog)	26
6.1.4.3 Настройка пользовательских параметров Ведомого устройства	26
6.1.4.4 Настройка Sync/Freeze и группирования	26
6.1.5 Настройка параметров шины PROFIBUS	28
6.1.6 Настройка IP-адреса для подключения ПО PBConfi	28
6.1.7 Загрузка конфигурации	29
6.1.8 Настройка сетевых параметров PBM-ETH-3.0	29
6.1.9 Сохранение и загрузка файла конфигурации PBM-ETH-3.0	30
Глава 7. Онлайн-отладка с помощью PBConfi	31
7.1 Работа с функциями Master-устройства	31
7.2 Работа с функциями Slave-устройства (функции DPV1)	32
7.3 Мониторинг состояния устройств, подключённых к шлюзу	32
7.4 Мониторинг обмена I/O данными PROFIBUS DPV0	33
7.5 Экспорт текущего состояния всех Ведущих и Ведомых устройств системы	33
7.6 Изменение адреса Ведомого устройства (специальные Slave-устройства)	34
7.7 Просмотр системного журнала (log)	34
Глава 8. Обновление встроенного ПО (прошивки)	35
Глава 9. Мониторинг состояния ведомых устройств PROFIBUS	36
9.1 Область входных регистров	36
9.2 Область регистров хранения (Holding Registers)	41
9.3 Коды ответов Modbus шлюза PBM-ETH-3.0	43
9.4 Работа с областью данных Modbus DPV0 шлюза PBM-ETH-3.0	44
9.4.1 Операция чтения данных DPV0	44
9.4.2 Операция записи данных DPV0	46
9.5 Работа с областью данных Modbus DPV1 шлюза PBM-ETH-3.0	47
9.5.1 Назначение телеграмм запроса и ответа DPV1C1 (чтение и запись)	48
9.5.2 Примеры телеграмм запроса и ответа DPV1C1 (чтение и запись)	50

9.5.3	Назначение телеграмм запроса и ответа DPV1C2 (чтение и запись)	50
9.5.4	Примеры телеграмм запроса и ответа DPV1C2 (чтение и запись)	52
9.5.5	Асинхронные события DPV1C2	53
9.5.6	Использование DPV1C2 с помощью DTM шлюза.....	55
9.6	Работа с данными диагностики устройств PROFIBUS Slave через область интерфейса Modbus	55
9.6.1	Структура диагностических данных Ведомого устройства в Modbus 3х	55
9.6.2	Работа с областью системного журнала через Modbus-интерфейс шлюза.....	56
9.6.2.1	Формат системного журнала.....	56
9.6.2.2	Идентификаторы событий системного журнала (Log ID).....	57
9.7	Подробное описание функционального назначения областей регистров Modbus шлюза PBM-ETH-3.0	59
Глава 10. Пример организации связи с устройствами PROFIBUS PA на основе концепции FDT/DTM.....		63
10.1	Обзор функций PROFIBUS PA и DTM шлюза PBM-ETH-3.0.....	63
10.1.1	Подключение устройств PROFIBUS PA	63
10.1.2	Обмен данными с использованием DTM	63
10.1.3	Использование DTM для инициирования обмена данными DPV1	64
10.2	Описание системы примера и используемое ПО	65
10.3	Настройка устройств PROFIBUS PA	66
10.4	Применение DTM в обмене данными PROFIBUS PA.....	67
10.4.1	Коммуникационный DTM шлюза + DTM устройства PA.....	67
Приложение 1. Термины и определения		71
Приложение 2. Поиск и устранение типовых неисправностей (дополняется)		73
Приложение 3. Краткое описание спецификации обмена данными по Modbus-TCP		74
1.	Общие сведения	74
1.1	Ориентация на Соединение.....	74
1.2	Кодирование данных	75
1.3	Пояснения к справочным номерам.....	75
1.4	Основной принцип неявной длины	75
2.	Обзор классов совместимости (Conformance Classes)	76
2.1	Функции Класcа 0	76
2.2	Функции Класcа 1	76
2.3	Функции Класcа 2	76
2.4	Специальные функции устройства/производителя/сети	77
3.	Структура протокола	77
4.	Ссылочные значения протокола для классов совместимости.....	78
4.1	Подробное описание команд Класcа 0	78
4.1.1	Чтение нескольких регистров хранения (FC 03).....	78
4.1.2	Запись нескольких регистров хранения (FC 16)	79
4.2	Подробное описание команд Класcа 1	79
4.2.1	Чтение дискретных выходов (FC 01)	79
4.2.2	Чтение дискретных входов (FC 02).....	80
4.2.3	Чтение входных регистров (FC 04)	80
4.2.4	Запись одного дискретного выхода (FC 05).....	80
4.2.5	Запись одного регистра хранения (FC 06)	81
4.2.6	Чтение слова состояния исключений (FC 07)	81
4.3	Подробное описание команд Класcа 2	82
4.3.1	Запись нескольких дискретных выходов (FC 15)	82
4.3.2	Чтение общей ссылки (FC 20)	82
4.3.3	Запись общей ссылки (FC 21)	83
4.3.4	Маскированная запись регистра (FC 22)	84
4.3.5	Чтение/запись нескольких регистров (FC 23)	84
4.3.6	Чтение очереди FIFO (FC 24)	85
5.	Коды исключений	85
6.	Области памяти Modbus	87
7.	Функции Modbus.....	88
7.1	Чтение регистров хранения (Read Holding Registers)	89

7.2	Чтение входных регистров (Read Input Registers).....	90
7.3	Запись одного регистра (Write Single Register)	91
7.4	Запись нескольких регистров (Write Multiple Registers)	92

Глава 1. Обзор продукции

1. Назначение PBM-ETH-3.0

Сетевой шлюз протоколов PBM-ETH-3.0 предназначен для преобразования протоколов Modbus-TCP в PROFIBUS. PBM-ETH-3.0 выступает в роли Ведомого (Server) устройства на стороне MODBUS/TCP и в роли Главной станции (Ведущего Master-устройства) на стороне PROFIBUS, обеспечивая подключение Ведомых устройств PROFIBUS DP к сети Modbus-TCP.

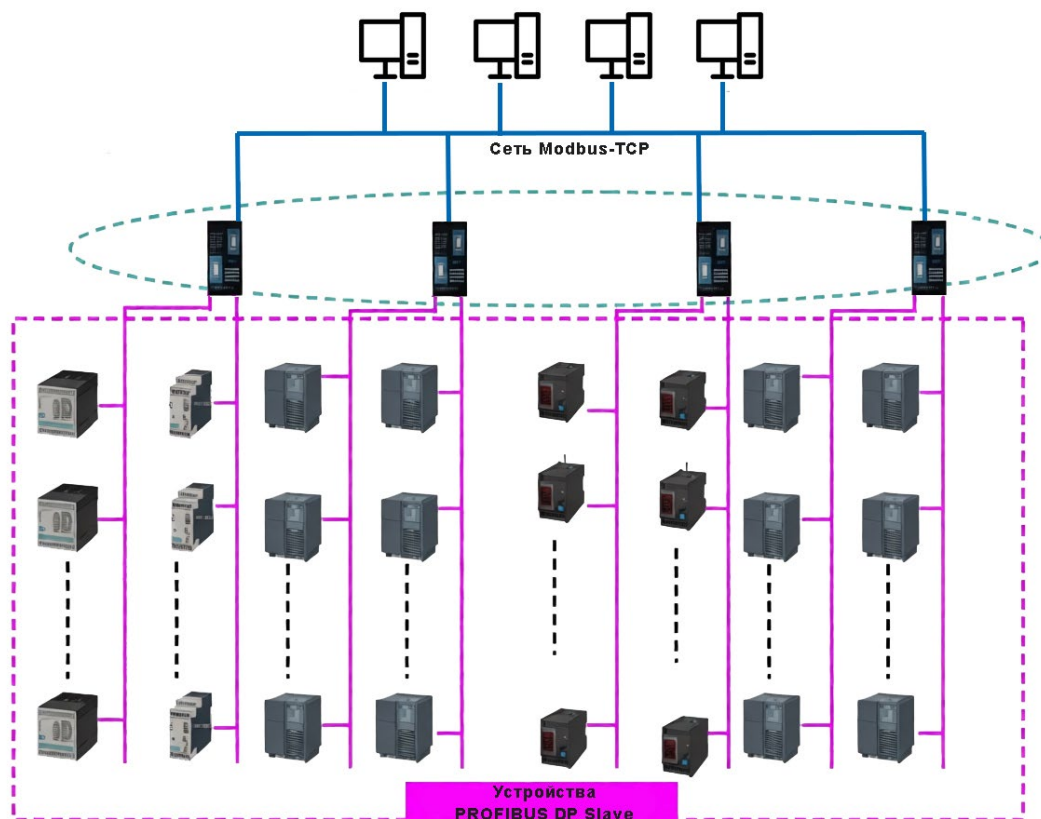


Рис. 1-1. Пример подключения.

2. Особенности PBM-ETH-3.0

- **Лёгкость в применении**
Пользователю не требуется навыков программирования и глубокие познания сетей PROFIBUS и MODBUS/TCP. Достаточно выполнить настройку в соответствии с требованиями системы, следуя указаниям настоящего руководства, что упрощает процесс интеграции.
- **Гибкость применения**
Встроенный коммутатор Ethernet и два сетевых порта с двумя режимами работы позволяют использовать устройство в различных сетевых топологиях.
- **Высокая производительность**
Используются два независимых процессора: один – как Главная станция PROFIBUS, второй – как Ведомое устройство MODBUS/TCP. Они работают параллельно, так что связь по PROFIBUS и связь по Ethernet не влияют друг на друга. Обмен данными между ними осуществляется через двухканальное ОЗУ (DPRAM).
- **Функции диагностики и статистики**
Комплексный мониторинг работы самого шлюза, Главной станции PROFIBUS и Ведомых устройств осуществляется с помощью светодиодных индикаторов на передней панели, поставляемого программного обеспечения PBConfig или через специальную область данных Modbus устройства, что позволяет быстро выявлять неисправности.
- **Поддержка обновления встроенного ПО**
Возможность обновления внутренней прошивки позволяет не только быстро устранять уязвимости и повышать стабильность изделия, но расширять и дополнять его функционал.
- **Поддержка системного журнала (логирования)**
Системный журнал регистрирует ключевые события в процессе работы устройства, что облегчает техническое обслуживание системы.

3. Технические характеристики PBM-ETH-3.0

Интерфейс PROFIBUS-DP

- Разъём: 2 разъёма DB9 типа «мама»
- Гальваническая развязка: Оба интерфейса имеют независимую гальваническую развязку; напряжение изоляции – 1 кВ
- Скорость передачи данных (бод): 9,6k; 19,2k; 45,45k; 93,75k; 187,5k; 500k; 1,5M; 3M; 6M (устанавливается в конфигурационном ПО)
- Протокол связи: PROFIBUS DPV0, DPV1C1, DPV1C2 (IEC 61158-3, GB/T 20540-2006)
- Поддержка изменения адреса Ведомого устройства в режиме онлайн
- Поддержка многомастерности: нет
- Максимальное количество Ведомых устройств: 62 (2 разъёма DB9, каждый разъём без использования повторителя – не более 31 ведомого устройства)
- Максимальный объём данных на одно Ведомое устройство: 244 байта входных и 244 байта выходных данных
- Максимальный объём конфигурационных данных: 8 кБ
- Максимальный объём диагностических данных: 4 кБ
- Максимальный объём выходных данных: 8 кБ
- Максимальный объём входных данных: 8 кБ
- Максимальное количество модулей DPV0 (в конфигурации) на одно Ведомое устройство: 32
- Количество одновременно поддерживаемых соединений DPV1C2 с Ведомыми устройствами: 1

Интерфейс Ethernet

- Разъём: 2 разъёма RJ45 с функцией коммутатора
- Напряжение изоляции: 1,0 кВ
- Характеристики сетевых портов: 10/100 Мбит/с, автосогласование, поддержка автоматического определения полярности (Auto-MDI/MDIX)
- Поддерживаемые протоколы: ARP, ICMP, IGMP, IP, TCP, UDP, MODBUS/TCP Server
- Поддержка функции ограничения скорости трафика
- Функция обнаружения конфликта IP-адресов

-
- Рекомендуемый кабель: специальный кабель для промышленного Ethernet
 - Режим работы: Modbus-TCP Server/Slave
 - Количество соединений: в режиме коммутатора – 4 соединения; в режиме двух сетевых портов – по 4 соединения на каждый сетевой адрес
 - Функция обнаружения разрыва соединения
 - Поддерживаемые функциональные команды MODBUS/TCP: 0x03, 0x04, 0x06, 0x10
 - Количество записей системного журнала: 256

Программное обеспечение конфигурации

- Требования к версии: 3.8 и выше
- Поддержка интеграции GSD-файлов Ведомых устройств
- Функция мониторинга состояния работы
- Функция мониторинга данных IO DPV0
- Возможность обновления встроенного ПО (прошивки)
- Функция логгирования (Журнал событий устройства)
- Функция создания системного отчёта
- Поддержка функций DPV1 C1/C2

Поддержка PROFIBUS-PA и функций FDT/DTM

- Поддержка DTM, возможность настройки Ведомых устройств при помощи FDT/DTM (Подробное описание использования см. в примере, Глава 8).

Электропитание

- Разъём: 6-контактная клеммная колодка
- Напряжение питания: 24 В ($\pm 20\%$)
- Рабочий ток: 152,95 мА (при 24 В, и подключении 31 Ведомого устройства)
- Номинальная потребляемая мощность: 3,6708 Вт
- Резервирование питания: поддерживается

Степень защиты

- Степень пылевлагозащиты: IP20

Условия окружающей среды

- Температура транспортировки и хранения: от -40°C до $+70^{\circ}\text{C}$
- Диапазон рабочих температур: от -25°C до $+55^{\circ}\text{C}$
- Относительная влажность: от 5% до 90% (без конденсации)

Механические характеристики

- Материал корпуса: высокопрочный пластик
- Габаритные размеры (Ш $\times$ Г $\times$ В): 55 мм $\times$ 103 мм $\times$ 120 мм

Уровень электромагнитной совместимости (ЭМС)

- Испытание на электрические быстрые переходные процессы (EFT/Burst): IEC 61000-4-4, 2 кВ (критерий А)
- Испытание на импульсные перенапряжения (Surge): IEC 61000-4-5, CM: ± 2 кВ, DM: ± 1 кВ (критерий А)

Глава 2. Внешний вид, индикация и разъёмы

2.1 Расположение элементов устройства и индикация

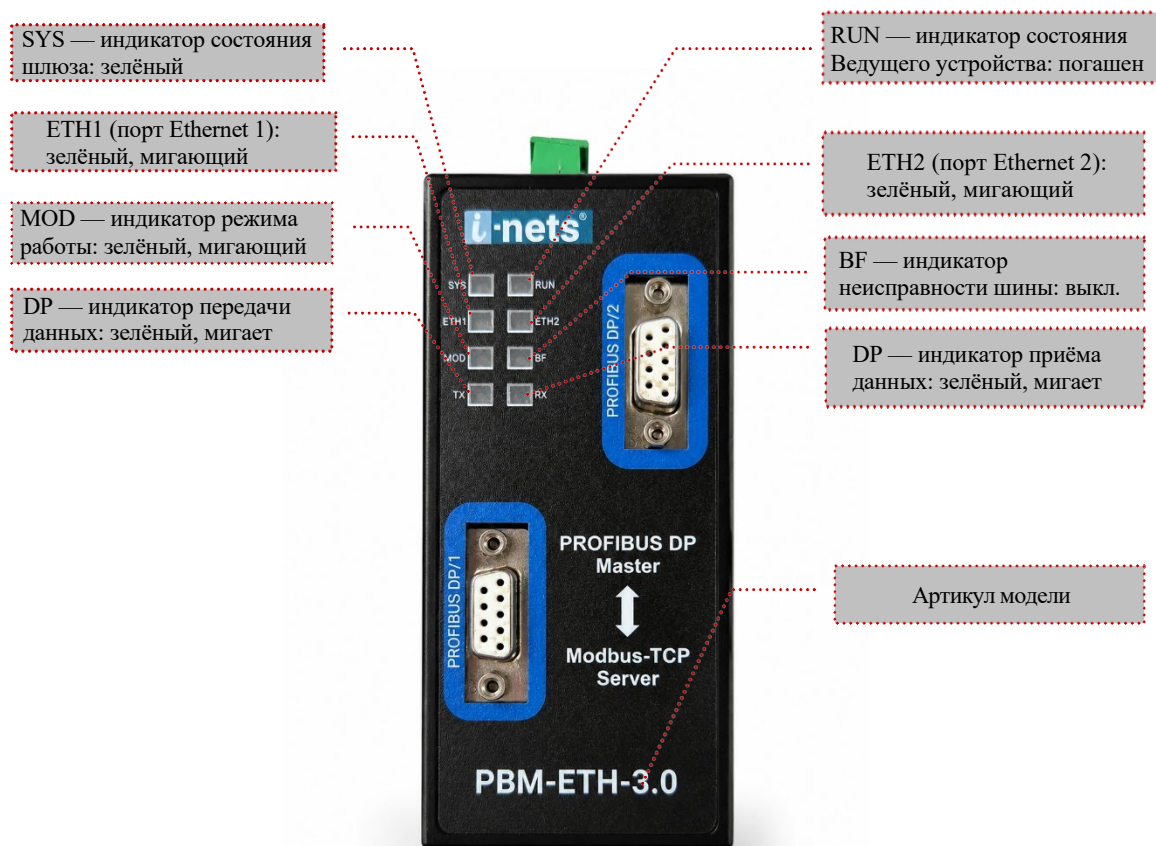


Рис. 2-1. Индикаторы шлюза PBM-ETH-3.0

Табл. 2-1. Назначение разъёмов на лицевой панели шлюза.

Наименование разъёма	Назначение
PROFIBUS-DP/1	Стандартный интерфейс PROFIBUS DP, используется со специальным разъёмом PROFIBUS.
PROFIBUS-DP/2	Стандартный интерфейс PROFIBUS DP, используется со специальным разъёмом PROFIBUS.

Таблица 2-2. Индикаторы состояния на передней панели (часть 1).

Наименование LED-индикатора	Цвет индикации	Статус	Описание
SYS — индикатор состояния системы	Двухцветный (красный/зелёный)	не горит	Питание отключено
		постоянно зелёный	Нормальный запуск в рабочем режиме
			Перед обновлением прошивки (режим обновления)
			Обновление прошивки успешно
		зелёный мигающий	Идёт обновление прошивки
			Идентификация устройства включена
		постоянно красный	Запуск устройства
			Ошибка обновления прошивки
			Ошибка загрузки конфигурации
RUN — индикатор работы Главной станции (Ведущего устройства)	Двухцветный (красный/жёлтый)	красный мигающий	Ошибка запуска рабочей прошивки (нормальный режим)
		не горит	Главная станция в состоянии RUN (выполнение)
		постоянно жёлтый	Главная станция в состоянии STOP (останов)
ETH1 — индикация порта Ethernet 1	Двухцветный (красный/зелёный)	постоянно красный	Главная станция в состоянии OFFLINE (отключена от сети)
		постоянно красный	Физическое соединение порта 1 отсутствует
		красный, мигает	Конфликт IP-адресов на порту 1 (в сети есть устройство с тем же IP)
		постоянно зелёный	Порт 1 работает нормально
ETH2 — индикация порта Ethernet 2	Двухцветный (красный/зелёный)	зелёный мигает	Идёт обмен MODBUS-пакетами по порту 1
		постоянно красный	Физическое соединение порта 2 отсутствует
		красный, мигает	Конфликт IP-адресов на порту 2 (в сети есть устройство с тем же IP)
		постоянно зелёный	Порт 2 работает нормально
		зелёный, мигает	Идёт обмен MODBUS-пакетами по порту 1

Таблица 2-2. Индикаторы состояния на передней панели (часть 2).

Наименование LED-индикатора	Цвет индикации	Статус	Описание
MOD — индикатор режима работы	Красный, зелёный	Красный, горит постоянно	Состояние IDLE главной станции
		Зелёный, мигает	Главная станция находится в рабочем режиме
BF — индикатор неисправности шины	Красный, желтый, двухцветный	Не горит / погашен	Все сконфигурированные Ведомые устройства находятся в состоянии обмена данными, аварийные сигналы отсутствуют
		постоянно жёлтый	Одно или несколько Ведомых устройств выдали аварию высокого приоритета; индикация гаснет только после сброса этой аварии
		постоянно красный	Отсутствует обмен данными с одним или несколькими сконфигурированными Ведомыми устройствами
		красный + жёлтый одновременно	Одновременно: одно или несколько Ведомых устройств потеряли связь, и одно или несколько Ведомых устройств выдали аварию высокого приоритета
TX — индикатор передачи данных DP	Красный/зелёный	Не горит / погашен	Главная станция не передаёт данные по шине DP
		Зелёный, мигает	Главная станция передаёт данные по шине DP
		Красный	Главная станция передаёт по шине DP ошибочные сообщения (пакеты)
RX — индикатор приёма данных DP	Красный/зелёный	Не горит / погашен	Главная станция не получает данные по шине DP
		Зелёный, мигает	Главная станция получает данные по шине DP
		Красный	Главная станция получает по шине DP ошибочные сообщения (пакеты)

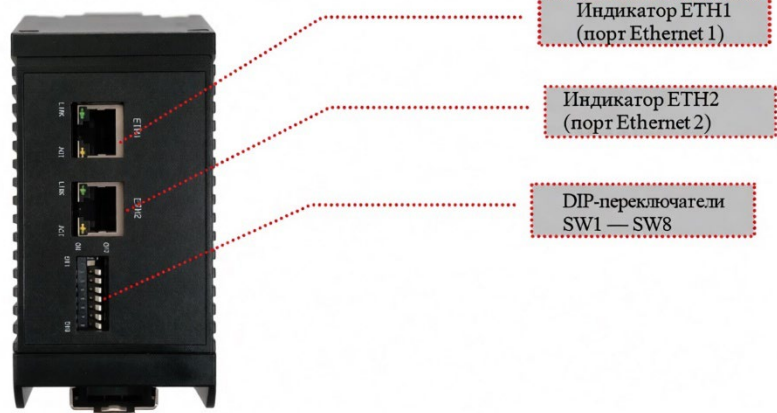


Рис. 2-2. PBM-ETH-3.0, вид снизу

Таблица 2-3. Описание сетевых разъемов RJ45 (Ethernet).

Наименование разъёма	Описание
ETH1	Интерфейс Fast Ethernet: зелёный индикатор — установлено физическое подключение к сети, оранжевый индикатор — идёт обмен данными.
ETH2	Интерфейс Fast Ethernet: зелёный индикатор — установлено физическое подключение к сети, оранжевый индикатор — идёт обмен данными.

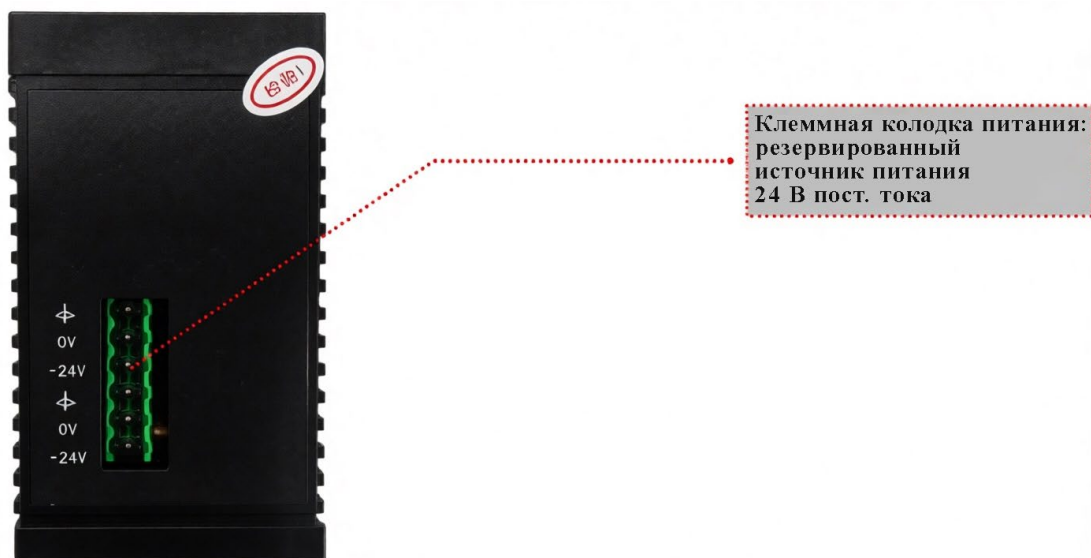


Рис. 2-2. PBM-ETH-3.0, вид сверху

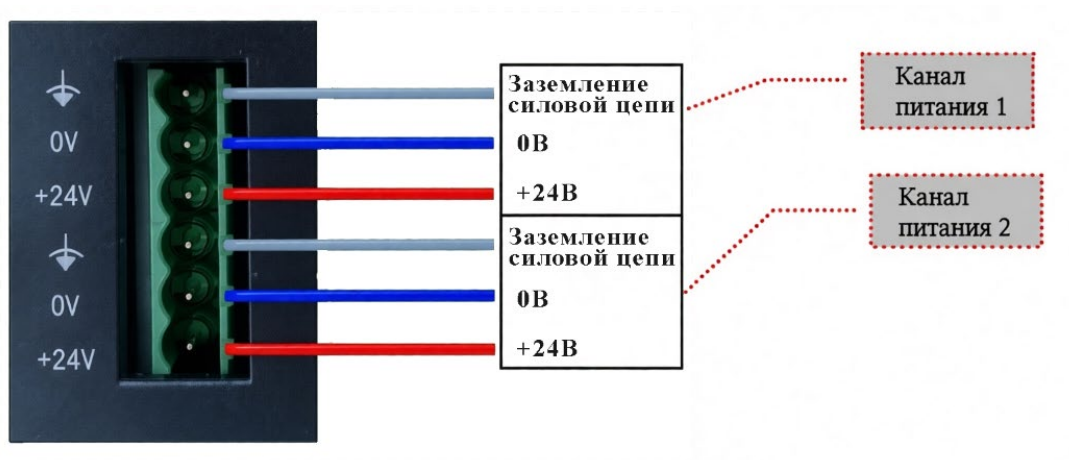


Рис. 2-4. Контакты разъёма питания PBM-ETH-3.0

Табл. 2-4. Назначение контактов разъёма питания.

Наименование контакта	Назначение
24 В пост. тока (питание 1)	Вход положительного полюса первого канала питания 24 В постоянного тока
24 В пост. тока (питание 2)	Вход положительного полюса второго канала питания 24 В постоянного тока
0В (GND)	Общий провод (0 В) для обоих каналов питания 24 В постоянного тока
Функциональное заземление (FE)	Заземление модуля (FE)

2.2 Конструкция и габариты

Габаритные размеры: 55 мм (ширина) × 103 мм (глубина) × 120 мм (высота). Схема габаритных размеров показана на Рис. 2-5.

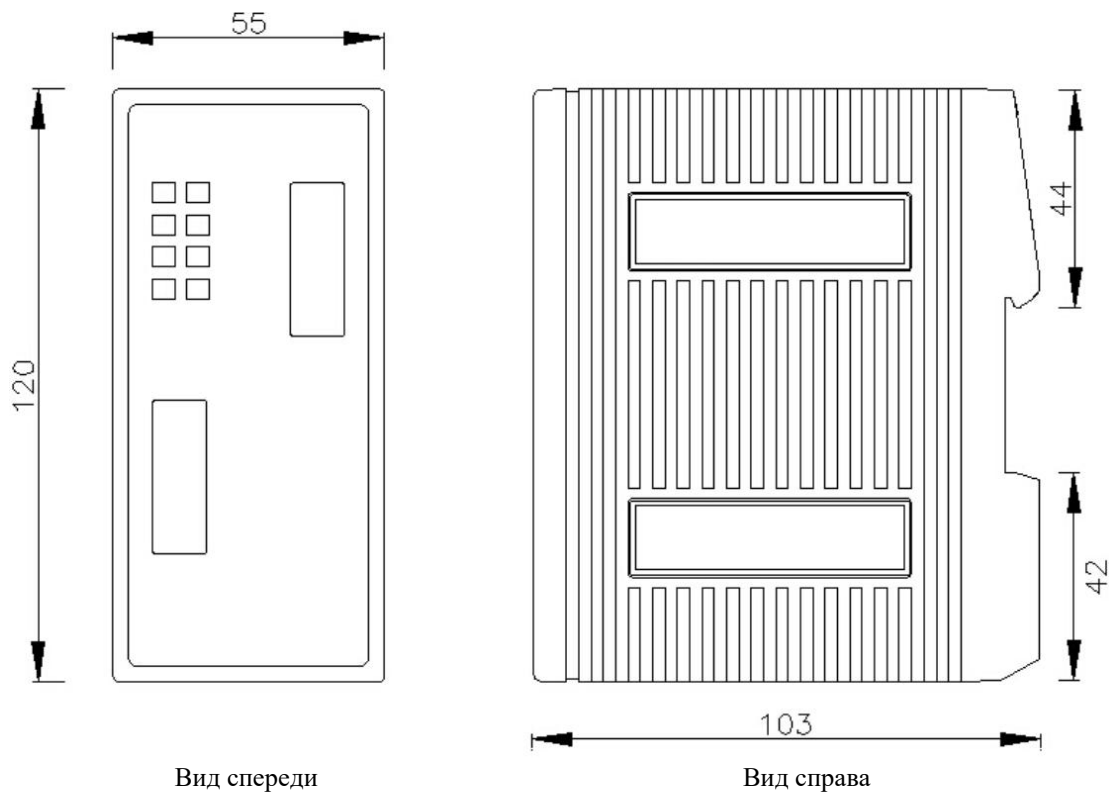


Рис. 2-5. Габаритные размеры PBM-ETH-3.0

2.3 Назначение DIP-переключателей

Функциональные DIP-переключатели изделия (SW1–SW8). В настоящее время используется только один переключатель из группы SW1–SW4. Остальные переключатели (SW5–SW8) зарезервированы. Схема расположения показана на рисунке ниже.

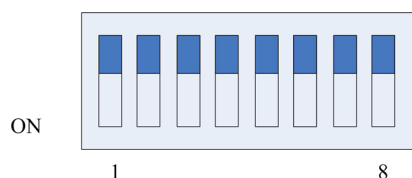


Таблица 2-5. Настройка DIP-переключателей.

Номер переключателя	Функция	Описание
Bit1	Сетевые параметры по умолчанию	ON — запуск с сетевыми параметрами по умолчанию; OFF — запуск с пользовательскими сетевыми параметрами.

Bit2	Зарезервирован	-
Bit3	Режим обновления прошивки	ON — режим обновления встроенного ПО; OFF — нормальный рабочий режим.
Bit4	Автономный режим (OFFLINE)	ON — главная станция в режиме OFFLINE, локальная конфигурация не используется, сообщения в шину DP не передаются. При невозможности запуска шлюза из-за ошибки конфигурации можно переключиться в этот режим, загрузить новую конфигурацию при помощи конфигурационного ПО и, затем, вернуться в нормальный режим работы.
Bit5~8	Зарезервирован	-

Примечание: шлюз считывает состояние DIP-переключателей только в момент включения питания, после чего переходит в нормальный режим работы. Изменение положения DIP-переключателей вступит в силу только после перезагрузки шлюза.

Сетевые параметры по умолчанию

При отсутствии заданных пользователем сетевых параметров или при установке DIP-переключателя в положение «Сетевые параметры по умолчанию» шлюз запускается со следующими параметрами сети:

Таблица 2-6. Сетевые параметры Ethernet-интерфейса по умолчанию.

Параметр	Значение
Режим работы Ethernet-интерфейса	Режим двух портов
IP-адрес Разъёма 1	192.168.1.10
Маска подсети Разъёма 1	255.255.255.0
IP-адрес основного шлюза Разъёма 1	192.168.1.1
IP-адрес Разъёма 2	192.168.2.10
Маска подсети Разъёма 2	255.255.255.0
IP-адрес основного шлюза Разъёма 2	192.168.2.1

2.4 Режимы работы портов Ethernet

РВМ-ЕТН-3.0 имеет два разъёма Ethernet, которые могут работать в режиме коммутатора (switch) или в режиме двух независимых портов. При настройке шлюза РВМ-ЕТН-3.0 с помощью поставляемого конфигурационного ПО РВ-Conf1 пользователь может выбрать один из этих режимов работы.

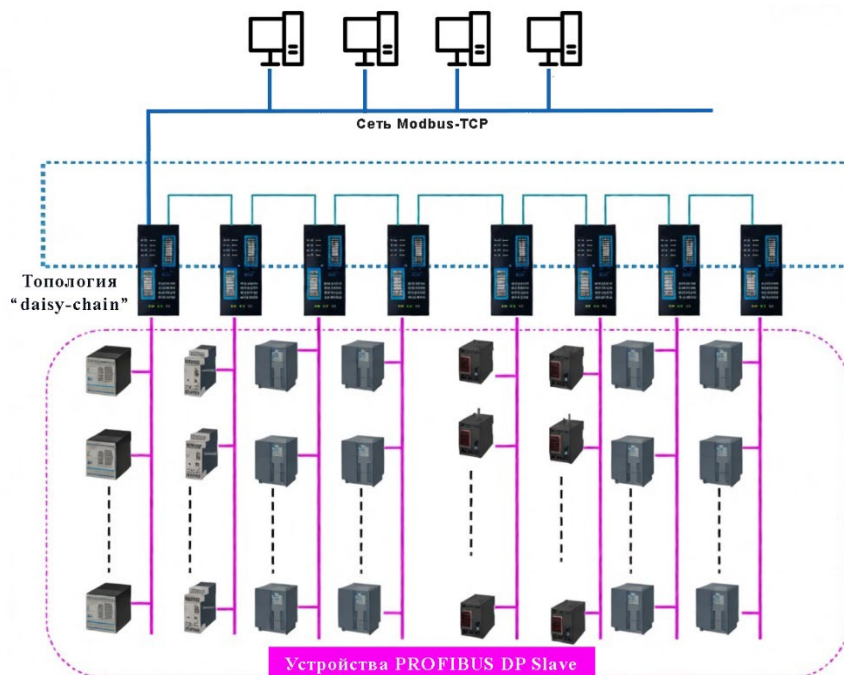


Рис. 2-6. Подключение к сети в режиме коммутатора (топология «daisy chain», использование внешнего коммутатора не требуется)

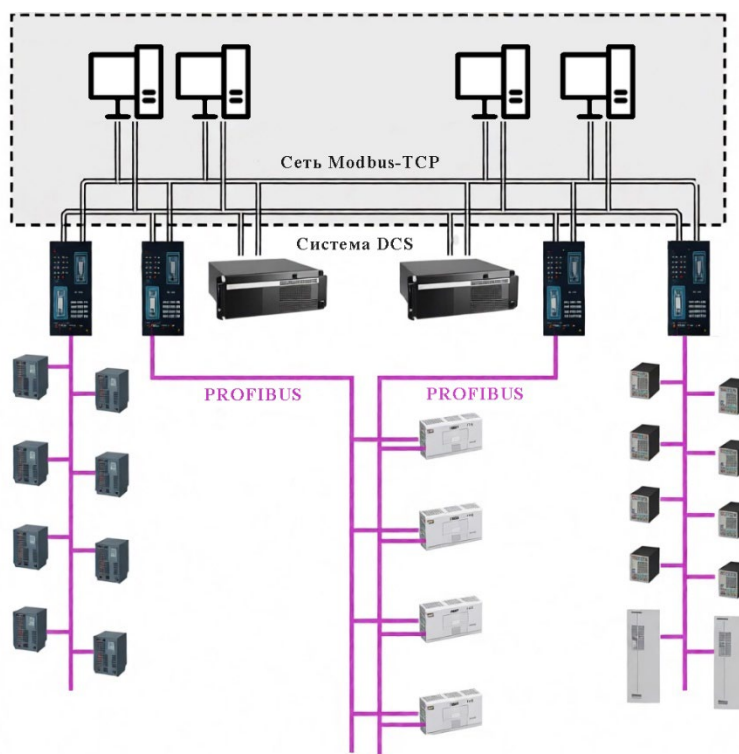


Рис. 2-7. Подключение РВМ-ЕТН-3.0 в режиме двух сетевых портов

Глава 3. Установка и подключение

3.1 Установка на DIN-рейку

При установке преобразователя PBM-ETH-3.0 следует применять стандартную DIN-рейку шириной 35 мм. Рейка устанавливается горизонтально. Над и под шлюзом должно быть обеспечено свободное пространство не менее 40 мм для надлежащего отвода тепла.



Рис. 3-1 а. Установка

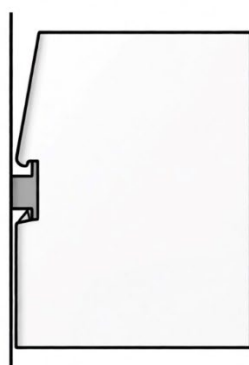


Рис. 3-1 б. Фиксация

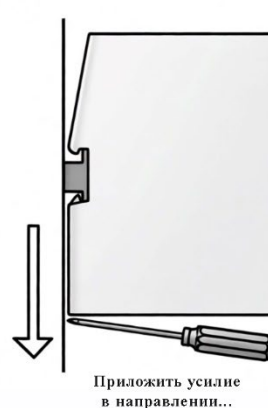
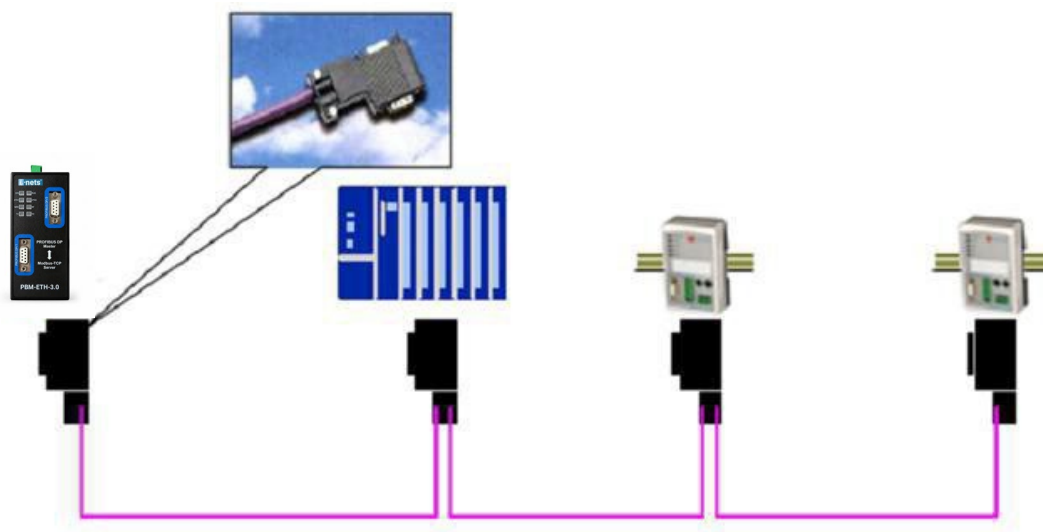


Рис. 3-1 с. Демонтаж

3.2 Разъём интерфейса PROFIBUS и его установка

Разъём интерфейса PROFIBUS шлюза PBM-ETH-3.0 выполнен в виде стандартного 9-контактного D-образного разъёма (розетки). Рекомендуется использовать стандартный штекер PROFIBUS и стандартный кабель PROFIBUS, а также устанавливать оконечные резисторы (терминаторы) на обоих концах шины. Требования к монтажу PROFIBUS приведены в соответствующих технических стандартах PROFIBUS. См. рис. 3-2 ниже:



3.3 Подключение питания

Отдельное подключение к источнику питания 24 В постоянного тока ($\pm 20\%$).

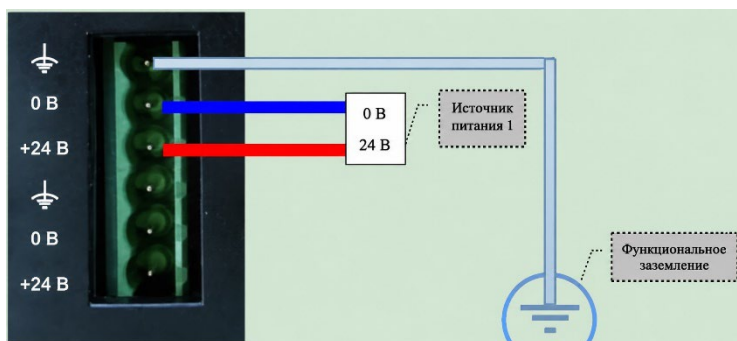


Рис. 3-3а. Подключение одного (отдельного) источника питания

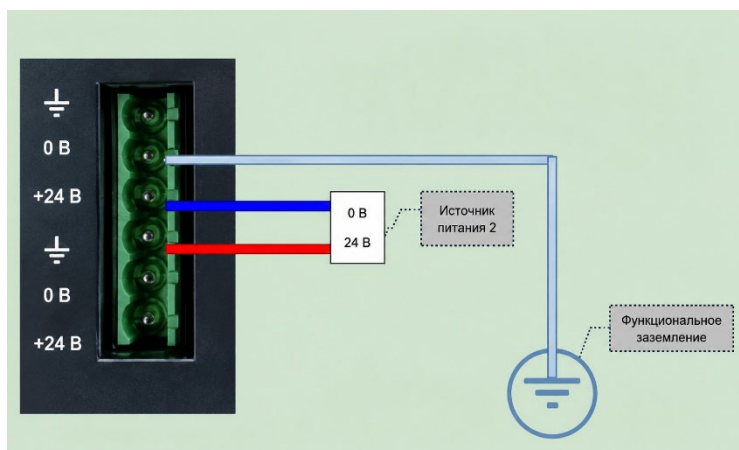


Рис. 3-3б. Подключение второго источника питания

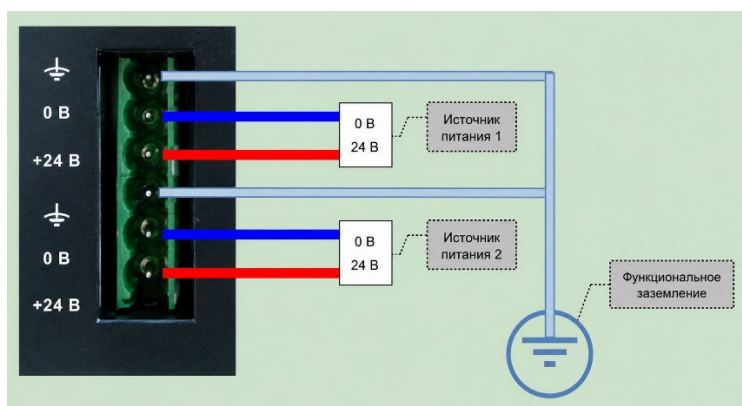


Рис. 3-4. Стандартная схема подключения питания в режиме резервирования

3.4 Подключение кабелей интерфейсов Ethernet и PROFIBUS

Порт Ethernet: разъём RJ45, с автоматическим определением прямого/кроссового кабеля. Для повышения надёжности связи и помехозащищённости рекомендуется использовать экранированный четырёхжильный кабель Industrial Ethernet, что существенно повышает стабильность системы связи.

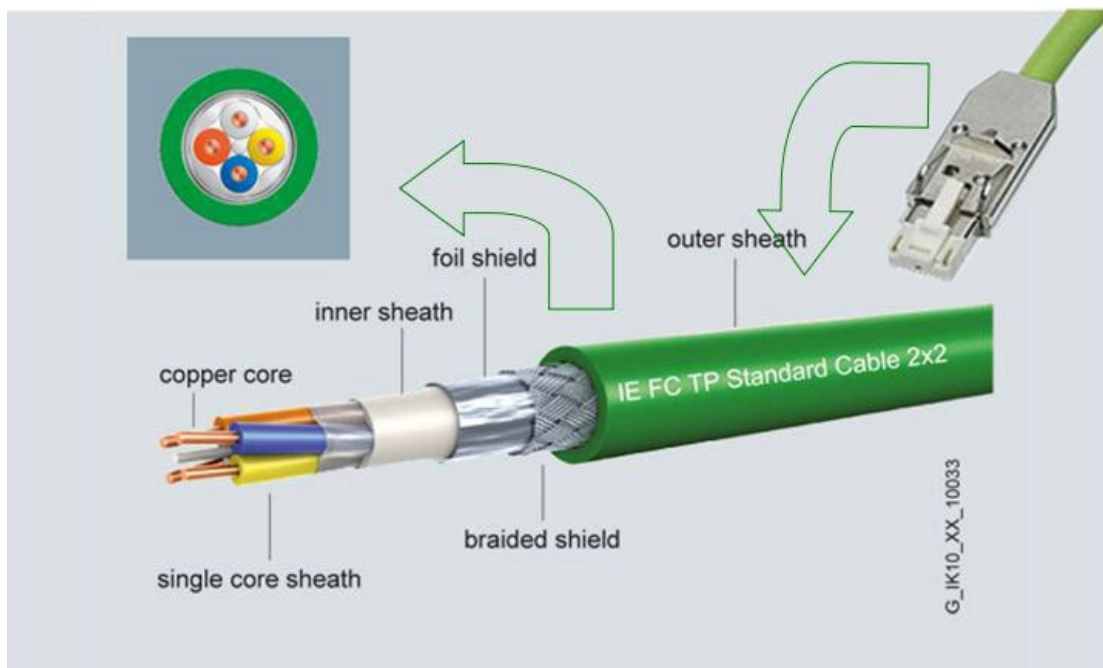


Рис. 3-5. Кабель Industrial Ethernet

Дополнительные сведения по монтажу кабеля PROFIBUS-DP приведены в «Руководстве по установке полевой шины PROFIBUS».

Глава 4. Принцип работы PBM-ETH-3.0

PBM-ETH-3.0 выступает в роли MODBUS/TCP Server (Slave) устройства. Посредством области данных MODBUS пользователю доступны функции управления, диагностики и обмена данными по протоколу DP с ведомыми устройствами PROFIBUS-DP. Поддерживаются следующие функциональные коды MODBUS: 0x03, 0x04, 0x06 и 0x10.

Шлюз PBM-ETH-3.0 позволяет реализовать мониторинг и управление, а также совместное использование данных между MODBUS и PROFIBUS посредством отображения (мэппинга) циклических/ациклических данных PROFIBUS и внутренних данных шлюза в область памяти MODBUS. Пользовательское устройство MODBUS/TCP Client (Master) может выполнять чтение и запись циклических/ациклических данных PROFIBUS путём операций чтения/записи в 3XXXX и 4XXXX областях памяти MODBUS.

Ниже показана схема преобразования и передачи данных внутри шлюза:

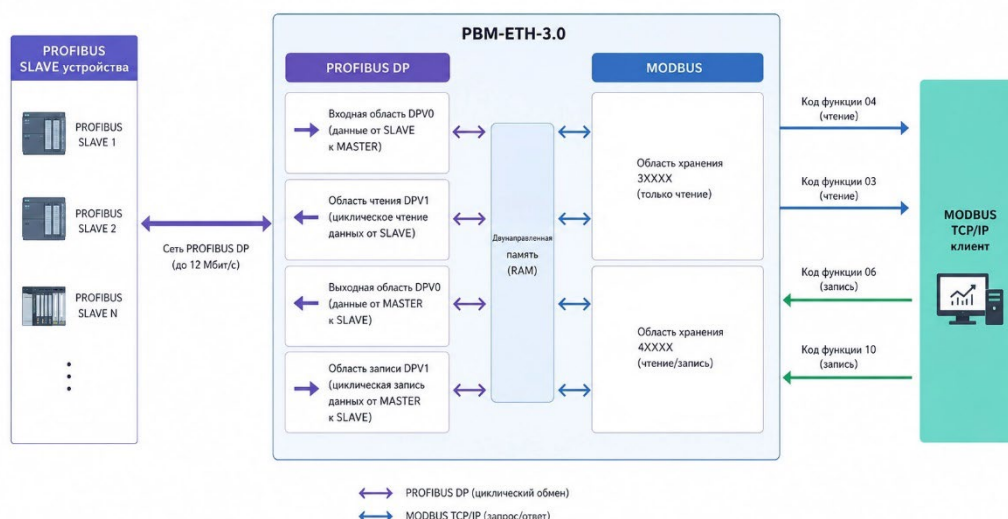


Рис. 4.1 Схема мэппинга данных внутри шлюза

Входные и выходные данные Ведомых устройств PROFIBUS DP размещаются в DPRAM (двухпортовой оперативной памяти). Центральный Процессор (далее ЦП), на котором выполняется главный стек протокола PROFIBUS, отвечает за обмен данными Ввода-Вывода с Ведомыми устройствами PROFIBUS DP. Он считывает данные из Выходной области памяти DPRAM и отправляет их Ведомым устройствам PROFIBUS DP, а входные данные, полученные от Ведомых устройств PROFIBUS DP, записывает во входную область памяти DPRAM.

Со стороны Ethernet Modbus TCP ЦП отвечает за обработку запросов Ведущего устройства Modbus TCP. При поступлении запроса на запись данные Modbus-TCP записываются в регистр, соответствующий Выходной области памяти DPRAM, согласно настроенному соотношению данных Ведомого устройства на стороне PROFIBUS DP. При запросе на чтение данные Modbus TCP извлекаются из соответствующего регистра Входной области данных DPRAM согласно настроенному соотношению данных Ведомого устройства PROFIBUS DP и возвращаются Ведущему устройству Modbus TCP (Modbus TCP Client).

Глава 5. Начало работы с PBM-ETH-3.0

Для соединения шлюза PBM-ETH-3.0 с Ведомыми устройствами PROFIBUS DP следует использовать стандартный кабель PROFIBUS. Для подключения шлюза PBM-ETH-3.0 к Ведущему устройству Modbus-TCP следует использовать стандартный кабель Industrial Ethernet (приме которого показан в п 3.4 на Рис. 3-5).

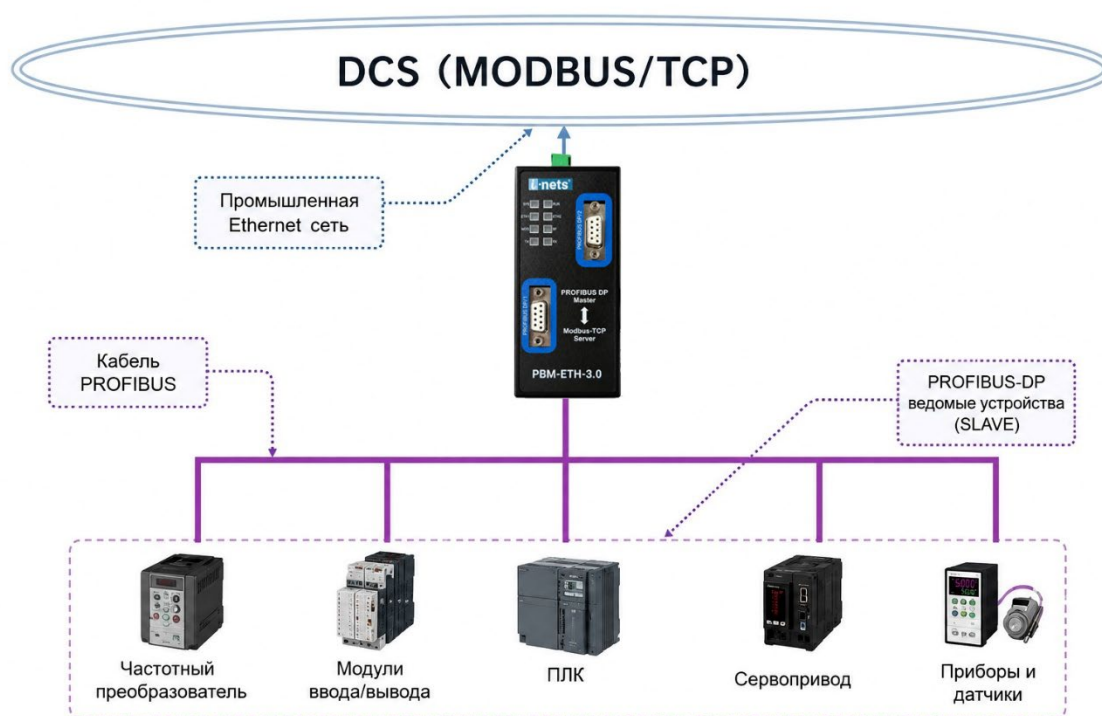


Рис. 5-1. Пример подключения

Перед началом работы необходимо настроить параметры PROFIBUS и Modbus-TCP шлюза PBM-ETH-3.0 при помощи ПО PBConfi. Соотношение отображения (мэппинг) данных между интерфейсами приведены в Главе 4.

В следующей главе описывается, как загрузить конфигурацию в шлюз с помощью ПО PBConfi.

Глава 6. Настройка с помощью конфигурационного ПО PVBConf

В данной главе в основном описываются функции программного обеспечения для конфигурирования PVBConf и методы конфигурирования PBM-ETH-3.0.

Функции конфигурационного ПО PVBConf:

- ❖ Конфигурирование и загрузка — настройка соответствующих параметров PBM-ETH-3.0 и загрузка конфигурации в шлюз по интерфейсу Ethernet.
- ❖ Онлайн-мониторинг — получение данных ввода-вывода Ведомых устройств PROFIBUS DP, выполнение коммуникации DPV1/C1/C2, получение диагностической информации о состоянии системы (Ведущего устройства и подключённых к нему Ведомых устройств), регистрация ключевых событий работы системы с помощью встроенной функции системного журнала (лога).
- ❖ Обновление встроенного ПО — быстрое обновление функциональных возможностей и исправление дефектов с помощью встроенной функции обновления микропрограммного обеспечения устройства (обновление прошивки шлюза).

Принцип настройки PBM-ETH-3.0:

При загрузке конфигурации рекомендуется подключить DP-порт шлюза PBM-ETH-3.0 к Ведомым устройствам PROFIBUS DP (для удобства последующей онлайн-отладки с помощью PVBConf). Сетевой порт Ethernet подключите к ПК с уже установленным конфигурационным ПО PVBConf. Ведущее устройство Modbus-TCP подключать пока не требуется — подключить его можно будет после завершения загрузки конфигурации или для отладки. Схема подключения показана ниже:

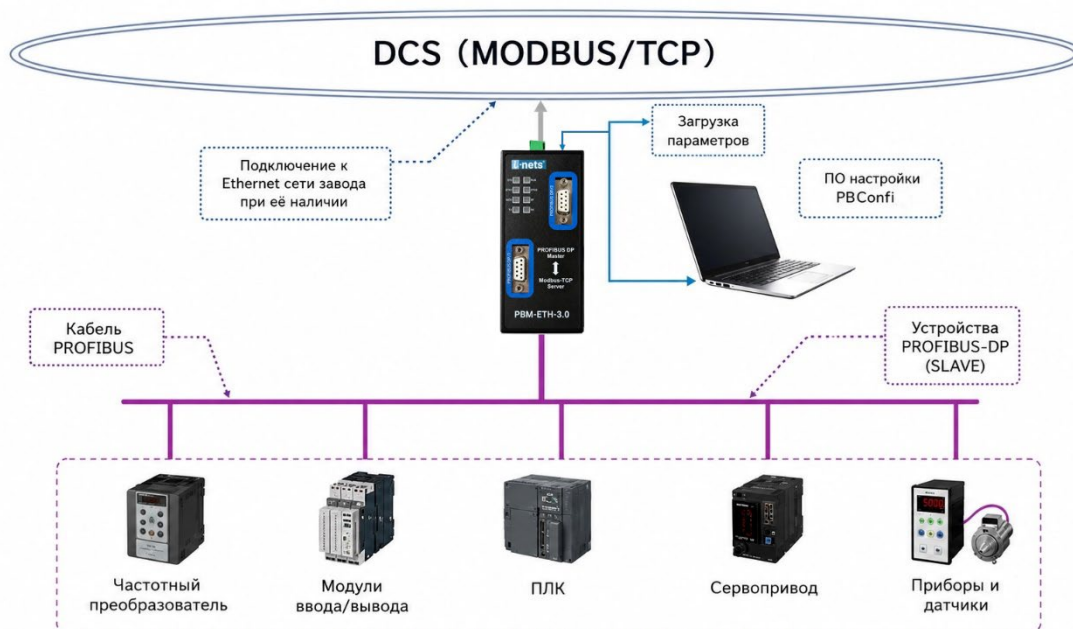


Рис. 6-1. Схема подключения для загрузки конфигурации

6.1 Настройка с помощью PBConfi в режиме оффлайн

Ниже приведён пример настройки для одного из вариантов применения. Порядок настройки следующий:

Таблица 6-1. Таблица устройств и ПО, используемых в примере.

№	Наименование устройства	Модель	Кол-во	Примечание
1	Шлюз	PBM-ETH-3.0	1	Данное устройство
2	Ведомое устройство PROFIBUS	PB-DSDPV1	1	Допускается любое другое Ведомое устройство
3	Клиент MODBUS/TCP	ПК	1	Симуляция устройства Modbus-TCP Client
		ПО ModScan32.exe	1	
4	Кабель PROFIBUS DP (с разъёмом)	Стандартный кабель PROFIBUS	1	Для подключения со стороны PROFIBUS
5	Сетевой кабель Ethernet (с разъёмом RJ45)	Обычный сетевой кабель Ethernet	1	Для подключения со стороны Ethernet

6.1.1 Создание нового проекта

Чтобы открыть главное окно PBConfi дважды кликните по ярлыку ПО PBConfi. После открытия окна конфигурации программы отобразится следующий интерфейс:

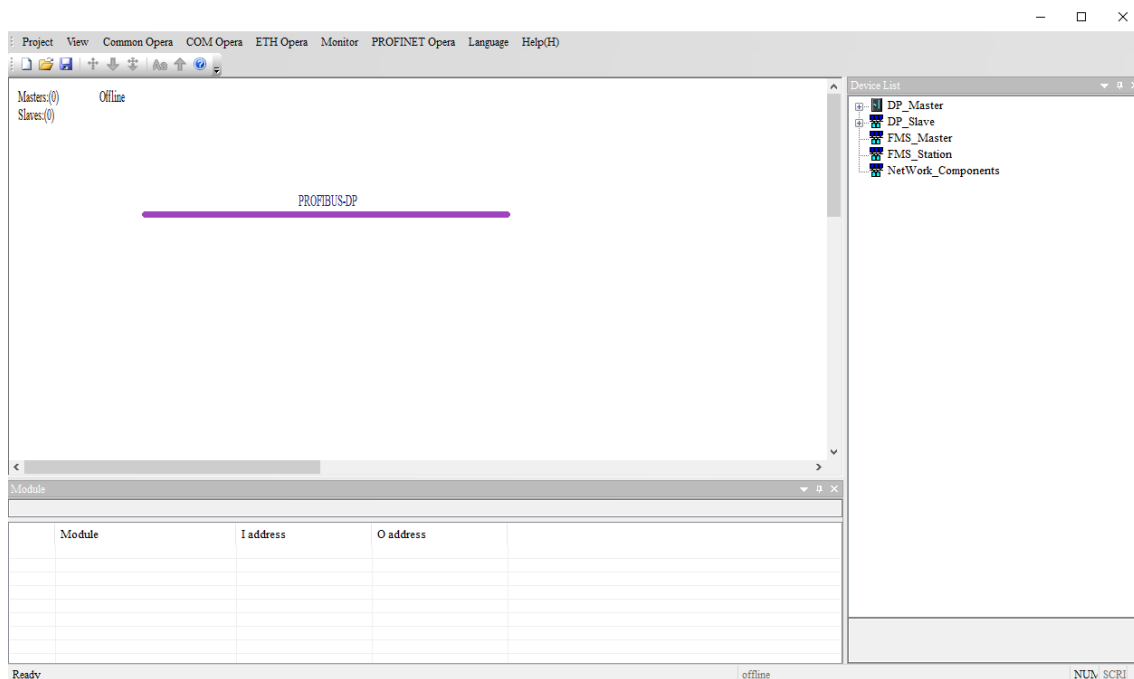


Рис. 6-2. Окно нового проекта ПО PBConfi

6.1.2 Обновление каталога устройств

Если файл GSD Ведомого устройства, которое требуется сконфигурировать пользователю, ещё не помещён в соответствующий каталог программы PBConf, то нужно перейти по пути «View» → «Work Path» → «GSD Path» и скопировать нужный GSD-файл Ведомого устройства в открывшуюся папку «GSD» каталога ПО PBConf.

После добавления GSD-файла Ведомого устройства необходимо обновить текущий каталог устройств. Кликните «Project» → «ReloadGSD», как показано на Рис. 6-3, после чего каталог устройств в левой части окна программы обновится. При этом соответствующее Ведомое устройство должно появиться в каталоге устройств в папке «DP_Slave».

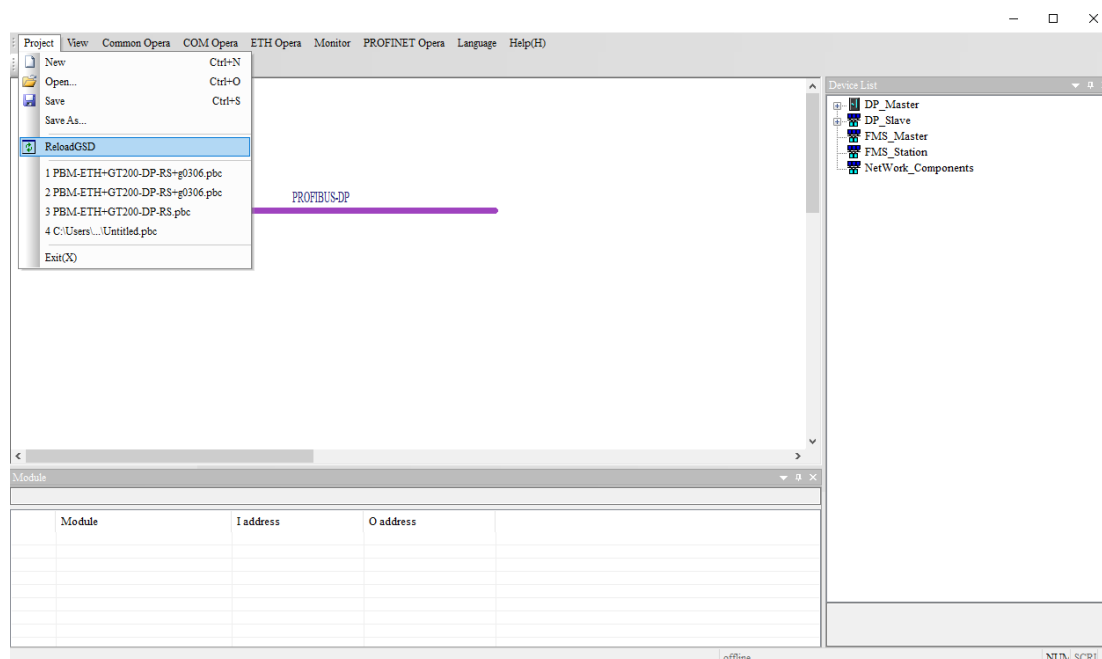


Рис. 6-3. Обновление каталога устройств (GSD)

6.1.3 Добавление Ведущего устройства (Master)

На панели аппаратных устройств в левой части интерфейса ПО выберите «DP_Master» → «PBM-ETH-3.0» и дважды кликните по нему левой кнопкой мыши. ПО автоматически добавит Ведущее устройство PBM-ETH-3.0 в конфигурацию сети.

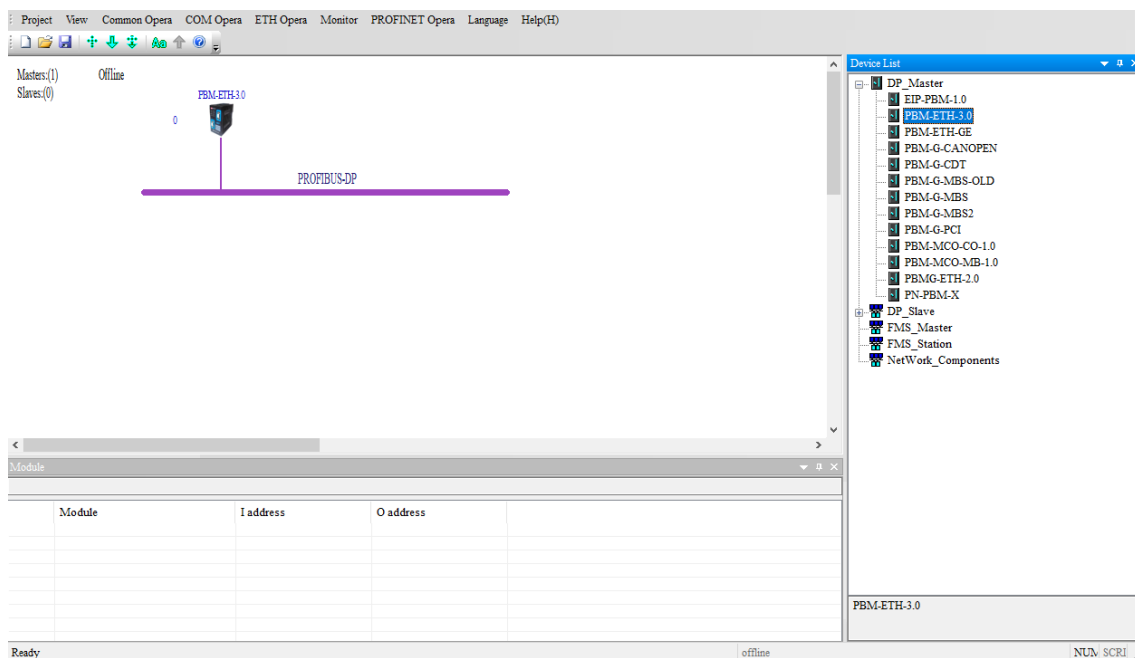


Рис. 6-4. Добавление ведущего устройства

Дважды кликните по значку Ведущего устройства в области отображения структуры сети — откроется окно определения свойств Ведущего устройства PROFIBUS, в котором можно задать сетевой адрес Ведущего устройства, скорость передачи данных по шине и другие соответствующие параметры.

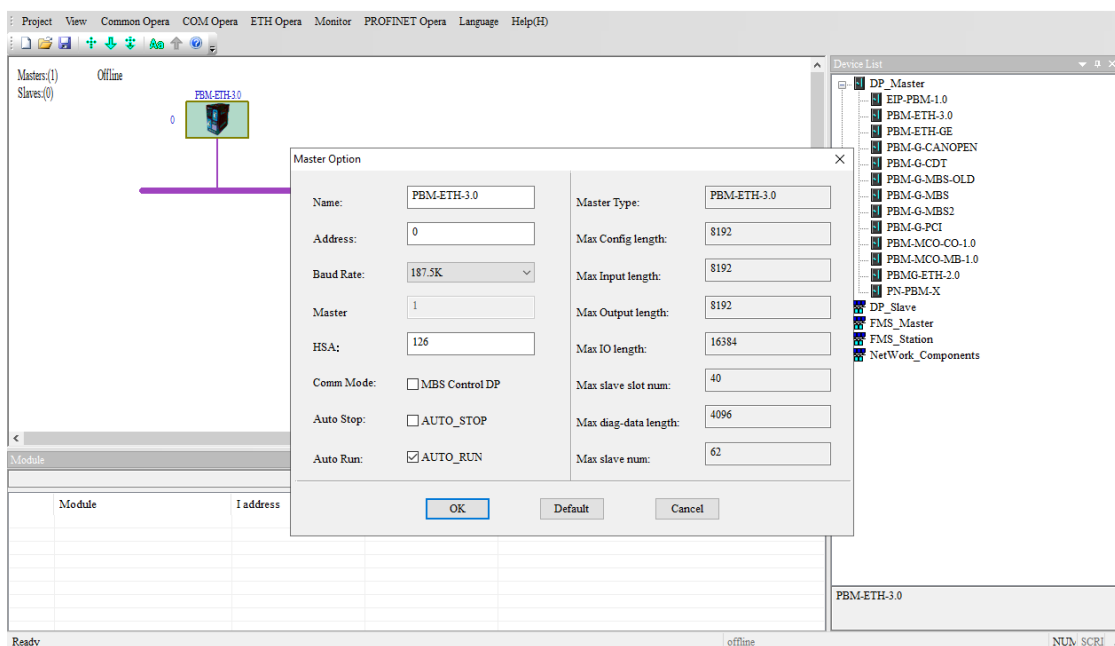


Рис. 6-5. Окно настройки параметров Ведущего устройства PROFIBUS (Master)

Таблица 6-2 — Режимы работы Ведущего устройства (Master).

Наименование режима работы	Описание
Автоматический останов AUTO STOP	Если функция включена, то при выходе любого сконфигурированного Ведомого устройства DP из состояния обмена данными Ведущее устройство автоматически переходит в состояние STOP. При этом, независимо от того, включена ли функция AUTO RUN, в режиме STOP попытка пользователя вручную переключить Ведущее устройство в состояние RUN будет отклонена. Если после этого все Ведомые устройства вновь вернуться в состояние обмена данными, то: • при отключённой функции AUTO RUN Ведущее устройство останется в состоянии STOP, и для возврата в состояние RUN потребуются ручное переключение пользователем; • при включённой функции AUTO RUN Ведущее устройство автоматически вернётся в состояние RUN.
Автоматический запуск AUTO RUN	Если функция включена, Ведущее устройство автоматически переходит в состояние RUN при подаче питания. Если функция запрещена, Ведущее устройство при подаче питания переходит в состояние STOP, и для перехода в состояние RUN требуется ручное переключение пользователем. Совместная работа AUTO STOP и AUTO RUN: при одновременном включении обеих функций приоритет имеет AUTO STOP. Если сконфигурированное Ведомое устройство выходит из состояния обмена данными, Ведущее устройство переходит в состояние STOP; когда все Ведомые устройства возвращаются в состояние обмена данными, Ведущее устройство автоматически возвращается в состояние RUN. <i>Примечание по ручному переключению:</i> если при включённой функции AUTO RUN Ведущее устройство работает в состоянии RUN, и пользователь вручную переключает его в состояние STOP, Ведущее устройство останется в состоянии STOP в течение 50 мс, после чего автоматически вернётся в состояние RUN.

6.1.4 Добавление Ведомого устройства (Slave)

Кликните по панели аппаратных устройств в левой части интерфейса программы, затем выберите «DP_Slave» → «Gateways». Дважды кликните левой кнопкой мыши по Ведомому устройству PB-DSDPV1 в раскрывающемся меню, чтобы добавить его в конфигурацию сети.

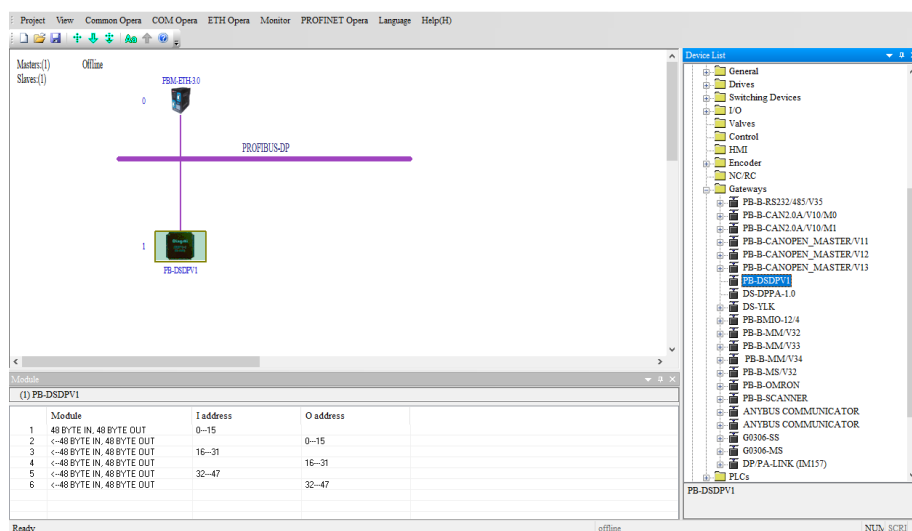


Рис. 6-6. Добавление Ведомого устройства PROFIBUS

Дважды кликните левой кнопкой мыши по значку Ведомого устройства в окне отображения структуры сети — в открывшемся окне можно настроить адрес Ведомого устройства, пользовательские параметры, включить или отключить сторожевой таймер WD (Watchdog) или задать другие связанные параметры.

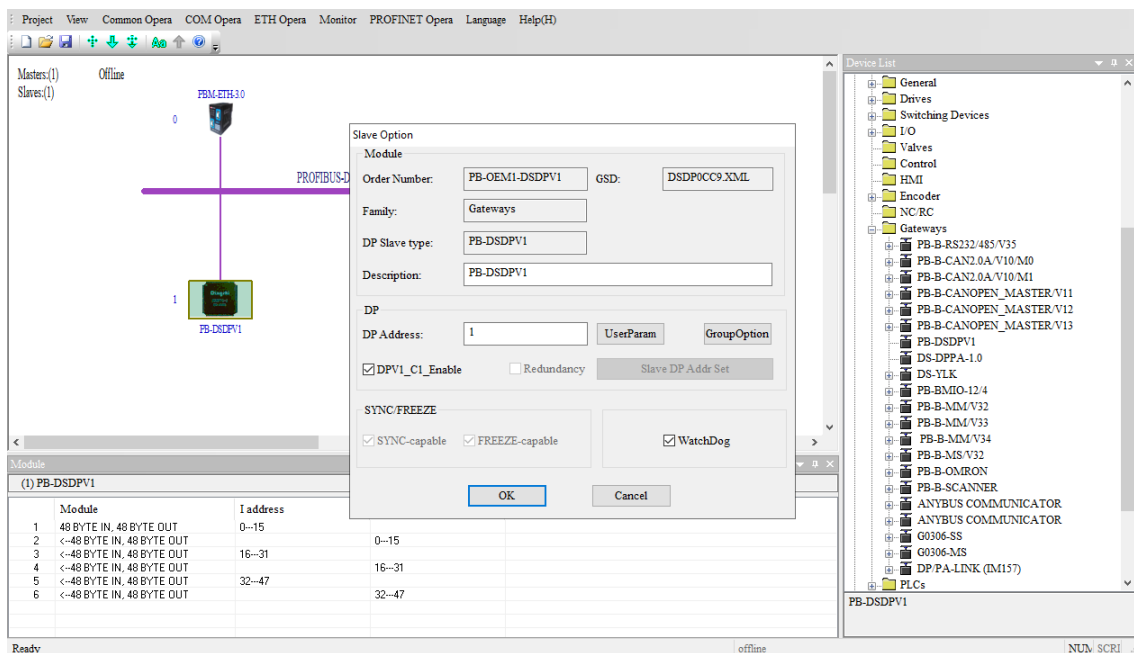


Рис. 6-7. Окно настройки параметров Ведомого устройства PROFIBUS (Slave)

6.1.4.1 Настройка параметров PROFIBUS DPV1

Для Ведомых устройств, поддерживающих функцию DPV1 (в файле GSD которых содержится описание DPV1_Slave = 1), в окне свойств Ведомого устройства опция DPV1_Enable выбрана по умолчанию. Если не планируется использовать функции DPV1C1/C2 данного Ведомого устройства, можно снять флажок DPV1_Enable (для некоторых Ведомых устройств DPV1 такая настройка может привести к нарушению связи по DPV0). Для Ведомых устройств, не поддерживающих функцию DPV1, опция DPV1_Enable по умолчанию не выбрана и недоступна для изменения.

6.1.4.2 Настройка сторожевого таймера (Watchdog)

Ведомое устройство может использовать сторожевой таймер (Watchdog) для мониторинга обмена данными по шине, чтобы подтверждать, что Ведущее устройство находится в рабочем состоянии, а значения технологических данных продолжают обновляться. Ведомое устройство получает значение времени сторожевого таймера посредством параметрирующей Телеграммы. Если из-за высокой загрузки шины Ведомое устройство не может перезапустить сторожевой таймер, оно переходит в состояние Wait_Prm и устанавливает выходы в безопасное состояние. Безопасное состояние зависит от конкретного применения и не может быть задано заранее.

6.1.4.3 Настройка пользовательских параметров Ведомого устройства

Пользовательские параметры Ведомых устройств в большинстве случаев различаются. У некоторых Ведомых устройств пользовательские параметры не подлежат изменению, в то время как другие Ведомые устройства предоставляют пользовательские параметры, доступные для настройки.

6.1.4.4 Настройка Sync/Freeze и группирования

В режиме синхронизации (Sync) выходные данные адресуемого Ведомого устройства фиксируются в текущем состоянии. В последующих циклах передачи пользовательских данных Ведомое устройство сохраняет принимаемые выходные данные, однако состояние его выходов остаётся неизменным. Сохранённые выходные данные передаются на периферийные устройства только при поступлении следующей команды синхронизации. Выход из режима синхронизации осуществляется пользователем с помощью команды отмены синхронизации (Unsync).

В режиме фиксации (Freeze) входные данные Ведомого устройства фиксируются в текущем состоянии и не обновляются до тех пор, пока Ведущее устройство не отправит следующую команду фиксации. Выход из режима фиксации осуществляется пользователем с помощью команды отмены фиксации (Unfreeze).

Операции синхронизации и фиксации реализуются путём отправки стандартных широковещательных DP-Телеграмм, адресованных одной или нескольким группам. Стандартом определено 8 групп, каждая из которых представлена одним битом в байте широковещательной Телеграммы. Для использования функции Sync/Freeze на этапе конфигурирования системы PROFIBUS необходимо определить назначение каждой из 8 групп, после чего отдельные Ведомые устройства могут быть опционально включены в соответствующие группы.

PBM-ETH-3.0 поддерживает функции Sync/Freeze в соответствии со стандартом PROFIBUS. Порядок настройки приведён ниже.

Сначала в окне «Group Option» на вкладке «Group Properties» (Рис. 6-8) нужно определить функцию каждой группы. Любая группа может быть настроена в одном из следующих режимов: группа Sync, группа Freeze, группа Sync/Freeze или запрещено.

Group	SYNC	FREEZE	Comment
Group 8:	☒	☒	
Group 7:	☒	☒	
Group 6:	☒	☒	
Group 5:	☒	☒	
Group 4:	☒	☒	
Group 3:	☒	☒	
Group 2:	☒	☒	
Group 1:	☒	☒	

Рис. 6-8. Окно «Group Option» (информации о группах)

Затем на вкладке «Group Assignment» (Рис. 6-9) нужно добавить Ведомые устройства в группы. Доступные варианты добавления: добавление в группу «SYNC», добавление в группу «FREEZE», добавление в группу «SYNC/FREEZE», не добавлять ни в одну группу. Ведомое устройство может быть добавлено только в те группы, которые были разрешены (включены) на вкладке «Group Properties».

Group NO.	8	7	6	5	4	3	2	1
SYNC	V	V	V	V	V	V	V	V
FREEZE	V	V	V	V	V	V	V	V
(1) PB-DSDPV1								

Рис. 6-9. Окно «Group Assignment» (распределения по группам)

6.1.5 Настройка параметров шины PROFIBUS

Дважды кликните по линии шины в области отображения структуры сети — откроется окно «Bus Param», показанное на Рис. 6-10 ниже, в котором отображаются такие параметры, как время слота (Time Slot), Gap, Retry, а также вычисляемые на их основе Tid, Ttr и время сторожевого таймера WD.

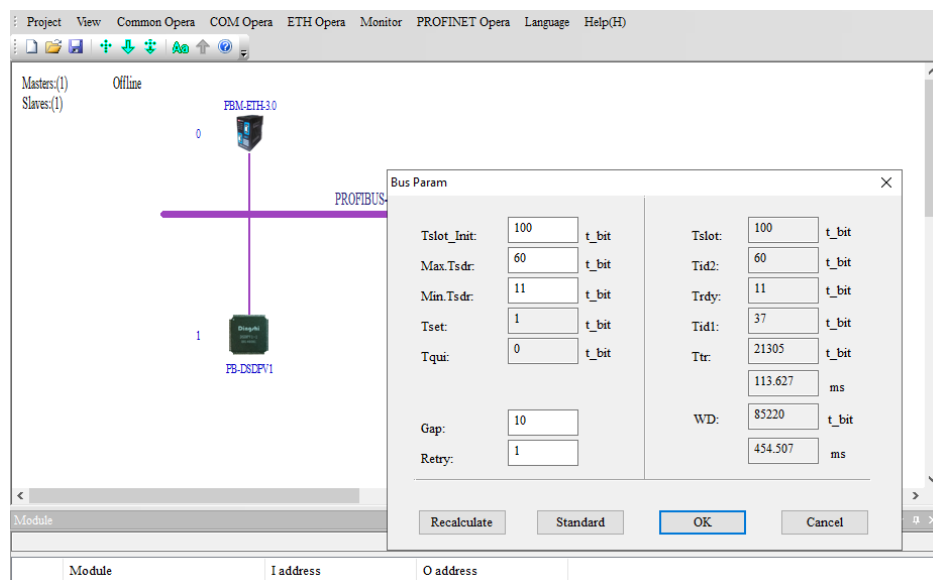


Рис. 6-10. Окно «Bus Param» (параметров шины PROFIBUS)

- **Tslot_Init** — определяет время ожидания (тайм-аут) ответа от Вещного устройства после отправки Ведущим устройством Телеграммы (в битовых интервалах, tbit).
- **Max.Tsdr** — используется для расчёта времени сторожевого таймера (Watchdog) и Ttr.
- **Gap** — определяет, через какое количество передач Токена (Token) отправляется одна Телеграмма FDL (Fieldbus Data Link).
- **Retry** — количество повторных попыток передачи Телеграммы.

6.1.6 Настройка IP-адреса для подключения ПО PBConfi

Для настройки IP-адресов интерфейсов Ethernet шлюза PBM-ETH-3.0 нужно перейти в меню ETH Opera → Access Ethernet Option.

При использовании ПО PBConfi необходимо установить IP-адрес доступа программы таким же, как и IP-адрес порта шлюза PBM-ETH-3.0. (По умолчанию IP-адрес для подключения — 192.168.1.10 IP-адреса PBM-ETH-3.0 по умолчанию: ETH1 — 192.168.1.10, ETH2 — 192.168.2.10).

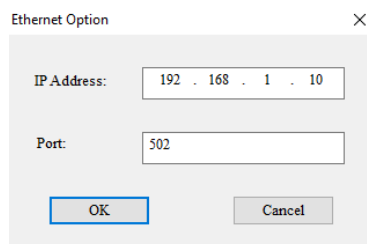


Рис. 6-11. Настройка IP-адреса для подключения ПО PBConfi

6.1.7 Загрузка конфигурации

Для загрузки конфигурации в шлюз PBM-ETH-3.0 нужно перейти в меню «Common Opera» → «Compile&Download» или непосредственно кликнуть по кнопке «Compile and Download» на панели инструментов — текущая конфигурация будет загружена в шлюз через порт Ethernet.

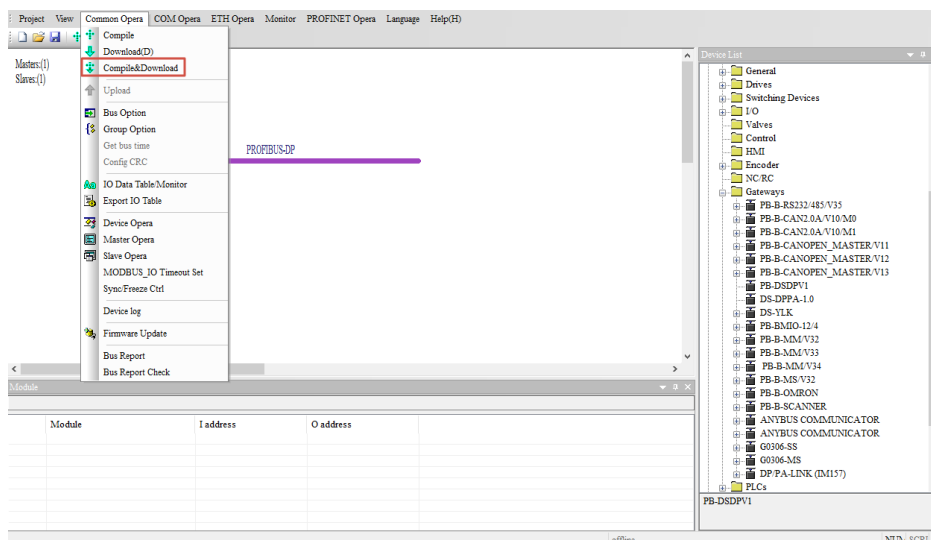


Рис. 6-12. Загрузка конфигурации в шлюз PBM-ETH-3.0

6.1.8 Настройка сетевых параметров PBM-ETH-3.0

Для настройки сетевых параметров шлюза PBM-ETH-3.0 нужно перейти в меню «ETH Opera» → «Ethernet Option Set» выбрать режим и изменить IP-адрес. Для применения изменений нужно после завершения загрузки перевести переключатель SW1 в положение OFF и перезапустить шлюз.

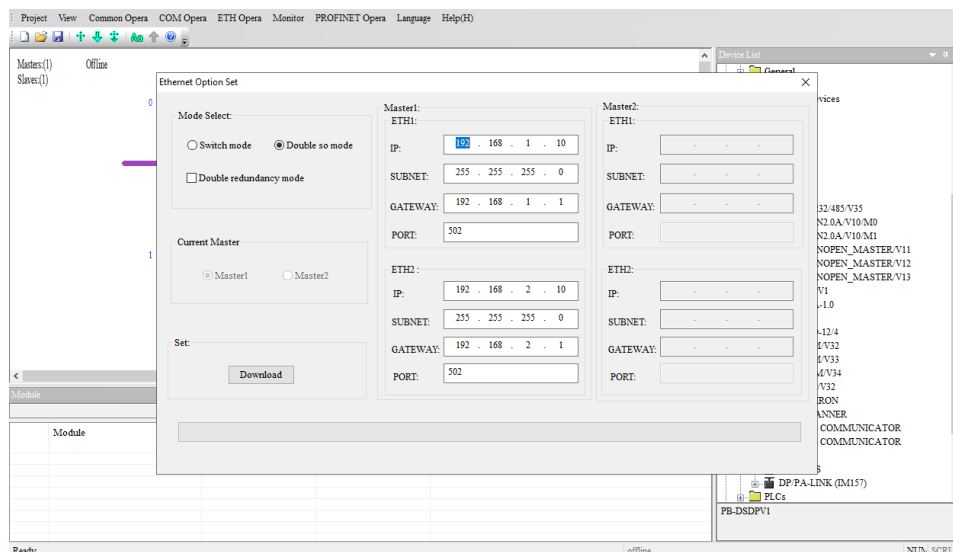


Рис. 6-13. Окно настроек сетевых параметров шлюза PBM-ETH-3.0

Примечание: в двухпортовом режиме IP-адреса первого и второго сетевых портов не должны находиться в одной подсети.

Работа шлюза в режиме коммутатора

В режиме коммутатора шлюз PBM-ETH-3.0 представляет собой для внешней сети один сетевой интерфейс с одним IP-адресом, два разъёма RJ45 можно рассматривать как два порта коммутатора. Связь с главным шлюзом может осуществляться через любой из двух сетевых интерфейсов. Режим работы коммутатора также может использоваться для построения сети по топологии «daisy chain».

Режим работы шлюза с двумя сетевыми портами

В режиме работы с двумя сетевыми портами шлюз RBM-ETH-3.0 представляет собой для внешней сети два независимых сетевых интерфейса с двумя IP-адресами, которые не должны находиться в одной подсети. Каждый из двух сетевых интерфейсов подключается к своей подсети. Режим работы с двумя сетевыми портами может использоваться для построения независимой резервированной двойной сети.

6.1.9 Сохранение и загрузка файла конфигурации RBM-ETH-3.0

Нажмите в меню «Project» → «Save», задайте имя файла и нажмите «Сохранить», как показано на Рис. 6-14.

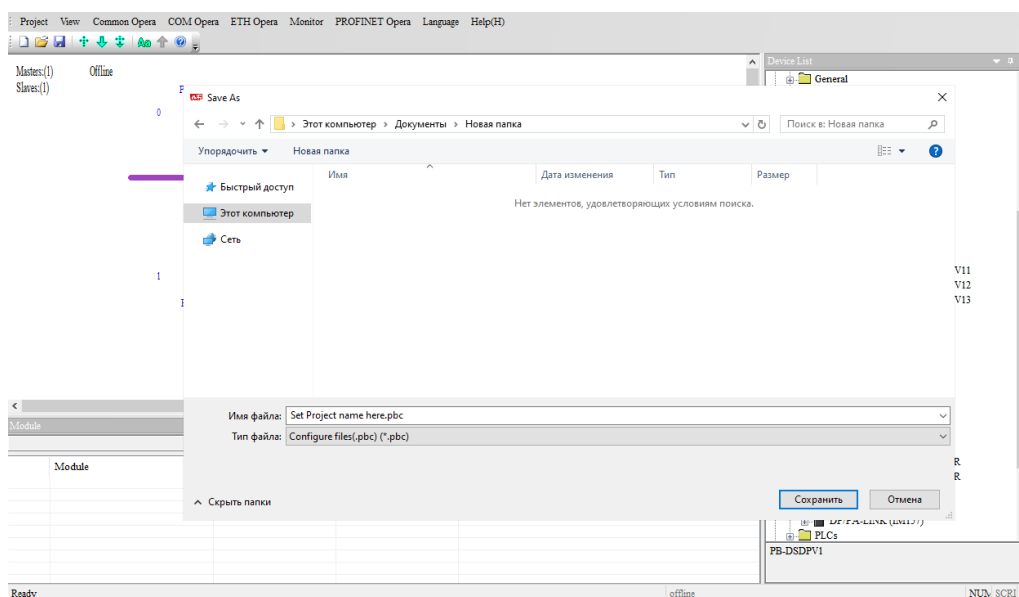


Рис. 6-14. Сохранение конфигурации

Глава 7. Онлайн-отладка с помощью PBConfi

Порядок работы: «Common Opera» → «Device Opera». Окно «Device Opera» используется для просмотра основной информации об устройстве, коррекции системного времени шлюза, мониторинга рабочей среды шлюза, а также для задания имени устройства и описания устройства.

The 'Device Opera' window is divided into several sections. The 'Device identify' section includes fields for 'Device type' (PBM-ETH-3.0), 'Device ID' (3779375F885169756F780301), 'Firmware ver.' (011E), and 'Hardware ver.' (0101), with an 'Identify' button. The 'The time calibration' section shows 'Computer time' (2026/08/26 09:56:14), 'Device time' (2026/06/29 01:31:52), and a 'Reset count' (0), with a 'Calibrate' button. The 'Vol' section displays 'Cur.' (1239 mV), 'Min.' (1205 mV), and 'Max.' (1243 mV). The 'Temp' section shows 'Cur.' (36 °C), 'Min.' (29 °C), and 'Max.' (41 °C). The 'Device Description' section has a 'Name' field (PBM2) and a 'Description' text area (PBM2), both with 'Set' buttons.

Рис. 7-1. Операции с устройством

7.1 Работа с функциями Master-устройства

Порядок работы: «Common Opera» → «Master Opera». Данное окно позволяет отслеживать рабочее состояние Ведущего устройства, настраивать параметры Ведущего устройства, управлять рабочим режимом Ведущего устройства и т.д.

The 'Master Control' window contains several control elements. At the top, 'Auto-run mode' is set to 'Enable' and 'Auto-stop mode' is set to 'Disable'. Below these are 'Send error count' (1) and 'Receive error count' (0). The 'Current config and Device' status is shown as 'consistent'. The 'C2 Timeout' is set to 1000 with a '*10ms' multiplier and a 'Modify' button. The 'Master state' is currently 'RUN', with 'RUN' and 'STOP' buttons available. At the bottom, there are two groups of 'clear' buttons: 'Master count clear' (Clear master send count, Clear master receive count) and 'Slave count clear' (Clear slave resend count, Clear slave offline count).

Рис. 7-2. Работа с функциями Master-устройства

Переключение состояния Ведущего устройства RUN/STOP осуществляется кнопками «RUN» и «STOP».

7.2 Работа с функциями Slave-устройства (функции DPV1)

Порядок работы: «Common Opera» → «Slave Opera». Данное окно отображает рабочее состояние Ведомых устройств, диагностическую информацию и служит для выполнения операций DPV1.

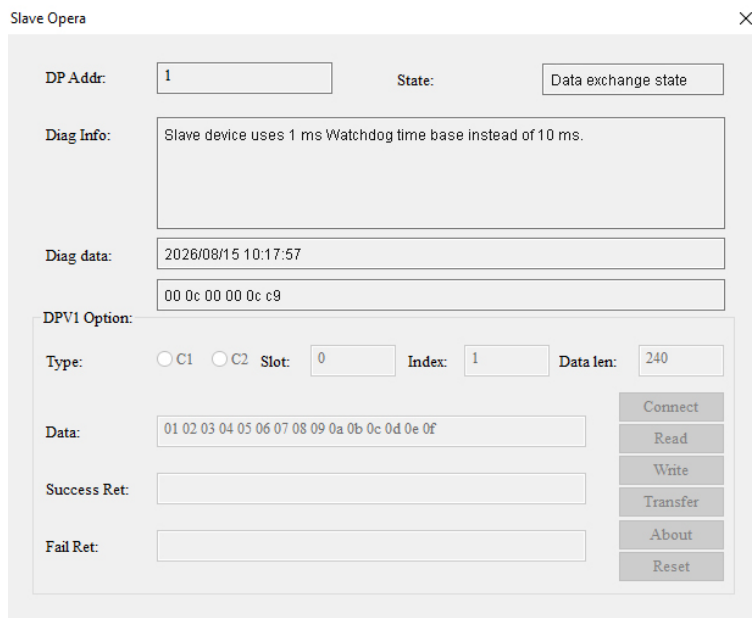


Рис. 7-3. Работа с функциями Slave-устройства

7.3 Мониторинг состояния устройств, подключённых к шлюзу

Получение рабочего состояния Ведущего устройства и отдельных Ведомых устройств — состояние доступно для всех Ведомых устройств, независимо от того, были они сконфигурированы или нет. Нажмите «Monitor» → «Start», после чего отобразится интерфейс, показанный ниже на Рис. 7-4.

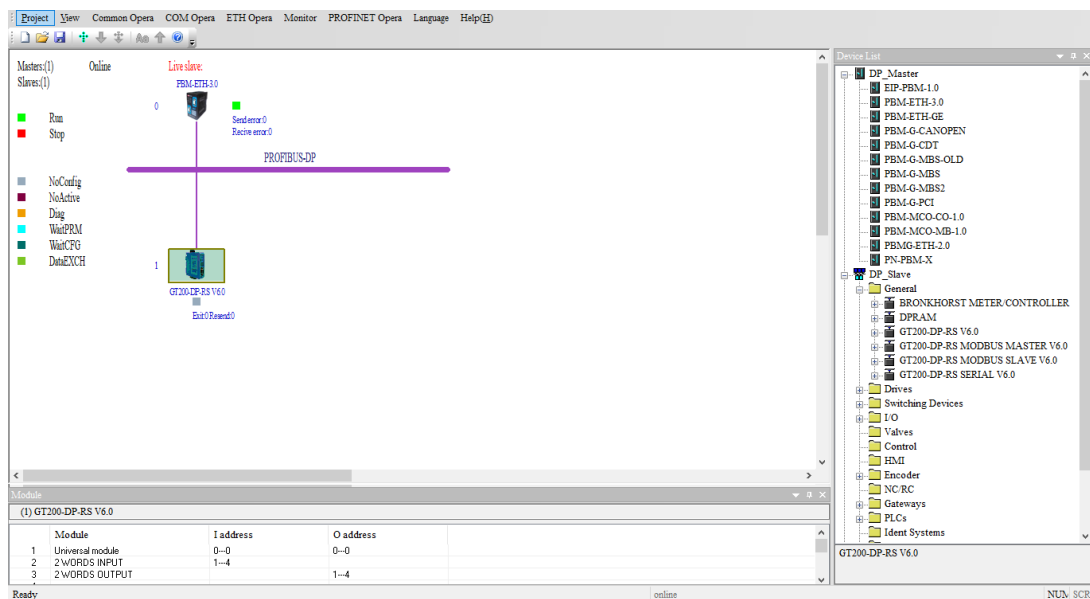


Рис. 7-4. Дополненный мониторингом интерфейс ПО PVBConf

Различные цветовые индикаторы значков отображают рабочее состояние Ведущего и Ведомых устройств. Под значком Ведущего устройства отображается количество ошибок при отправке телеграмм и количество ошибок при приёме телеграмм. Под значком Ведомого устройства отображается количество повторных попыток передачи телеграмм данному Ведомому устройству и количество случаев выхода данного Ведомого устройства из состояния обмена данными.

7.4 Мониторинг обмена I/O данными PROFIBUS DPV0

Для начала мониторинга передачи I/O данных необходимо перейти по следующему пути: «Common Opera» → «I/O Data Table Monitor». Откроется окно, показанное ниже на Рис. 7-5.

Table/Monitor

PROFIBUSADDRESS					MODBUSADDRESS					PB-Input MODBUS					PB-Output MODBUS				
No.	I(byte)	O(byte)	I(word)	O(word)		high	low			high	low								
*1	0---15	0---15	0---7	0---7	0-->	01	02	<--1		0-->	f0	f0	<--1						
		0---15		0---7	2-->	03	04	<--3		2-->	00	00	<--3						
	16---31		8---15		4-->	05	06	<--5		4-->	00	00	<--5						
		16---31		8---15	6-->	07	08	<--7		6-->	00	00	<--7						
	32---47		16---23		8-->	09	0a	<--9		8-->	00	00	<--9						
		32---47		16---23	10-->	0b	0c	<--11		10-->	00	00	<--11						
					12-->	0d	0e	<--13		12-->	00	00	<--13						
					14-->	0f	10	<--15		14-->	00	00	<--15						
					16-->	11	12	<--17		16-->	00	00	<--17						
					18-->	13	14	<--19		18-->	00	00	<--19						
					20-->	15	16	<--21		20-->	00	00	<--21						
					22-->	17	18	<--23		22-->	00	00	<--23						
					24-->	19	1a	<--25		24-->	00	00	<--25						
					26-->	1b	1c	<--27		26-->	00	00	<--27						
					28-->	1d	1e	<--29		28-->	00	00	<--29						
					30-->	1f	20	<--31		30-->	00	00	<--31						
					32-->	21	22	<--33		32-->	00	00	<--33						
					34-->	23	24	<--35		34-->	00	00	<--35						
					36-->	25	26	<--37		36-->	00	00	<--37						
					38-->	27	28	<--39		38-->	00	00	<--39						
					40-->	29	2a	<--41		40-->	00	00	<--41						
					d2-->	2b	2c	<--d3		d2-->	00	00	<--d3						

Cur Slave No:

Start

Рис. 7-5. Окно мониторинга передачи I/O данных

Выберите Ведомое устройство для мониторинга — перед номером соответствующей станции появится звёздочка «*», а в поле «Cur Slave No:» отобразится выбранное Ведомое устройство. Нажмите «Start», после чего Входные и Выходные данные этого Ведомого устройства отобразятся в областях данных PB-Input и PB-Output. Выберите данные в области PB-Output, измените их в окне «Modify» (поле «Data») и нажмите «OK» для отправки — изменённые данные будут отправлены соответствующему Ведомому устройству.

7.5 Экспорт текущего состояния всех Ведущих и Ведомых устройств системы

Для экспорта текущего состояния шлюза, Ведущего устройства и подключённых к нему Ведомых устройств необходимо воспользоваться соответствующей функцией по пути «Common Opera» → «Bus Report». После этого откроется окно отчёта, показанное на Рис. 7-6.

File: Протокол Формат Вид Справка

File name :Report-20260826-100514.txt
Note : this document for the software automatically generated, please do not modify!
Check code :879A68308DF8B7CB
Create time :PBConf Time:2026/08/26 10:05:14 Device Time:2026/08/29 01:40:52

Device status

Product Type :PBMG-ETH-3.0
Unique ID :3779375f885169756f780301
Hardware version :0101
Firmware version :011e
Device name :PBM2
Device description :PBM2
Restart Count :0
Working voltage :Min 1199mV Max 1243mV Cur 1212mV
Working temperature :Min 29°K Max 46°K Cur 41°K

Net state

Work model :Double so mode
Net Port 1 :IP addr 192.168.1.10 Port 502 SUBNET 255.255.255.0 Gateway 192.168.1.1 ConnNum 1
Net Port 2 :IP addr 192.168.2.10 Port 502 SUBNET 255.255.255.0 Gateway 192.168.2.1 ConnNum 0

Master state

Master addr :3
HSA :126
Baud :6M
Gap :1
Resend count :1
Tsl :600
Ttr(tbit) :32673
Real Ttr(tbit) :1042
Max Ttr(tbit) :3587
Send err count :0
Rev_err count :0
Auto mode :Auto opera Enable Auto stop Disable GE safe model Disable
Config state :The main configuration success
V0 state :RUN
C? Time:ntf(1Time) :1000

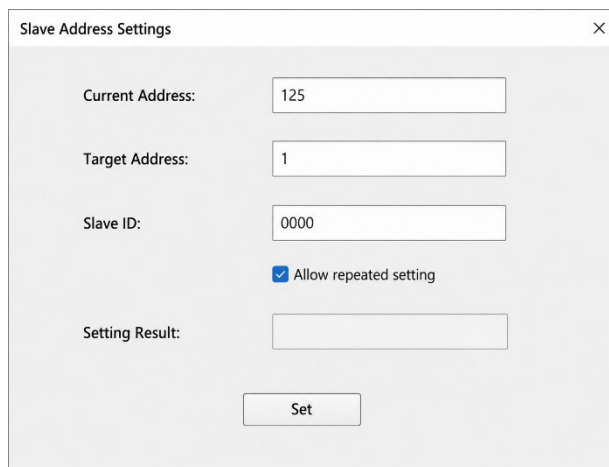
Ср. 1, cтраница 1 100% Windows (CRLF) ANSI

Рис. 7-6. Окно отчёта о работе шины PROFIBUS

7.6 Изменение адреса Ведомого устройства (специальные Slave-устройства)

Некоторые Ведомые устройства PROFIBUS DP не оснащены DIP-переключателями для задания сетевого адреса, вместо этого адрес задаётся с помощью специальной телеграммы установки адреса. Обычно заводской адрес таких Ведомых устройств по умолчанию равен 125. Заданный адрес сохраняется в энергонезависимой памяти Ведомого устройства и не теряется при последующем включении питания.

Программа PVBConf поддерживает функцию установки адреса Ведомого устройства. Для этого необходимо перейти к соответствующей функции по пути: «Common Opera» → «Slave Opera».



Slave Address Settings

Current Address: 125

Target Address: 1

Slave ID: 0000

☒ Allow repeated setting

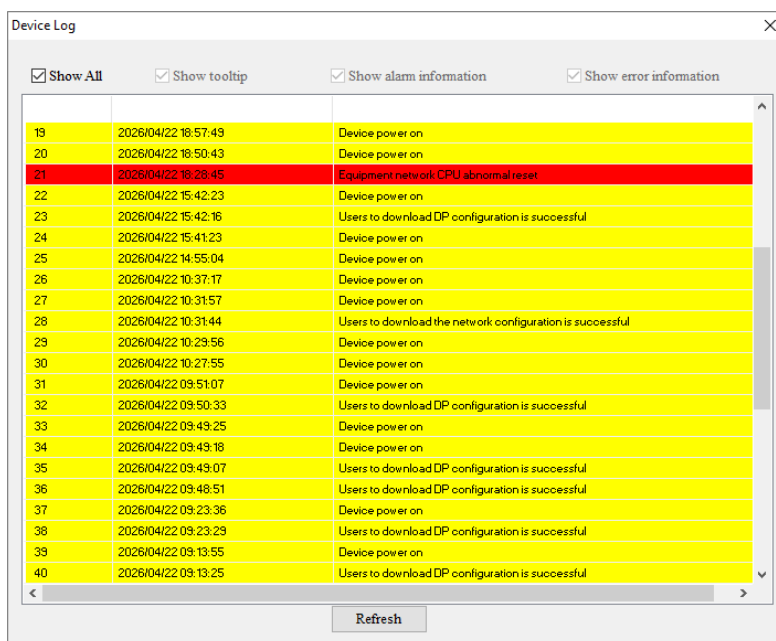
Setting Result:

Set

Рис. 7-7. Окно изменения сетевого адреса Ведомого устройства

7.7 Просмотр системного журнала (log)

Системный журнал можно просмотреть с помощью PVBConfig перейдя по пути «Common Opera» → «Device log». После выбора данной опции откроется окно, показанное ниже на Рис. 7-8.



Device Log

☒ Show All ☒ Show tooltip ☒ Show alarm information ☒ Show error information

19	2026/04/22 18:57:43	Device power on
20	2026/04/22 18:50:43	Device power on
21	2026/04/22 18:28:45	Equipment network CPU abnormal reset
22	2026/04/22 15:42:23	Device power on
23	2026/04/22 15:42:16	Users to download DP configuration is successful
24	2026/04/22 15:41:23	Device power on
25	2026/04/22 14:55:04	Device power on
26	2026/04/22 10:37:17	Device power on
27	2026/04/22 10:31:57	Device power on
28	2026/04/22 10:31:44	Users to download the network configuration is successful
29	2026/04/22 10:29:56	Device power on
30	2026/04/22 10:27:55	Device power on
31	2026/04/22 09:51:07	Device power on
32	2026/04/22 09:50:33	Users to download DP configuration is successful
33	2026/04/22 09:49:25	Device power on
34	2026/04/22 09:49:18	Device power on
35	2026/04/22 09:49:07	Users to download DP configuration is successful
36	2026/04/22 09:48:51	Users to download DP configuration is successful
37	2026/04/22 09:23:36	Device power on
38	2026/04/22 09:23:29	Users to download DP configuration is successful
39	2026/04/22 09:13:55	Device power on
40	2026/04/22 09:13:25	Users to download DP configuration is successful

Refresh

Рис. 7-8. Системный журнал

События с идентификатором (ID) меньше 0x0013 относятся к критическим событиям. Эти события сохраняются в энергонезависимой памяти шлюза и не теряются при отключении питания. Шлюз может одновременно хранить до 256 записей системного журнала. Когда количество записей превышает 256, самые новые записи перезаписывают самые старые.

Глава 8. Обновление встроенного ПО (прошивки)

Для своевременного исправления программных дефектов шлюза или обновления функциональных возможностей изделия PBM-ETH-3.0 предусматривает функцию обновления встроенного ПО по сети Ethernet.

Для использования функции обновления встроенного ПО необходимо, чтобы шлюз работал в режиме обновления прошивки. Установите DIP-переключатель SW3 («Обновление прошивки») в положение, соответствующее режиму обновления (SW3 = 1), и перезапустите шлюз по питанию. Если канал связи Ethernet исправен, индикатор SYS перейдет из состояния постоянного красного свечения в состояние постоянного зеленого свечения; в противном случае индикатор SYS будет оставаться в состоянии постоянного красного свечения до тех пор, пока соединение по Ethernet не будет установлено.

Запустите программу PBConf, откройте окно обновления встроенного ПО ghb gjvjob меню «Common Opera» → «Firmware Update». В раскрывающемся списке выберите версию прошивки, до которой необходимо обновить устройство, и кликните по кнопке «Update» для запуска процесса обновления. Непрерывное заполнение индикатора выполнения свидетельствует о том, что обновление встроенного ПО выполняется.

В процессе обновления прошивки индикатор SYS должен мигать зеленым цветом. При успешном завершении обновления индикатор выполнения в ПО заполнится до конца и будет выдано соответствующее уведомление; одновременно индикатор SYS перейдет в состояние постоянного зеленого свечения. После завершения обновления верните DIP-переключатель SW3 в положение SW3 = 0, соответствующее нормальному рабочему режиму, и перезапустите шлюз по питанию.

Если обновление прошивки прошло успешно, устройство запустится с новейшей версией встроенного ПО, при этом индикатор SYS будет светиться постоянным зеленым цветом. Если обновление встроенного ПО не было успешно завершено по какой-либо причине (закрытие программы PBConf, отсоединение сетевого кабеля, отключение питания шлюза и т. п.), то после переключения в нормальный режим и запуска индикатор SYS будет мигать красным цветом, указывая на сбой запуска рабочей прошивки. В этом случае необходимо снова перейти в режим обновления прошивки и повторно обновить встроенное ПО устройства.

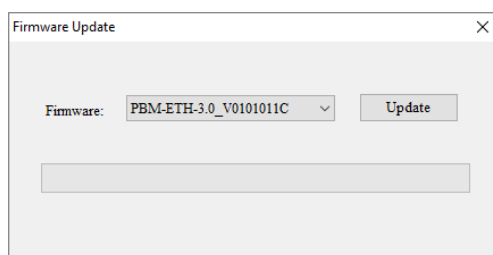


Рис. 8-1. Окно обновления встроенного ПО (прошивки)

Глава 9. Мониторинг состояния Ведомых устройств PROFIBUS

В данной главе описывается, как с помощью устройства Modbus-TCP Client (Master) можно реализовать мониторинг Ведомых устройств PROFIBUS-DP через шлюз PBM-ETH-3.0. Как было указано предыдущих главах, это осуществляется путём чтения и записи регистров MODBUS. Прежде всего необходимо ознакомиться с областями данных MODBUS шлюза. Значение областей данных MODBUS для PBM-ETH-3.0 приведено в таблице ниже.

Таблица 9-1. Области данных MODBUS шлюза.

Область MODBUS	Наименование	Атрибут операции	Функциональный код	Назначение данных
Область MODBUS 3x	Регистры ввода (Input Registers)	Только чтение	0x04: Чтение входных регистров	- Входные данные PROFIBUS DPV0 - Данные чтения PROFIBUS DPV1 - Диагностические данные Ведомых устройств PROFIBUS - Данные системного журнала шлюза - Регистры состояния шлюза
Область MODBUS 4x	Регистры хранения (Holding Registers)	Чтение и запись	0x03: Чтение регистров хранения 0x06: Запись одного регистра 0x10: Запись нескольких регистров	- Выходные данные PROFIBUS DPV0 - Данные записи PROFIBUS DPV1 - Управляющие регистры шлюза

9.1 Область входных регистров

Область данных MODBUS 3x подразделяется на область Входных данных DPV0, область данных, полученных по DPV1 (DPV1 Read), область диагностических данных DP, область данных системного журнала и область регистров состояния шлюза. Каждый запрос чтения MODBUS может обращаться только к одной области данных; пересекающийся (кросс-областной) доступ не допускается. Функциональное назначение адресов данных области MODBUS 3x для PBM-ETH-3.0 приведено в таблице ниже.

Таблица 9-2. Функциональное назначение адресов данных области MODBUS 3x.

Тип	Наименование	Адрес	Длина (слов)	Описание
Область Входных данных DPV0	DPV0_IDATA	0x0000~0x0FFF (0~4095)	4k (4096)	Входные данные PROFIBUS DPV0
Область Входных данных DPV1C1	DPV1_C1RSP	0x2000~0x207F (8192~8319)	128	Входные данные PROFIBUS DPV1C1
Область Входных данных DPV1C2	DPV1_C2RSP	0x3000~0x307F (12288~12415)	128	Входные данные PROFIBUS DPV1C2
Область диагностических данных DP	SLAVE_DIAGRSP	0x4000~0x407F (16384~16511)	128	Диагностические данные PROFIBUS DIAG
Область данных системного журнала	SYSLOG	0x5000~0x55FF (20480~22015)	1280	Данные системного журнала, максимальный объём хранения — 256 записей

Таблица 9-3. Область Регистров состояния системы (часть 1).

Тип	Адрес	Длина (слов)	Описание
Область Регистров состояния системы	0x8000 (32768)	1	Идентификатор устройства (ID), значение: 0x06FA.
	0x8001 (32769)	6	Уникальный идентификатор устройства (ID); каждое устройство имеет уникальный ID.
	0x8007 (32775)	1	Версия аппаратного обеспечения.
	0x8008 (32776)	1	Версия встроенного ПО: старший байт — номер основной версии, младший байт — номер дополнительной версии.
	0x8009 (32777)	1	Регистр состояния системы *
	0x800A (32778)	1	Регистр счётчика перезапусков системы: старший бит (бит 15) — бит переполнения; биты 14–0 — разряды счётчика. При переполнении счётчика бит переполнения остаётся в состоянии 1.
	0x800B (32779)	1	Количество записей в области системного журнала
	Reserved		Зарезервировано
	0x8015 (32789)	1	Текущее значение напряжения питания устройства в течение данного сеанса работы (с момента подачи питания), в мВ.
	0x8016 (32790)	1	Минимальное значение напряжения питания устройства в течение данного сеанса работы (с момента подачи питания), в мВ.
	0x8017 (32791)	1	Максимальное значение напряжения питания устройства в течение данного сеанса работы (с момента подачи питания), в мВ.
	0x8018 (32792)	1	Текущее значение рабочей температуры устройства в течение данного сеанса работы (с момента подачи питания), в дополнительном коде, в градусах Цельсия.
	0x8019 (32793)	1	Минимальное значение рабочей температуры устройства в течение данного сеанса работы (с момента подачи питания), в дополнительном коде, в градусах Цельсия.
	0x801A (32794)	1	Максимальное значение рабочей температуры устройства в течение данного сеанса работы (с момента подачи питания), в дополнительном коде, в градусах Цельсия.
	Reserved		Зарезервировано
	0x801F (32799)	1	Значение CRC текущей конфигурации
	0x8020 (32800)	2	IP-адрес сетевого интерфейса Ethern 1 шлюза, заданный пользователем. Старший байт первого слова соответствует младшему байту сетевого адреса; младший байт последнего слова соответствует старшему байту сетевого адреса.

Таблица 9-4. Область Регистров состояния системы (часть 2).

Тип	Адрес	Длина (слов)	Описание
Область регистров состояния системы	0x8022 (32802)	2	Маска подсети сетевого интерфейса ETN1, заданная пользователем. Старший байт первого слова соответствует младшему байту сетевого адреса; младший байт последнего слова соответствует старшему байту сетевого адреса.
	0x8024 (32804)	2	IP-адрес основного шлюза (шлюза по умолчанию) для сетевого интерфейса ETN1, заданный пользователем. Старший байт первого слова соответствует младшему байту сетевого адреса; младший байт последнего слова соответствует старшему байту сетевого адреса.
	0x8026 (32806)	1	Номер порта службы Modbus-TCP для сетевого интерфейса ETN1, заданный пользователем.
	0x8027 (32807)	1	Количество TCP-подключений сетевого интерфейса ETN1
	0x8028 (32808)	2	IP-адрес сетевого интерфейса ETN2, заданный пользователем. Старший байт первого слова соответствует младшему байту сетевого адреса; младший байт последнего слова соответствует старшему байту сетевого адреса.
	0x802A (32810)	2	Маска подсети сетевого интерфейса ETN2, заданная пользователем. Старший байт первого слова соответствует младшему байту сетевого адреса; младший байт последнего слова соответствует старшему байту сетевого адреса.
	0x802C (32812)	2	IP-адрес основного шлюза (шлюза по умолчанию) для сетевого интерфейса ETN2, заданный пользователем. Старший байт первого слова соответствует младшему байту сетевого адреса; младший байт последнего слова соответствует старшему байту сетевого адреса.
	0x802E (32814)	1	Номер порта службы Modbus-TCP для сетевого интерфейса ETN2, заданный пользователем.
	0x802F (32815)	1	Количество TCP-подключений сетевого интерфейса ETN2
	Reserved		Зарезервировано

Таблица 9-5. Область регистров состояния Ведущего устройства.

Тип	Адрес	Длина (слов)	Описание
Область регистров состояния Ведущего устройства	0x8040 (32832)	1	Сетевой адрес Ведущего устройства и максимальный адрес станции: младший байт — максимальный адрес станции (HSA), старший байт — адрес Ведущего устройства.
	0x8041 (32833)	1	Скорость передачи данных Ведущего устройства (бодовая скорость). Значение и её расшифровка: 1:9.6k, 2:19.2k, 3:45.45k, 4:93.75k, 5:187,5k 6:500k, 7:1.5M, 8:3M, 9:6M
	0x8042 (32834)	1	GAP и Retry: старший байт — GAP (через какое количество передач токена отправляется одна телеграмма FDL), младший байт — количество повторных попыток (Retry).
	0x8043 (32835)	1	Период тайм-слота Ведущего устройства (Tslot)
	0x8044 (32836)	1	Целевое время оборота токена Ведущего устройства (Ttr), старшее слово
	0x8045 (32837)	1	Целевое время оборота токена ведущего устройства (Ttr), младшее слово
	Reserved		Зарезервировано
	0x8047 (32839)	1	Регистр автоматического режима Ведущего устройства *
	0x8048 (32840)	1	Регистр конфигурации Ведущего устройства *
	0x8049 (32841)	1	Регистр состояния конфигурации Ведущего устройства *
	0x804A (32842)	1	Фактическое время оборота токена Ведущего устройства (Ttr), старшее слово
	0x804B (32843)	1	Фактическое время оборота токена Ведущего устройства (Ttr), младшее слово
	0x804C (32844)	1	Максимальное время оборота токена Ведущего устройства, старшее слово (максимальное время оборота токена не может превышать Ttr)
	0x804D (32845)	1	Максимальное время оборота токена Ведущего устройства, младшее слово (максимальное время оборота токена не может превышать Ttr)
	Reserved		Зарезервировано
	0x8051 (32849)	1	Счётчик ошибок при отправке телеграмм DP ведущим устройством: старший бит (бит 15) — бит переполнения; биты 14–0 — разряды счётчика. При переполнении счётчика бит переполнения остаётся в состоянии 1.
	0x8052 (32850)	1	Счётчик ошибок при приёме телеграмм DP ведущим устройством: старший бит (бит 15) — бит переполнения; биты 14–0 — разряды счётчика. При переполнении счётчика бит переполнения остаётся в состоянии 1.

Таблица 9-6. Область регистров состояния ведомых устройств.

Тип	Адрес	Длина (слов)	Описание
Область регистров состояния Ведомых устройств	Reserved		Зарезервировано
	0x805B (32859)	1	Регистр количества сконфигурированных Ведомых устройств
	0x805C (32860)	1	Регистр состояния установки сетевого адреса Ведомого устройства
	0x805D (32861)	1	Регистр состояния Ведомого устройства 0 *
			
	0x80DA (32986)	1	Регистр состояния Ведомого устройства 125
	0x80DB (32987)	1	Регистр количества выходов Ведомого устройства 0 из состояния обмена данными: старший бит (бит 15) — бит переполнения; биты 14–0 — разряды счётчика. При переполнении счётчика бит переполнения остаётся в состоянии 1.
			
	0x8158 (33112)	1	Регистр количества выходов Ведомого устройства 125 из состояния обмена данными: старший бит (бит 15) — бит переполнения; биты 14–0 — разряды счётчика. При переполнении счётчика бит переполнения остаётся в состоянии 1.
	0x8159 (33113)	1	Регистр количества повторных попыток передачи данных для Ведомого устройства 0: старший бит (бит 15) — бит переполнения; биты 14–0 — разряды счётчика. При переполнении счётчика бит переполнения остаётся в состоянии 1.
			
	0x81D6 (33238)	1	Регистр количества повторных попыток передачи данных для Ведомого устройства 125: старший бит (бит 15) — бит переполнения; биты 14–0 — разряды счётчика. При переполнении счётчика бит переполнения остаётся в состоянии 1.

Примечания:

(1) Указанные выше адреса являются адресами данных в телеграмме MODBUS, адресация начинается с 0. При использовании инструмента ModScan (Modbus Poll) вводимый адрес должен быть на 1 больше вышеуказанного (адрес + 1).

(2) Подробное описание значений, отмеченных знаком «*», приведено в разделе 9.7 данной главы «Подробное описание функционального назначения областей регистров Modbus шлюза РВМ-ЕТН-3.0».

9.2 Область регистров хранения (Holding Registers)

Область данных Modbus 4х подразделяется на область выходных данных DPV0, область данных запроса DPV1 и область управляющих регистров. Каждый запрос Modbus может обращаться только к одной области данных; пересекающийся (кросс-областной) доступ не допускается. Функциональное назначение адресов данных области Modbus 4х для PBM-ETH-3.0 приведено в таблице ниже.

Таблица 9-7. Функциональное назначение адресов данных области Modbus 4х.

Тип	Наименование	Адрес	Длина (слов)	Описание
Область Выходных данных DPV0	DPV0_ODATA	0x0000 ~ 0x0FFF (0 ~ 4095)	4k	Выходные данные PROFIBUS DPV0
Область данных запроса DPV1C1	DPV1_C1REQ	0x2000 ~ 0x207F (8192 ~ 8319)	128	Телеграмма запроса PROFIBUS DPV1C1
Область данных запроса DPV1C2	DPV1_C2REQ	0x3000 ~ 0x307F (12288 ~ 12415)	128	Телеграмма запроса PROFIBUS DPV1C2
Область управляющих регистров системы	RUNTIME_YEAR	0x8000 (32768)	1	Время работы устройства: год *
	RUNTIME_MONDAY	0x8001 (32769)	1	Время работы устройства: месяц и день *
	RUNTIME_HOUR_MIN	0x8002 (32770)	1	Время работы устройства: часы и минуты *
	RUNTIME_MSEC	0x8003 (32771)	1	Время работы устройства: миллисекунды *
	DEVICE_CTRL	0x8004 (32772)	1	Управляющий регистр устройства: bit2 = 1 - идентификация устройства включена, индикатор SYS мигает зелёным цветом; bit2 = 0 - идентификация устройства выключена.
		Reserved		Зарезервировано
	DEVICE_NAME	0x8010 ~ 0x802F (32784 ~ 32815)	32	Имя устройства. Сохраняется в энергонезависимой памяти (не теряется при отключении питания).
	DEVICE_DESCRIP	0x8030 ~ 0x80AF (32816 ~ 32943)	128	Описание устройства. Сохраняется в энергонезависимой памяти (не теряется при отключении питания).

Таблица 9-8. Область управляющих регистров.

Тип	Наименование	Адрес	Длина (слов)	Описание
Область управл. регистров Ведущего устройства	MASTER_OPCTRL	0x80B0 (32944)	1	Регистр конечного автомата Ведущего устройства (Master State Machine) *
		Reserved		
	MASTER_C2RSPTOUT	0x80B2 (32946)	1	Регистр времени ожидания ответа DPV1C2 Ведущего устройства. После записи не теряется при отключении питания. Допустимый диапазон значений регистра: 100–3000. Фактическое время ожидания определяется как значение регистра × 10 мс. Значение данного регистра должно быть больше значения атрибута C2_Response_Timeout, указанного в GSD-файле Ведомого устройства C2.
	MASTER_COMCNTCLR	0x80B3 (32947)	1	Регистр сброса счётчиков связи Ведущего устройства *
		Reserved		Зарезервировано
Область управл. регистров Ведомых устройств	SLAVE_SYNFRZCTRL	0x80C0 (32960)	1	Регистр управления синхронизацией и фиксацией Ведомых устройств (Sync/Freeze) *
	SLAVE_COMCNTCLR	0x80C1 (32961)	1	Регистр сброса счётчиков связи Ведомого устройства *
	SLAVE_SETSADDR_ADDR	0x80C2 (32962)	1	Регистр установки адреса Ведомого устройства: старший байт — текущий адрес Ведомого устройства, младший байт — назначаемый (новый) адрес.
	SLAVE_SETSADDR_IDNUM	0x80C3 (32963)	1	Регистр Ident Number для установки адреса Ведомого устройства
	SLAVE_SETSADDR_NOCHG	0x80C4 (32964)	1	Регистр разрешения повторного изменения адреса Ведомого устройства: 0 — повторное изменение разрешено; ненулевое значение — повторное изменение запрещено.

Примечания:

(1) Указанные выше адреса являются адресами данных в телеграмме Modbus, адресация начинается с 0. При использовании инструмента ModScan (или Modbus Poll) вводимый адрес должен быть на 1 больше вышеуказанного (адрес + 1).

(2) Подробное описание значений, отмеченных знаком «*», приведено в разделе 9.7 данной главы «Подробное описание функционального назначения областей регистров Modbus шлюза РВМ-ETH-3.0».

9.3 Коды ответов Modbus шлюза PBM-ETH-3.0

Шлюз анализирует запросы Modbus-TCP, поступающие от Клиента (Master-устройства Modbus). Если запрос на обслуживание содержит ошибку или шлюз не может выполнить данный запрос, он возвращает соответствующий код ошибки в ответной транзакции Modbus. Возможные коды ошибок, возвращаемые шлюзом, и их значение приведены в таблице ниже.

Таблица 9-9. Значения кодов ответов протокола Modbus шлюза PBM-ETH-3.0.

Наименование кода ответа Modbus	Код ответа	Описание
MBSRET_INVALIDFC	0x01	Недопустимый функциональный код Modbus
MBSRET_OFFSETERR	0x02	Недопустимый адрес данных
MBSRET_VALUEERR	0x03	Недопустимое значение данных
MBSRET_AREA_INVALID	0x10	Обращение к недопустимой области Modbus
MBSRET_AREA_CROSS	0x11	Одновременное обращение к двум областям данных Modbus
MBSRET_SERVIC_BUSY	0x12	Устройство занято обработкой запроса
MBSRET_SETTIME_FAIL	0x20	Ошибка установки системного времени Ведущего устройства
MBSRET_GETLOG_FAIL	0x21	Ошибка получения системного журнала
MBSRET_USRCONF_ERR	0x40	Ошибка пользовательской конфигурации Ведущего устройства
MBSRET_NOTPMaster_ERR	0x41	Текущее Ведущее устройство не является активным Ведущим устройством PROFIBUS (P-Master)
MBSRET_REDUSW_FAIL	0x42	Ошибка переключения резервного Ведущего устройства
MBSRET_RUNSTOPSW_FAIL	0x43	Ошибка переключения RUN/STOP Ведущего устройства
MBSRET_HIGH_DIAG	0x60	Аварийное сообщение (Alarm) высокого приоритета от Ведомого устройства
MBSRET_SLAVE_OFFLINE	0x61	Запрашиваемое ведомое устройство отключено (Ведущее устройство отправило ему телеграмму SD2, ответ не получен)
MBSRET_SLAVE_NDEX	0x62	Запрашиваемое Ведомое устройство не находится в состоянии обмена данными (Data Exchange)
MBSRET_V0_UNREADY	0x63	Данные DPV0 запрашиваемого Ведомого устройства не готовы
MBSRET_SYNFRZ_FAIL	0x64	Ошибка выполнения операции синхронизации/фиксации (Sync/Freeze) Ведомого устройства
MBSRET_V1_FAIL	0x65	Ошибка выполнения запроса операции DPV1 Ведомого устройства
MBSRET_V1_UNREADY	0x66	Ответ DPV1 ведомого устройства не готов
MBSRET_V1_NOTSUPP	0x67	Ведомое устройство не поддерживает функцию DPV1
MBSRET_SETSADDR_FAIL	0x68	Ошибка установки адреса Ведомого устройства

9.4 Работа с областью данных Modbus DPV0 шлюза PBM-ETH-3.0

Ведущее устройство Modbus-TCP Client может получать Входные данные DPV0 Ведомых устройств путём чтения Входных данных DPV0 из области Modbus 3х шлюза PBM-ETH-3.0. Выходные данные DPV0 Ведомых устройств обновляются путём записи Выходных данных DPV0 в область Modbus 4х шлюза PBM-ETH-3.0.

9.4.1 Операция чтения данных DPV0

В данном разделе в табличной форме показан только принцип отображения Входных данных PROFIBUS DPV0 в области данных Modbus. Конкретное соотношение адресов отображения задаётся таблицей адресного мэппинга, формируемой конфигурационным программным обеспечением PBConfi.

Таблица 9-10. Таблица мэппинга адресов (отображения адресов).

Относительное смещение адреса (в словах)	Тип	Описание данных	Примечания
0000	Входные данные DPV0 сконфигурированного Ведомого устройства 1	Входные данные Module 1 данного Ведомого устройства	Ведомое устройство обычно содержит один или несколько модулей (Module), поэтому Входные данные DPV0 Ведомого устройства также состоят из нескольких Модулей Входных данных. Если длина данных Module нечётная, то в последнем слове Входных данных этого Module в области данных Modbus: • старший байт содержит действительные Входные данные; • младший байт равен нулю.
.....		Входные данные Module 2 данного Ведомого устройства	
.....			
n		Входные данные Module X данного Ведомого устройства	
n + 0001	Входные данные DPV0 сконфигурированного Ведомого устройства 2	Входные данные Module 1 данного Ведомого устройства	
.....		Входные данные Module 2 данного Ведомого устройства	
.....			
m		Входные данные Module X данного Ведомого устройства	
m + 0001	Входные данные DPV0 сконфигурированного Ведомого устройства 3		

Ведомое устройство может состоять из одного или нескольких Модулей (Module). На Рис. 9-1 ниже упрощённо показано соотношение отображения Входных данных отдельных Модулей Ведомого устройства в области данных PROFIBUS и в области Modbus 3.

[Пример] Ведомое устройство 1 имеет в общей сложности 10 байт входных данных и состоит из трёх Module: 2 байта, 5 байт и 3 байта соответственно.

- Module 1 содержит 2 байта Входных данных. Относительное смещение адреса в области данных PROFIBUS — байты 0–1 (два байта); в области Modbus 3 — адрес 30001 (одно слово).
- Module 2 содержит 5 байт Входных данных. Относительное смещение адреса в области данных PROFIBUS — байты 2–6 (пять байт); в области Modbus 3 — адреса 30002–30004 (три слова).
- Module 3 содержит 3 байта входных данных. Относительное смещение адреса в области данных PROFIBUS — байты 7–9 (три байта); в области Modbus 3 — адреса 30005–30006 (два слова).

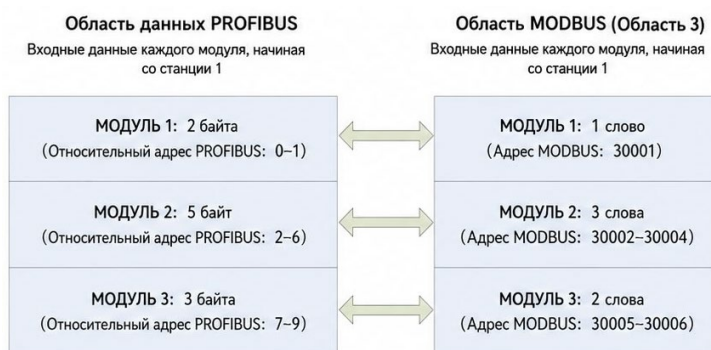


Рис. 9-1. Соотношение Входных данных Модулей Ведомого устройства

На Рис. 9-2 ниже показано соотношение отображения данных внутри отдельных Модулей в области данных PROFIBUS и в области данных Modbus.

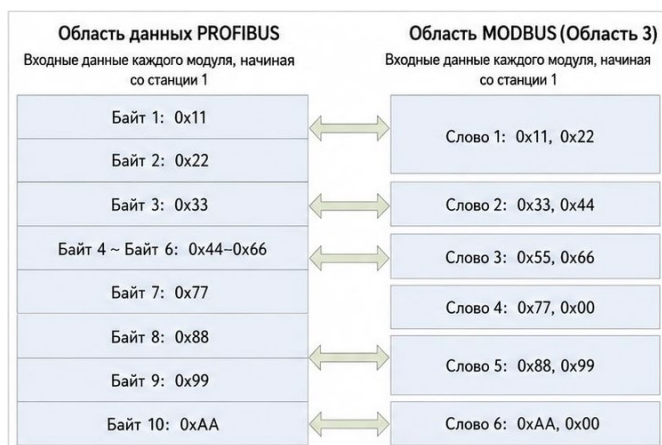


Рис. 9-2. Соотношение отображения данных внутри Модулей

Возможные ответные сообщения при выполнении операции чтения данных PROFIBUS DPV0 клиентом Modbus-TCP приведены ниже.

- Нормальный ответ с данными
- MBSRET_AREA_CROSS — одновременное обращение к двум функциональным областям Modbus
- MBSRET_USRCONF_ERR — ошибка конфигурации Ведущего устройства
- MBSRET_NOTPMaster_ERR — текущее Ведущее устройство не является активным P-Master
- MBSRET_HIGH_DIAG — аварийное сообщение высокого приоритета от Ведомого устройства
- MBSRET_SLAVE_NDEX — запрашиваемое Ведомое устройство не находится в состоянии обмена данными
- MBSRET_SLAVE_DURDY — данные DPV0 запрашиваемого Ведомого устройства не готовы

9.4.2 Операция записи данных DPV0

Пользователь может изменять Выходные данные DPV0 путём записи в область Modbus 4x.

В пределах установленного диапазона адресов смещение адреса и длина каждого запроса записи в область Modbus 4x не ограничены — за одну операцию можно записать Выходные данные одного или нескольких Ведомых устройств. Начальный адрес смещения не обязан совпадать с начальным адресом Modbus какого-либо Ведомого устройства.

При нормальном обмене данными по протоколу PROFIBUS DP между Ведущим и Ведомыми устройствами, когда Ведомые устройства уже находятся в состоянии обмена данными (Data Exchange), изменение Выходных данных DPV0, записанных пользователем в область Modbus 4x, зависит от рабочего состояния шлюза и от того, является ли он активным PROFIBUS Master-устройством.

Записанные пользователем в область Modbus 4x данные становятся действительными Выходными данными только в том случае, если Ведущее устройство находится в состоянии RUN.

В данном документе в табличной форме показан только принцип отображения Выходных данных PROFIBUS DPV0 в области данных Modbus. Конкретное соотношение адресов отображения задаётся конфигурационным файлом, формируемым программным обеспечением PBConfi.

Таблица 9-11. Таблица мэппинга адресов (отображения адресов).

Относительное смещение адреса (в словах)	Тип	Описание данных	Примечания
0000	Выходные данные DPV0 сконфигурированного Ведомого устройства 1	Выходные данные Module 1 данного Ведомого устройства	Ведомое устройство обычно содержит один или несколько модулей (Module), поэтому Выходные данные DPV0 Ведомого устройства также состоят из нескольких Модулей Выходных данных. Если длина данных Module нечётная, то в последнем слове Выходных данных этого Модуля в области данных Modbus:
.....		Выходные данные Module 2 данного Ведомого устройства	
.....			
n		Выходные данные Module X данного Ведомого устройства	
n + 0001	Выходные данные DPV0 сконфигурированного Ведомого устройства 2	Выходные данные Module 1 данного Ведомого устройства	• старший байт содержит действительные Выходные данные; • младший байт равен нулю.
.....		Выходные данные Module 2 данного Ведомого устройства	
.....			
m		Выходные данные Module X данного Ведомого устройства	
m + 0001	Выходные данные DPV0 сконфигурированного Ведомого устройства 3		

Ведомое устройство может состоять из одного или нескольких Модулей. На Рис. 9-3 ниже упрощённо показано соотношение отображения Выходных данных отдельных Модулей Ведомого устройства в области данных PROFIBUS и в области Modbus 4х.

Поскольку способ отображения Выходных данных совпадает со способом отображения Выходных данных, подробное описание здесь не приводится.

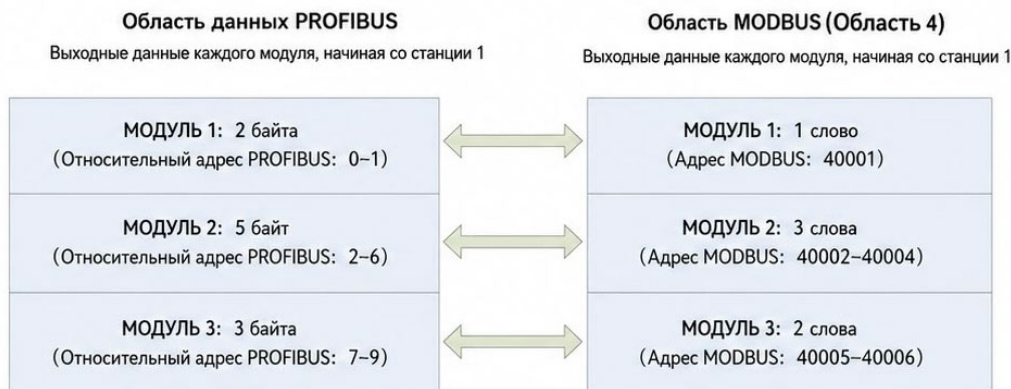


Рис. 9-3. Соотношение отображения Выходных данных Модулей Ведомого устройства

Возможные ответные сообщения при выполнении операции записи данных PROFIBUS DPV0 клиентом Modbus/TCP приведены ниже.

- Нормальный ответ (данные приняты)
- MBSRET_AREA_CROSS — пересекающееся обращение к двум функциональным областям Modbus
- MBSRET_USRCONF_ERR — ошибка пользовательской конфигурации Ведущего устройства
- MBSRET_HIGH_DIAG — аварийное сообщение высокого приоритета от Ведомого устройства
- MBSRET_SLAVE_NDEX — запрашиваемое Ведомое устройство не находится в состоянии обмена данными
- MBSRET_SLAVE_DURDY — данные DPV0 запрашиваемого Ведомого устройства не готовы

9.5 Работа с областью данных Modbus DPV1 шлюза PBM-ETH-3.0

Шлюз PBM-ETH-3.0 поддерживает функции Ведущего устройства DPV1C1 и DPV1C2. Чтобы инициировать запрос DPV1, пользователь записывает данные в область запроса DPV1 (область Modbus 4х). **Запрос DPV1 можно инициировать только записью по начальному адресу 0x2000 этой области.** Получив запрос, шлюз отправляет его соответствующему Ведомому устройству.

Обмен данными DPV1 — ациклический, поэтому ответ от Ведомого устройства поступает не сразу, а через несколько циклов шины. Получив ответ, шлюз помещает его в область ответа DPV1 (область Modbus 3х). Чтобы узнать, поступил ли ответ, и получить ответные данные нужно выполнить операцию чтения из этой области. **Операция Чтения возможно только с начального адреса 0x2000. Шлюз возвращает пользователю результат ответа на последний запрос DPV1.**

Обмен данными DPV1 с Ведомым устройством возможен, только когда Ведущее устройство находится в состоянии RUN. За один раз пользователь может выполнить операцию только с одним слотом-индексом одного Ведомого устройства. Пока текущая операция чтения не завершена (успешно или с ошибкой), другие запросы DPV1 отклоняются.

Для операций DPV1C1 дополнительно требуется, чтобы Ведомое устройство находилось в состоянии обмена данными. Поэтому связь по DPV1C1 возможна только с теми Ведомыми устройствами, которые сконфигурированы Ведущим устройством, у которых функция DPV1 разрешена в параметрирующей телеграмме и которые находятся в состоянии обмена данными.

Связь по DPV1C2 может осуществляться с любым Ведомым устройством на шине, поддерживающим DPV1C2, даже если это устройство не было сконфигурировано в данном шлюзе.

9.5.1 Назначение телеграмм запроса и ответа DPV1C1 (чтение и запись)

Шлюз PBM-ETH-3.0 поддерживает запросы DPV1C1 через область запроса DPV1 (область Modbus 4x). Форматы запросов и возможных ответов приведены в таблице ниже.

Таблица 9-12. Телеграммы запроса и ответа DPV1C1 (чтение и запись).

Тип операции	Телеграмма Запроса	Возможные коды исключений Modbus, возвращаемые на телеграмму запроса	Телеграмма Ответа
Read (Чтение данных из указанного слота/индекса Ведомого устройства DPV1C1)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5E): SLOT • INDEX:LENGTH	MBSRET_V1_FAIL — ошибка выполнения запроса DPV1. MBSRET_V1_BUSY — Главный шлюз занят обработкой запроса DPV1. MBSRET_V1_NOTSUPP — запрашиваемое Ведомое устройство не поддерживает DPV1 или функция DPV1 не разрешена.	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5E):SLOT • INDEX:LENGTH • DATA (Поле DATA дополняется нулями до чётного количества байт.)
			• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0xDE): ERR_DE CODE • ERR_CODE1:ERR_CODE2 • REQ_RESULT (2 байта)
Write (Запись данных в указанный слот/индекс Ведомого устройства DPV1C1)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5F): SLOT • INDEX:LENGTH • DATA (Если длина поля DATA составляет нечётное количество байт, последний байт размещается в старшем байте слова Modbus; значение младшего байта произвольное.)		• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5F): SLOT • INDEX:LENGTH
			• OPTIONID • SLAVE_ADDR: RSV • FUNC_NUM(0xDF): ERR_DEC ODE • ERR_CODE1:ERR_CODE2 • REQ_RESULT (2 байта)

Значения отдельных полей приведены ниже.

Таблица 6-13. Назначение полей DPV1C1.

Наименование поля	Описание поля
OPTIONID	Идентификатор (ID) операции запроса DPV1. Чтобы различать отдельные операции DPV1, рекомендуется увеличивать значение этого поля на 1 после каждой операции DPV1. Начальное значение: 0. В ответной телеграмме DPV1 значение этого поля совпадает со значением, указанным в телеграмме запроса.
RSV	Зарезервированное поле (Reserved)
SLAVE_ADDR	Адрес Ведомого устройства для операции DPV1
DPV1_CLASS	Тип операции DPV1: 1 — операция DPV1C1.
FUNC_NUM	Номер функции операции DPV1: 0x5E — чтение DPV1C1; 0x5F — запись DPV1C1. Если в ответной телеграмме это поле равно 0, значит ответ DPV1C1 от Ведомого устройства ещё не поступил.
SLOT	Номер слота Ведомого устройства для операции DPV1.
INDEX	Номер индекса Ведомого устройства для операции DPV1.
LENGTH	Длина данных операции DPV1, в байтах.
DATA	Поле данных операции DPV1.
ERR_DECODE ERR_CODE1 ERR_CODE2	Код ошибки операции DPV1
REQ_RESULT	Результат ошибки запроса DPV1, возвращаемый Главным шлюзом. 0x0000 — ошибка не обнаружена. 0x0001 — ответ на данный запрос DPV1 не получен по истечении времени ожидания (опросный пакет poll имеет ответ, но ответ на запрос DPV1 не получен). 0x0002 — истекло время ожидания ответа на телеграмму DPV1 (нет ответа на запрос или poll-пакет). 0x0003 — ошибка кадра ответа DPV1. 0x0004 — соединение DPV1C2 закрыто. 0x0005 — ошибка связи DPV1C1: Ведомое устройство не находится в состоянии обмена данными. 0x0006 — Ведущее устройство не находится в состоянии RUN. 0x0007 — Ведущее устройство не является активным PROFIBUS-Master.

9.5.2 Примеры телеграмм запроса и ответа DPV1C1 (чтение и запись)

Ниже приведены примеры телеграмм запроса (чтение и запись) и ответа при работе с DPV1C1 через область данных Modbus.

Таблица 9-14. Примеры телеграмм запроса и ответа DPV1C1 (чтение и запись)

Наименование телеграммы	Формат телеграммы
Операция чтения DPV1C1 (240 байт) из слота 0, индекса 1 Ведомого устройства с адресом 7	
Запрос чтения DPV1C1	00 00 00 00 00 0F 00 10 20 00 00 04 08 <u>00 01 07 00 5E 00 01 F0</u>
Операция записи DPV1C1 (30 байт) в слот 1, индекс 2 Ведомого устройства с адресом 7	
Запрос записи DPV1C1	00 00 00 00 00 2D 00 10 20 00 00 13 26 <u>00 01 07 00 5F 01 02 1E 01 01 00 00 CD CC 8C 3F 00</u> <u>00 B0 40 E6 00 00 00 00 00 C8 42 E6 00 00 05 00 00 BE 42 B8 0B</u>
Чтение 32 байт из области ответных данных DPV1C1 (Modbus 3x)	
Получение ответа DPV1C1	00 00 00 00 00 06 00 04 20 00 00 10

9.5.3 Назначение телеграмм запроса и ответа DPV1C2 (чтение и запись)

Шлюз PBM-ETH поддерживает запросы DPV1C2 через область запроса DPV1 (Modbus 4x). Форматы запросов и возможных ответов приведены в таблице ниже.

Таблица 9-15. Телеграмма инициализации (Initiate) Ведомого устройства DPV1C2

Тип телеграммы	Формат телеграммы	Формат возможной ответной телеграммы
Initiate (подключение к Ведомому устройству DPV1C2)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x57):RSV1 • RSV2:RSV3 • RSV4: RSV5 • FEATURE_SUPT1: FEATURE_SUPT 2 • PROFILE_FEATURE_SUPT1: PROFILE_FEATURE_SUPT2 • PROFILE_ID_NUM (2 байта) • S_TYPE:SLEN • D_TYPE:DLEN • S_API:S_SCL • S_NETADDR:S_MACADDR • D_API:D_SCL • D_NETADDR:D_MACADDR	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x57):MAX_LEN_DUNT • FEATURE_SUPT1:FEATURE_SUPT2 • PROFILE_FEATURE_SUPT1: PROFILE_FEATURE_SUPT2 • PROFILE_ID_NUM (2 байта) • S_TYPE:SLEN • D_TYPE:DLEN • S_API:S_SCL • S_NETADDR:S_MACADDR • D_API:D_SCL • D_NETADDR:D_MACADDR
		• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0xD7):ERR_DECODE • ERR_CODE1:ERR_CODE2 • REQ_RESULT (2 байта) • OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0xD8):LOCAL_GENERATE • SUBNET:INSTANCE_REASON_CODE • ABORT_DETAREQ_RESULT (2 байта)

Таблица 9-16. Телеграмма чтения (Read) Ведомого устройства DPV1C2

Тип телеграммы	Формат телеграммы	Формат возможной ответной телеграммы
Read (Чтение из указанного слота/индекса Ведомого устройства DPV1C2)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5E):SLOT • INDEX:LENGTH	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5E):SLOT • INDEX:LENGTH • DATA (Поле DATA дополняется нулями до чётного количества байт)
		• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0xDE):ERR_DECODE • ERR_CODE1:ERR_CODE2 • REQ_RESULT (2 байта)
Write (Запись в указанный слот/индекс Ведомого устройства DPV1C2)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5F):SLOT • INDEX:LENGTH • DATA (Поле DATA дополняется нулями до чётного количества байт)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x5F):SLOT • INDEX:LENGTH
		• OPTIONID (2 байта) • RSV1: RSV • FUNC_NUM(0xDF):ERR_DECODE • ERR_CODE1:ERR_CODE2 • REQ_RESULT (2 байта)
Trans (Передача данных в указанный слот/индекс Ведомого устройства DPV1C2)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x51):SLOT • INDEX:LENGTH • DATA (Поле DATA дополняется нулями до чётного количества байт)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x51):SLOT • INDEX:LENGTH • DATA (Поле DATA дополняется нулями до чётного количества байт)
		• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0xD1):ERR_DECODE • ERR_CODE1:ERR_CODE2 • REQ_RESULT (2 байта)
Abort (Закрытие соединения с Ведомым устройством DPV1C2)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x58):SUBNET(0x00) • INSTANCE_REASON_CODE(0x20): 0x00	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x58/0xD8):LOCAL_GENERATE • SUBNET:INSTANCE_REASON_CODE • ABORT_DETAIL • REQ_RESULT (2 байта)
Reset (Сброс конечного автомата Ведущего устройства DPV1C2)	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x61): 0x00 • 0x00:0x00	• OPTIONID (2 байта) • SLAVE_ADDR: RSV • FUNC_NUM(0x0x61):0x00 • REQ_RESULT (2 байта)

Значения полей в основном совпадают с DPV1C1. Отличающиеся поля приведены ниже.

Таблица 9-17. Наименования полей DPV1C1

Наименование поля	Описание поля
RSV1~5	Зарезервированное поле, значение произвольное.
FUNC_NUM	Номер функции операции DPV1: 0x57 — установка соединения DPV1C2 0x51 — передача данных DPV1C2 0x58 — закрытие соединения DPV1C2 0x61 — сброс конечного автомата Ведущего устройства DPV1C2 Если в ответной телеграмме это поле равно 0, значит ответ DPV1C2 от Ведомого устройства ещё не поступил.
SADDR	Адрес целевого Ведомого устройства для операции DPV1.
DPV1_CLASS	Тип операции DPV1: 2 — операция DPV1C2.
LOCAL_GENERATE	Если Ведущее устройство обнаружило собственную ошибку, в это поле записывается 1; в противном случае — 0.
ABORT_DETAIL	Ненулевое значение — время Send Timeout из ответной телеграммы RM Ведомого устройства C2.
.....	

9.5.4 Примеры телеграмм запроса и ответа DPV1C2 (чтение и запись)

Ниже приведены примеры телеграмм Запроса (чтение и запись) и Ответа для Ведомого устройства с сетевым адресом «7» при работе с DPV1C2 через область данных Modbus.

Таблица 9-18. Примеры телеграмм запроса и ответа DPV1C2 (чтение и запись)

Наименование телеграммы	Формат телеграммы
Запрос соединения DPV1C2 (Initiate)	00 00 00 00 00 23 00 10 20 00 00 0E 1C 00 01 07 00 57 00 00 00 00 00 00 00 00 00 02 00 02 00 00 00 00 00 00 00
Запрос чтения DPV1C2	00 00 00 00 00 0F 00 10 20 00 00 04 08 00 01 07 00 5E 00 01 F0
Запрос записи DPV1C2	00 00 00 00 00 2D 00 10 20 00 00 13 26 00 01 07 00 5F 01 02 1E 01 01 00 00 CD CC 8C 3F 00 00 B0 40 E6 00 00 00 00 00 C8 42 E6 00 00 05 00 00 BE 42 B8 0B
Запрос передачи данных DPV1C2	00 00 00 00 00 2D 00 10 20 00 00 13 26 00 01 07 00 51 04 01 1E 01 01 00 00 CD CC 8C 3F 00 00 B0 40 E6 00 00 00 00 00 C8 42 E6 00 00 05 00 00 BE 42 B8 0B
Запрос закрытия соединения DPV1C2 (Abort)	00 00 00 00 00 0F 00 10 20 00 00 04 08 00 01 07 00 58 00 20 00
Запрос сброса DPV1C2 (Reset)	00 00 00 00 00 0F 00 10 20 00 00 04 08 00 01 07 00 61 00 00 00
Получение ответа DPV1C2	00 00 00 00 00 06 00 04 20 00 00 10

9.5.5 Асинхронные события DPV1C2

В процессе работы конечного автомата DPV1C2 Ведущее устройство может обнаружить нарушения обмена данными. Шлюз сообщает о них через асинхронные события, помещая данные в область ответа DPV1 (Modbus 3х). Формат асинхронного события приведён в таблице ниже.

Таблица 9-19. Формат асинхронного события DPV1C2

Наименование асинхронного события	Описание поля
Reject	• OPTIONID(0x0000) (2 байта) • SLAVE_ADDR:RSV • FUNC_NUM(0xF1):REASON_CODE • REQ_RESULT (2 байта)
Abort	• OPTIONID (2 байта) • SLAVE_ADDR:RSV • FUNC_NUM(0x58/0xD8):LOCAL_GENERATE • SUBNET:INSTANCE_REASON_CODE • ABORT_DETAIL • REQ_RESULT (2 байта)

Значения поля REASON_CODE в асинхронном событии Reject приведены в таблице ниже.

Таблица 9-20. Значения поля REASON_CODE

Наименование поля	Значение поля	Описание поля
REJ_LE	0x09	Переполнение максимальной длины блока данных (Max_Len_Data_Unit overflow)
REJ_PS	0x0A	Превышено количество одновременных запросов на обслуживание (Number of parallel service requests exceeded)

Значения поля SUBNET в асинхронном событии Abort приведены в таблице ниже.

Таблица 9-21. Значения поля SUBNET

Наименование поля	Значение поля	Описание поля
SUBNET_NO	0	Подсеть отсутствует (No Subnet)
SUBNET_LOCAL	1	Локальная подсеть (Local Subnet)
SUBNET_REMOTE	2	Удалённая подсеть (Remote Subnet)
SUBNET_FALSE	3	Ошибка информации о подсети (Subnet Information Error)

Поле INSTANCE_REASON_CODE возвращает асинхронное событие, обнаруженное шлюзом.

Значения поля INSTANCE_REASON_CODE приведены в таблице ниже.

Таблица 9-22. Значения поля INSTANCE_REASON_CODE

Наименование поля	Значение поля	Описание поля
Биты 7–4: источник события	INSTANCE_DLL 0x10	Уровень канала передачи данных (Data Link Layer, DLL)
	INSTANCE_MS2 0x20	Уровень Ведущего устройства DPV1C2 (MS2)
	INSTANCE_USER 0x30	Прикладной уровень (User Layer)
Биты 3–0: код события при источнике ошибки INSTANCE_DLL (биты 7–4 = INSTANCE_DLL)	UE 1	Ошибка интерфейса Remote-DMPM / DL (Data Link)
	RR 2	Ресурсы удалённого объекта DL (Data Link) недостаточны или недоступны (Resources of the remote-DL Entity not sufficient or not available)
	RS 3	Услуга или удалённый адрес на удалённой точке доступа (DLSAP) не активированы (Service or remote-address at remote-DLSAP or remote-DLSAP not activated): – удалённая станция не является станцией DP (remote-station is no DP-Station) – удалённая станция ещё не готова к выполнению данного сервиса (remote-station is not yet ready for this Service) – удалённая станция связана с другим запрашивающим устройством (remote-station is associated with another Requester) – дополнительный сервис недоступен (optional service not available)
	NR 9	Нет ответных данных (No Response data)
	DH 10	Положительное подтверждение отправленных данных, доступны ответные данные с высоким приоритетом (Positive acknowledgement for sent data, reply data with high priority available)
	RDL 12	Ответные данные с низким приоритетом, ресурсы для отправленных данных отсутствуют (Response data low and no resource for sent data)
	RDH 13	Ответные данные с высоким приоритетом, ресурсы для отправленных данных отсутствуют (Response data high and no resource for sent data)
	NA 15	Отрицательное подтверждение, нет реакции от удалённой станции (Negative ack, no reaction from remote station)
Биты 3–0: код ошибки при источнике ошибки INSTANCE_MS2 (биты 7–4 = INSTANCE_MS2)	ABT_SE(=1)	Ошибка последовательности (Sequence error)
	ABT_FE(=2)	Принят недопустимый запрос APDU (Invalid request APDU received)
	ABT_TO(=3)	Истекло время ожидания соединения (Connection timed out)
	ABT_RE(=4)	Принят недопустимый ответ APDU (Invalid response APDU received)
	ABT_IV(=5)	Недопустимый сервис от пользователя (Invalid service from the User)
	ABT_STO(=6)	Запрошенное значение Send_Timeout слишком мало (Requested value of Send_Timeout was too short)
	ABT_IA(=7)	Недопустимая дополнительная адресная информация (Invalid additional address information)
	ABT_OC(=8)	Истекло время S-таймера, ответный APDU ещё не отправлен (S-Timer expired, response APDU has not been sent yet)

9.5.6 Использование DPV1C2 с помощью DTM шлюза

Функции DPV1C2, предоставляемые интерфейсом Modbus шлюза, спроектированы в первую очередь для совместной работы с DTM. Предварительная настройка через PBConfi не требуется — все параметры задаются через DTM. DTM использует функции DPV1C2 через область данных DPV1 интерфейса Modbus шлюза. (См. Главу 10. «Пример организации связи с устройствами PROFIBUS PA на основе концепции FDT/DTM»).

9.6 Работа с данными диагностики устройств PROFIBUS Slave через область интерфейса Modbus

9.6.1 Структура диагностических данных Ведомого устройства в Modbus 3x

Пользователь получает диагностические данные всех сконфигурированных Ведомых устройств PROFIBUS и метки времени их возникновения путём чтения области диагностических данных Modbus 3x. В поле Unit Identifier телеграммы Modbus (седьмой байт телеграммы Modbus-TCP) необходимо указать адрес Ведомого устройства, диагностические данные которого запрашиваются. **При чтении диагностических данных начальный адрес должен совпадать с начальным адресом области диагностических данных PROFIBUS DP — 0x4000.** Структура диагностических данных Ведомого устройства в соответствующей области приведена в таблице ниже.

Таблица 9-23. Структура диагностических данных Ведомого устройства PROFIBUS

Относительное смещение адреса (в словах)	Назначение данных
0x00	Год метки времени диагностических данных
0x01	Месяц и день метки времени диагностических данных: первый байт — месяц, второй байт — день.
0x02	Часы и минуты метки времени диагностических данных: первый байт — часы, второй байт — минуты.
0x03	Миллисекунды метки времени диагностических данных
0x04	Фактическая длина диагностических данных, в байтах
0x05	Первое слово фактических диагностических данных
.....	
	Последнее слово фактических диагностических данных: если данные занимают один байт, он размещается в старшем байте; младший байт равен нулю.

В структуре диагностических данных Ведомого устройства PROFIBUS метка времени диагностических данных показывает момент получения Ведущим устройством ответной диагностической телеграммы от Ведомого устройства. Поскольку фактическая длина диагностических данных, возвращаемых Ведомым устройством, может быть меньше максимальной длины диагностических данных, поле «Фактическая длина диагностических данных» отражает объём действительных диагностических данных в структуре. **Если пользователь запрашивает длину, превышающую фактическую длину диагностических данных, возвращённых Ведомым устройством, данные за пределами фактической длины считаются недействительными.**

9.6.2 Работа с областью системного журнала через Modbus-интерфейс шлюза

Для упрощения анализа неисправностей системы PROFIBUS шлюз PBM-ETH-3.0 поддерживает функцию системного журнала, которая позволяет регистрировать ключевые события в процессе работы системы. Пользователь может прочитать события журнала через область данных системного журнала Modbus 3x. **При чтении данных системного журнала начальный адрес должен совпадать с начальным адресом области системного журнала — 0x5000.**

События системного журнала делятся на критические и некритические. Критические события шлюз PBM-ETH-3.0 записывает в энергонезависимую память — они сохраняются после отключения питания, что упрощает обслуживание устройства.

9.6.2.1 Формат системного журнала

Область системного журнала Modbus 3x имеет размер 3 КБ, каждая запись журнала занимает 10 байт. Максимальное количество хранимых записей — 256. В энергонезависимой памяти может храниться до 256 критических событий (с сохранением при отключении питания). Формат одной записи журнала приведён ниже.

Таблица 9-24. Формат записи системного журнала

Относительное смещение адреса (в словах)	Назначение данных
0x00	Год метки времени записи журнала
0x01	Месяц и день метки времени записи журнала: в телеграмме Modbus первый байт — месяц, второй байт — день.
0x02	Часы и минуты метки времени записи журнала: в телеграмме Modbus первый байт — часы, второй байт — минуты.
0x03	Миллисекунды метки времени записи журнала: в телеграмме Modbus старший байт миллисекунд — первый, младший байт — второй.
0x04	Идентификатор (ID) записи журнала: в телеграмме Modbus старший байт ID — первый.

Метка времени записи журнала занимает 8 байт, формат: «год:месяц-день:часы-минуты:миллисекунды». Показывает момент возникновения зарегистрированного системного события. Отсчёт времени ведётся по внутренним часам шлюза PBM-ETH-3.0.

Если требуется регистрировать системные события с привязкой к абсолютному времени, пользователь должен задавать системное время в шлюзе PBM-ETH-3.0 при каждом включении питания через регистры системного времени Modbus 4x.

Идентификатор (ID) записи журнала указывает тип зарегистрированного события и занимает 2 байта. Каждому типу события присвоен уникальный ID. Количество записей, находящихся в данный момент в области системного журнала, можно получить через регистр «Количество записей в области системного журнала».

9.6.2.2 Идентификаторы событий системного журнала (Log ID)

Поддерживаемые шлюзом PBM-ETH-3.0 в настоящее время идентификаторы системных событий приведены в таблице ниже.

Таблица 6-25. Идентификаторы событий системного журнала (Log ID)

Идентификатор события журнала (Log ID)	Код события	Описание события
SYSLOG_SYS_PWRUP	0x0001	Включение питания устройства (Device Power-On)
SYSLOG_SYS_PBMWTGRESET	0x0002	Перезапуск ЦПУ Ведущего устройства DP по сторожевому таймеру (DP Master CPU Watchdog Reset)
SYSLOG_SYS_ETHWTGRESET	0x0003	Перезапуск сетевого ЦПУ по сторожевому таймеру (Network CPU Watchdog Reset)
SYSLOG_SYS_PBMCPUEXCEPT	0x0006	Аварийный сброс ЦПУ Ведущего устройства (Master CPU Exception Reset)
SYSLOG_SYS_ETHCPUEXCEPT	0x0007	Аварийный сброс сетевого ЦПУ (Network CPU Exception Reset)
SYSLOG_SYS_TEMPUPLIMIT	0x0008	Превышение верхнего предела рабочей температуры устройства (Device Ambient Temperature Upper Limit Exceeded)
SYSLOG_SYS_TEMPLOWLIMIT	0x0009	Выход за нижний предел рабочей температуры устройства (Device Ambient Temperature Lower Limit Exceeded)
SYSLOG_SYS_VOLTUPLIMIT	0x000A	Превышение верхнего предела напряжения питания устройства (Device Supply Voltage Upper Limit Exceeded)
SYSLOG_SYS_VOLTLOWLIMIT	0x000B	Выход за нижний предел напряжения питания устройства (Device Supply Voltage Lower Limit Exceeded)
SYSLOG_SYS_ETH1IPCRASH	0x000E	Конфликт IP-адресов сетевого интерфейса 1 (Network Interface 1 IP Address Conflict)
SYSLOG_SYS_ETH2IPCRASH	0x000F	Конфликт IP-адресов сетевого интерфейса 2 (Network Interface 2 IP Address Conflict)
SYSLOG_SYS_USRLOADCONFOK	0x0010	Загрузка конфигурации пользователем выполнена успешно (User Configuration Download Success)
SYSLOG_SYS_USRLOADCONFERR	0x0011	Ошибка загрузки конфигурации пользователем (User Configuration Download Failed)
SYSLOG_SYS_FIRMWAREUPDATEOK	0x0012	Обновление встроенного ПО устройства выполнено успешно (Device Firmware Upgrade Success)
SYSLOG_SYS_FIRMWAREUPDATEERR	0x0013	Ошибка обновления встроенного ПО устройства (Device Firmware Upgrade Failed)
Все перечисленные выше события относятся к критическим. При возникновении они записываются в энергонезависимую память и сохраняются после отключения питания.		
SYSLOG_SYS_ETH1CONNECOFF	0x0021	Разрыв соединения сетевого интерфейса 1 (Network Interface 1 Link Down)
SYSLOG_SYS_ETH2CONNECOFF	0x0022	Разрыв соединения сетевого интерфейса 2 (Network Interface 2 Link Down)
SYSLOG_MASTER_P2IDLE	0x0040	Переключение резервированного режима работы устройства: из активного состояния (Primary) в состояние ожидания (Idle) (Device Redundancy State Switch from Primary to Idle)
SYSLOG_MASTER_B2IDLE	0x0041	Переключение резервированного режима работы устройства: из резервного состояния (Backup) в состояние ожидания (Idle) (Device Redundancy State Switch from Backup to Idle)

SYSLOG_MASTER_P2B	0x0042	Переключение резервированного режима работы устройства: из активного состояния (Primary) в резервное (Backup) (Device Redundancy State Switch from Primary to Backup)
SYSLOG_MASTER_B2P	0x0043	Переключение резервированного режима работы устройства: из резервного состояния (Backup) в активное (Primary) (Device Redundancy State Switch from Backup to Primary)
SYSLOG_MASTER_IDLE2B	0x0044	Переключение резервированного режима работы устройства: из состояния ожидания (Idle) в резервное (Backup) (Device Redundancy State Switch from Idle to Backup)
SYSLOG_MASTER_STOP2RUN	0x0045	Переключение рабочего состояния устройства: из состояния STOP в состояние RUN (Device Operating State Switch from STOP to RUN)
SYSLOG_MASTER_RUN2STOP	0x0046	Переключение рабочего состояния устройства: из состояния RUN в состояние STOP (Device Operating State Switch from RUN to STOP)
SYSLOG_SLAVE0_LEAVDEXCH	0x0060	Ведомое устройство с адресом 0 вышло из состояния обмена данными (Slave 0 Exited Data Exchange State)
.....		
SYSLOG_SLAVE125_LEAVDEXCH	0x00DD	Ведомое устройство с адресом 125 вышло из состояния обмена данными (Slave 125 Exited Data Exchange State)
SYSLOG_SLAVE0_HALARM	0x00DE	Ведомое устройство с адресом 0 выдало аварийное сообщение высокого приоритета (Slave 0 High Priority Alarm)
.....		
SYSLOG_SLAVE125_HALARM	0x015B	Ведомое устройство с адресом 125 выдало аварийное сообщение высокого приоритета (Slave 125 High Priority Alarm)

Чтение области системного журнала не требует использования размера, кратного длине записи журнала; начальный адрес чтения также не обязан совпадать с начальным адресом записи журнала. Если запрашиваемая длина данных превышает общий объем текущих записей журнала, недостающая часть ответа заполняется нулями.

9.7 Подробное описание функционального назначения областей регистров Modbus шлюза PBM-ETH-3.0

Регистр состояния системы — входной регистр, адрес 0x8009 (32777)

Таблица 9-26. Регистр состояния системы

Номер бита	Наименование	Функция
Bit3	TOO_HIGH_TEMP	1 — рабочая температура превышает допустимый предел (overtemperature)
Bit2	TOO_LOW_TEMP	1 — рабочая температура ниже допустимого предела (undertemperature)
Bit1	TOO_HIGH_VLT	1 — напряжение питания превышает допустимый предел (overvoltage)
Bit0	TOO_LOW_VLT	1 — напряжение питания ниже допустимого предела (undervoltage)

Операции с регистром конечного автомата Ведущего устройства — регистр хранения (Holding Register), адрес 0x80B0 (32944)

Значения битов при чтении:

Таблица 9-27. Значения битов при чтении регистра конечного автомата Ведущего устройства

Номер бита	Наименование	Функция
Bit2:0	Состояние работы Ведущего устройства	001 — Ведущее устройство в автономном состоянии (Offline) 010 — Ведущее устройство в состоянии конфигурирования (Configuration) 011 — Ведущее устройство в состоянии STOP 100 — Ведущее устройство в состоянии RUN

Значения битов при записи:

Таблица 9-28. Значения битов при записи в регистр конечного автомата Ведущего устройства

Номер бита	Наименование	Функция
Bit2:0	Управление работой Ведущего устройства	001 — недопустимое значение 010 — недопустимое значение 011 — перевести Ведущее устройство в состояние STOP 100 — перевести Ведущее устройство в состояние RUN

Операции с регистрами состояния Ведомых устройств — входные регистры, адреса 0x805D (32861) ... 0x80DA (32986). Отображают состояние Ведомых устройств с адресами от 0 до 125.

Каждому адресу Ведомого устройства соответствует свой уникальный регистр состояния. Формат регистра приведён ниже.

Таблица 9-29. Регистр состояния Ведомого устройства

Номер бита	Наименование	Функция
Bit15	SLAVE_WITHINCONF	1 — Ведомое устройство присутствует в текущей конфигурации
Bit14	SLAVE_ONLINE	Ведомое устройство в сети (активно, отвечает на телеграммы FDL)
Bit13~12	SLAVE_FSMSTATE	Состояние конечного автомата Ведомого устройства: 00 — состояние диагностики (Diagnosis) 01 — состояние параметризации (Parameterization) 10 — состояние конфигурирования (Configuration) 11 — состояние обмена данными (Data Exchange)
Bit11	WTGIMS_ENABLE	Ведомое устройство использует временной базис сторожевого таймера (Watchdog) 1 мс (вместо стандартных 10 мс)
Bit10	DEXCH_HALARM	Ведомое устройство в состоянии Data Exchange выдало аварийное сообщение высокого приоритета (High Priority Alarm)
Bit9	DIAGRESP_NOTSUPPORT	Диагностический ответ Ведомого устройства: часть сконфигурированных функций не поддерживается (Diagnosis Response: some configured functions not supported)
Bit8	DIAGRESP_STATIONNTRDY	Диагностический ответ Ведомого устройства: устройство не готово (Diagnosis Response: Slave not ready)
Bit7	DIAGRESP_PARAMCONFERR	Диагностический ответ Ведомого устройства: ошибка параметризации или конфигурирования (Diagnosis Response: Parameterization or Configuration Error)
Bit6	DIAGRESP_NEEDPARAM	Диагностический ответ Ведомого устройства: требуется параметризация (Diagnosis Response: Slave needs parameterization)
Bit5	DIAGRESP_EXTERNDIAGN	Диагностический ответ Ведомого устройства: расширенная диагностика активна (Diagnosis Response: Extended Diagnosis valid)
Bit4	DIAGRESP_STATICDIAGN	Диагностический ответ Ведомого устройства: статическая диагностика активна (Diagnosis Response: Static Diagnosis valid)
Bit3	DIAGRESP_SLAVELOCKED	Диагностический ответ Ведомого устройства: устройство заблокировано другим Ведущим устройством (Diagnosis Response: Slave locked by another Master)
Bit2	DIAGRESP_INVALIDMADDR	Диагностический ответ Ведомого устройства: неверный адрес Ведущего устройства (Diagnosis Response: Master Address Error)
Bit1	DIAGRESP_INVALIDSID	Диагностический ответ Ведомого устройства: неверный идентификатор (ID) Ведомого устройства (Diagnosis Response: Slave ID Error)
Bit0	DIAGRESP_LENVERRUN	Диагностический ответ Ведомого устройства: превышение допустимой длины диагностических данных (Diagnosis Response: Diagnosis Length Exceeded)

Операции с регистрами управления системой — регистры хранения (**Holding Registers**)

Функции, предоставляемые в настоящее время регистрами управления системой, приведены в таблице ниже.

Таблица 9-30. Регистры управления системой

Функция регистра управления системой	Способ выполнения операции
Коррекция системного времени 0x8000~0x8003 (32768~32771)	Системное время состоит из четырёх полей: год, месяц-день, часы-минуты, миллисекунды. Каждое поле занимает одно слово. Коррекцию времени можно выполнять для всех полей одновременно или для каждого поля по отдельности. Часы — 24-часовой формат.
Задание имени устройства 0x8010~0x802F (32784~32815)	Область энергонезависимой памяти размером 64 байта (сохраняется при отключении питания). Задаёт имя шлюза в системе.
Задание описания устройства 0x8030~0x80AF (32816~32943)	Область энергонезависимой памяти размером 256 байт (сохраняется при отключении питания). Содержит описание шлюза в системе.

Регистр сброса счётчиков связи Ведущего устройства — регистр хранения (**Holding Register**), адрес 0x80B3 (32947)

Таблица 9-31. Регистр сброса счётчиков связи Ведущего устройства

Номер бита	Наименование	Функция
Bit1	COMCNTCLR_TX	1 — сбросить счётчик ошибок передачи Ведущего устройства
Bit0	COMCNTCLR_RX	1 — сбросить счётчик ошибок приёма Ведущего устройства

Регистр автоматического режима Ведущего устройства — входной регистр, адрес 0x8047 (32839)

Таблица 9-32. Регистр автоматического режима Ведущего устройства

Номер бита	Наименование	Функция
Bit1	AUTO_RUN	1 — при подаче питания шлюз автоматически переходит в состояние RUN
Bit0	AUTO_STOP	1 — включена функция автоматического останова: если сконфигурированное Ведомое устройство выходит из состояния обмена данными, Ведущее устройство автоматически переключается в состояние безопасных выходов

Регистр конфигурации Ведущего устройства — входной регистр, адрес 0x8048 (32840)

Таблица 9-33. Регистр конфигурации Ведущего устройства

Номер бита	Наименование	Функция
Bit3	CONFCTRL_PREPM	1 — шлюз является предпочтительным P-Master (в нерезервированном режиме этот бит всегда равен 1)
Bit2	CONFCTRL_REDUN	1 — режим резервирования двух устройств (Dual Redundancy); 0 — нерезервированный режим

Регистр состояния конфигурации Ведущего устройства — входной регистр, адрес 0x8049 (32841)

Таблица 9-34. Регистр состояния конфигурации Ведущего устройства

Номер бита	Наименование	Функция
Bit3:2	CONFSTAT	00 — конфигурирование Ведущего устройства не завершено 01 — ошибка конфигурирования Ведущего устройства 10 — конфигурирование Ведущего устройства выполнено успешно

Регистр состояния установки адреса Введомого устройства — входной регистр, адрес 0x805C (32860)

Таблица 9-35. Регистр состояния установки адреса Введомого устройства

Номер бита	Наименование	Функция
Bit1:0	SET_SADDR_RSLT	00 — функция установки адреса Введомого устройства не готова 01 — установка адреса Введомого устройства выполнена успешно 10 — истекло время ожидания ответа при установке адреса Введомого устройства 11 — установка адреса Введомого устройства отклонена

Регистр управления синхронизацией и фиксацией (Sync/Freeze) Введомых устройств — регистр хранения (Holding Register), адрес 0x80C0 (32960)

Таблица 9-36. Регистр управления синхронизацией и фиксацией (Sync/Freeze) Введомых устройств

Номер бита	Наименование	Функция
Bit11	SYNC_ENABLE	1 — синхронизация (Sync) разрешена
Bit10	SYNC_DISABLE	1 — синхронизация (Sync) запрещена
Bit9	FREEZE_ENABLE	1 — фиксация (Freeze) разрешена
Bit8	FREEZE_DISABLE	1 — фиксация (Freeze) запрещена
Bit7:0	GROUP_NUM	Номер группы для синхронизации/фиксации (Sync/Freeze)

Регистр сброса счётчиков связи Введомого устройства — регистр хранения (Holding Register), адрес 0x80C1 (32961)

Таблица 9-37. Регистр сброса счётчиков связи Введомого устройства

Номер бита	Наименование	Функция
Bit1	COMCNTCLR_LEAVDEXC	1 — сбросить счётчик выходов Введомого устройства из состояния обмена данными
Bit0	COMCNTCLR_RETRY	1 — сбросить счётчик повторных попыток передачи для Введомого устройства

Глава 10. Пример организации связи с устройствами PROFIBUS PA на основе концепции FDT/DTM

В данной главе на конкретном примере подключения устройств PROFIBUS PA к шлюзу PBM-ETH-3.0 описывается, как с помощью конфигурационного ПО PBConf выполнить настройку сети с устройствами PROFIBUS PA, а также как использовать DTM шлюза и коммуникационный DTM шлюза в среде FDT.

В данном примере используются следующие устройства и программное обеспечение сторонних производителей:

- Используемый DP/PA-соединитель: Siemens DP/PA Coupler (преобразователь физического уровня) (доступен DP/PA Coupler торговой марки i-nets: [DS-DPPA-1.0](#))
- Используемое устройство PROFIBUS PA: ABB PA 2020TG
- Используемая программа FDT-среды: P+F PACTware (V4.1) — открытая среда FDT

10.1 Обзор функций PROFIBUS PA и DTM шлюза PBM-ETH-3.0

10.1.1 Подключение устройств PROFIBUS PA

Шлюз PBM-ETH-3.0 поддерживает обмен данными по протоколу PROFIBUS PA.

При подключении через преобразователь DP/PA (DP/PA Coupler) шлюз PBM-ETH-3.0 может осуществлять стандартный обмен данными DPV0/V1 с устройствами PA.

Способ настройки устройств PROFIBUS PA: устройства PA конфигурируются с помощью ПО PBConf. Порядок настройки описан ниже.

10.1.2 Обмен данными с использованием DTM

Шлюз PBM-ETH-3.0 может работать с подключаемыми устройствами через DTM, включая устройства PROFIBUS PA (требуется наличие преобразователя DP-PA на аппаратном уровне, например, [i-nets DS-DPPA Coupler](#)).

10.1.3 Использование DTM для инициирования обмена данными DPV1

Как уже было указано в технических характеристиках выше, шлюз PBM-ETH-3.0 поддерживает обмен данными DPV1C1/C2. Пользователь инициирует запрос DPV1 путём обращения к области Modbus 4х, начиная с адреса 0х2000.

На практике обмен данными DPV1 через шлюз обычно инициируется одним из следующих способов:

Клиент Modbus TCP (например, управляющий контроллер верхнего уровня) отправляет запрос Modbus TCP. Получив этот запрос, шлюз PBM-ETH-3.0 запускает обмен данными DPV1.

В системе управления устройствами (например, FDT/DTM, EDD) пользователь запускает чтение/запись данных устройства PROFIBUS PA через файл описания/управления устройством. В реальной коммуникационной сети этот запрос преобразуется в запрос чтения/записи Ethernet Клиента Modbus TCP, который направляется шлюзу PBM-ETH-3.0, а тот, в свою очередь, инициирует обмен данными DPV1.

В данном примере используется среда FDT/DTM. Поток данных в системе показан на рисунке ниже.

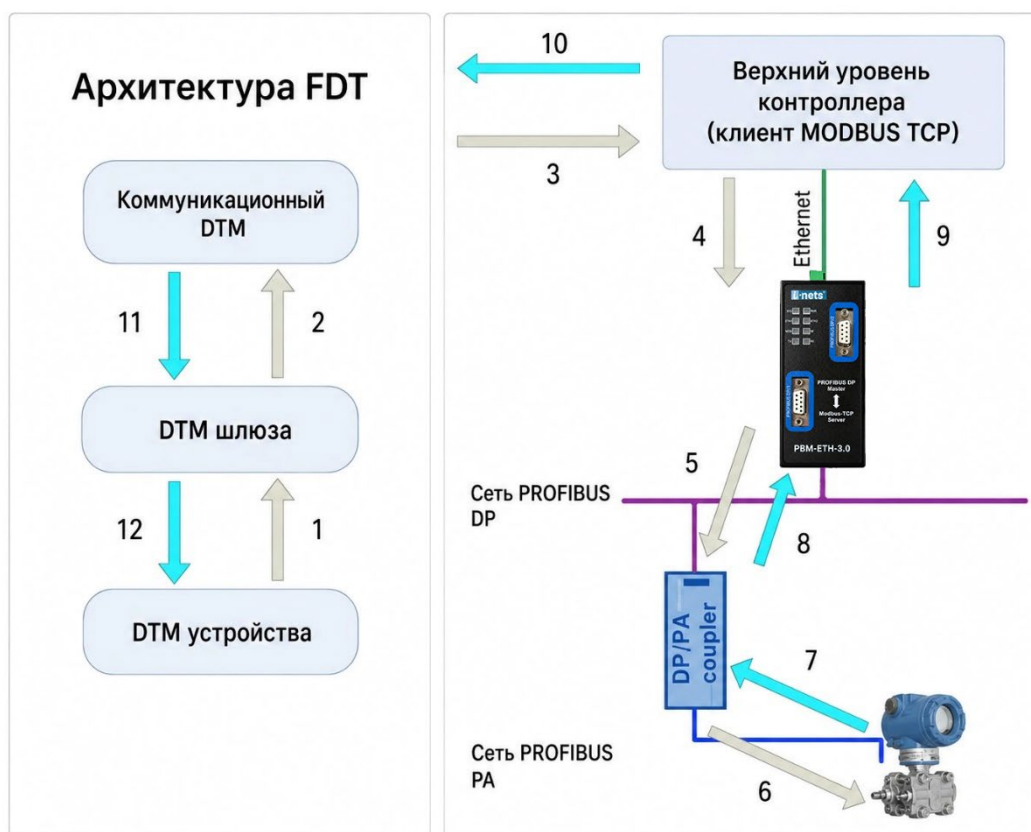


Рис. 10-1. Схема потоков данных в системе FDT/DTM

10.2 Описание системы примера и используемое ПО

В данном примере подключения устройств РА к Главному шлюзу PBM-ETH-3.0 используемое оборудование и программное обеспечение верхнего уровня подробно перечислены в таблице ниже.

Используемое в примере оборудование приведено в таблице ниже.

Таблица 10-1. Перечень аппаратных средств

Оборудование	Кол-во	Производитель	Описание
PBM-ETH-3.0	1	i-nets	Шлюз Ethernet/PROFIBUS DP
DP/PA Coupler	1	Siemens	Преобразователь физического уровня DP/PA
Transmitter 2020TG	1	ABB	Датчик ABB (PROFIBUS PA)
Кабель DP	1	Siemens	
Кабель PA	1	Siemens	
ПК с программным обеспечением			ПК с установленной средой FDT, конфигурационным ПО PBConfi и клиентским ПО Modbus TCP; используется в качестве системы управления верхнего уровня

Используемое ПО и установочные файлы:

Таблица 10-2. Перечень программного обеспечения для ПК

Программное обеспечение	Производитель	Описание
DSDPV1COMMDTMSSetup.exe	i-nets	Коммуникационный DTM шлюза
PBConfi		Конфигурационное ПО шлюза
3KXP000265S0005_DTM_2600T	ABB	DTM устройства РА 2020TG
YP0004C2.GSD	ABB	GSD-файл устройства РА 2020TG
РАСТware (V4.1)	P+F	Открытая среда FDT верхнего уровня

10.3 Настройка устройств PROFIBUS PA

Подключение устройств PROFIBUS PA к шлюзу PBM-ETH-3.0 настраивается с помощью конфигурационного ПО PBConfi. При использовании преобразователя DP/PA от Siemens окно настройки показано на Рис. 10-2 ниже.

Настройка шлюза PBM-ETH-3.0: выберите PBMG-ETH-3 в списке DP_Master и добавьте его в сеть PROFIBUS DP. В списке PA_SLAVE выберите нужное устройство PROFIBUS PA и двойным кликом добавьте его в сеть PROFIBUS DP. Поскольку используется преобразователь только физического уровня, настройка самого DP/PA Coupler не требуется.

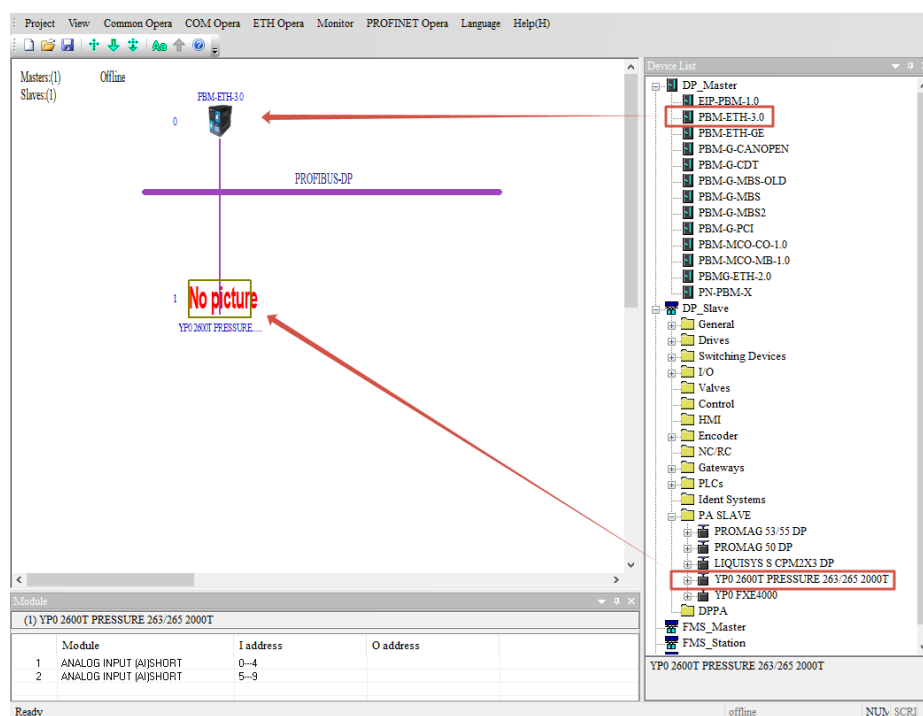


Рис. 10-2. Добавление устройств в сеть PROFIBUS

Поскольку используется преобразователь DP/PA (только физического уровня), установите скорость передачи данных (бодовую скорость) Ведущего устройства равной 45,45 Кбит/с.

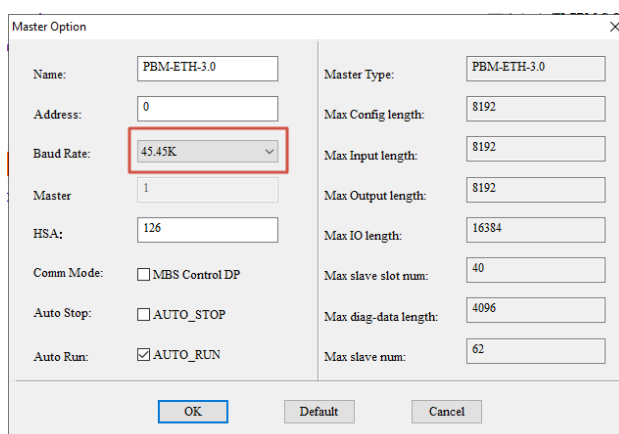


Рис. 10-3. Настройка скорости передачи данных

Загрузите данную конфигурацию в шлюз PBM-ETH-3.0. После этого шлюз сможет обмениваться данными с устройствами PROFIBUS PA, подключёнными аппаратно.

10.4 Применение DTM в обмене данными PROFIBUS PA

В данном примере система FDT/DTM используется в качестве системы управления устройствами верхнего уровня. Обмен данными DPV1 в сети PROFIBUS с устройствами PA инициируется со стороны DTM. В качестве среды FDT используется открытое ПО RACTware.

10.4.1 Коммуникационный DTM шлюза + DTM устройства PA

Шаг 1: запустить установочные EXE-файлы всех DTM и установить их на компьютер.

Шаг 2: открыть среду FDT верхнего уровня (в данном примере — RACTware). Настроить топологию сети в соответствии с реальной коммуникационной сетью.

Как показано на Рис. 10-4 ниже, выделите «Проект», нажмите правую кнопку мыши и в открывшемся меню выберите «Добавить прибор». В появившемся окне устройств, показанном на Рис. 10-5, выберите коммуникационный DTM шлюза «DS DPV1 CommDTM» и добавьте его в проект среды FDT.

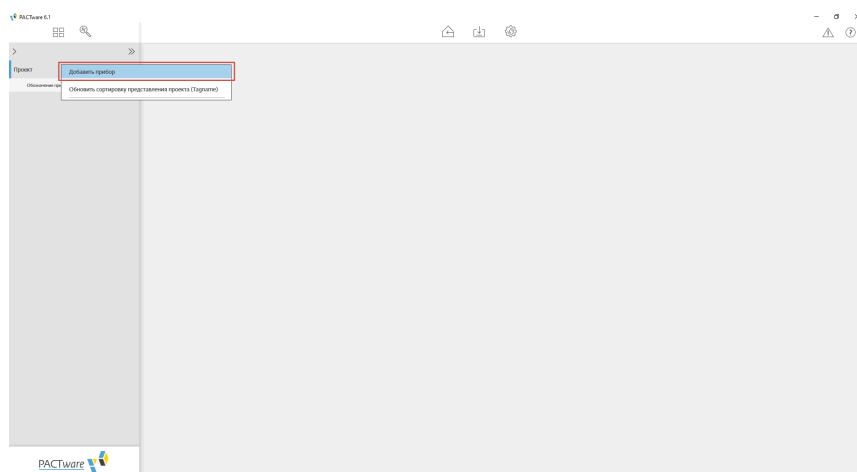


Рис. 10-4. Добавление устройства

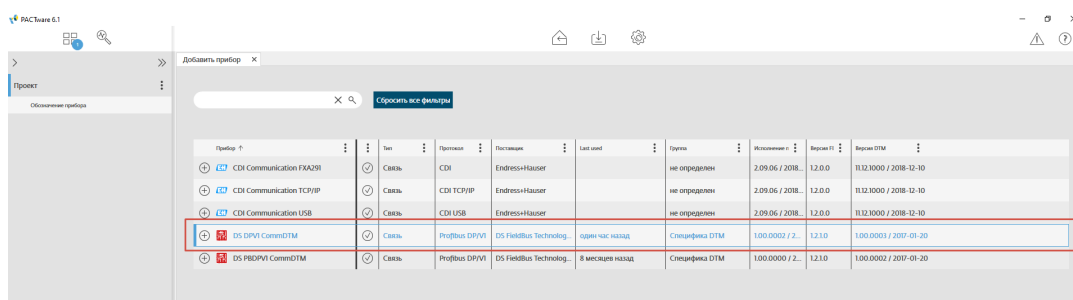


Рис. 10-5. Добавление коммуникационного DTM

Шаг 3: Настройка DTM подчинённого устройства. Выделите сконфигурированный коммуникационный DTM шлюза, нажмите правую кнопку мыши и в открывшемся меню выберите «Добавить прибор». В появившемся окне устройств выберите DTM устройства РА «DTM TG2600-РА» и добавьте его в проект среды FDT. Этот DTM будет размещён как подчинённый по отношению к коммуникационному DTM Главного шлюза.

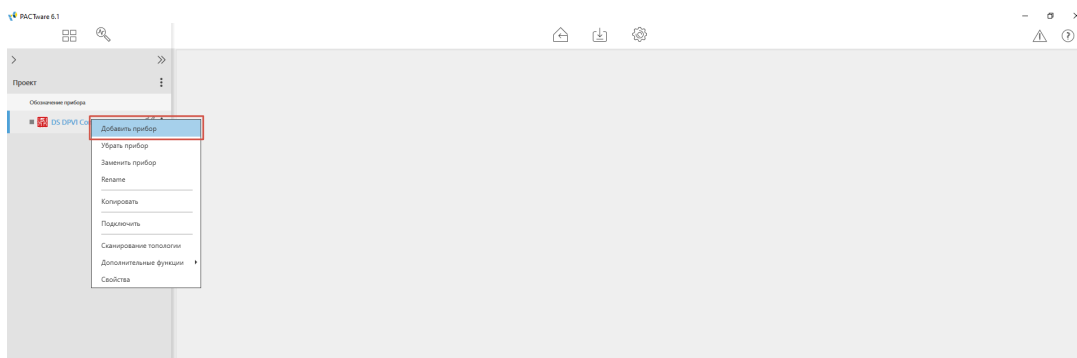


Рис. 10-6. Добавление подчинённого устройства

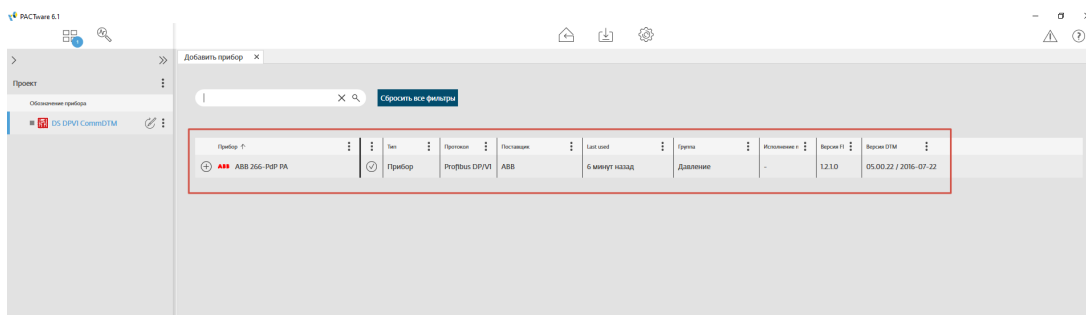


Рис. 10-7. Добавление DTM устройства

Шаг 4: Настройка адресной информации для обмена данными каждого узла.

Настройте адрес устройства PA 2020TG, как показано на Рис. 10-8 ниже: в открывшемся окне настройки адреса с сети PROFIBUS DP укажите фактический адрес обмена данными устройства PA.

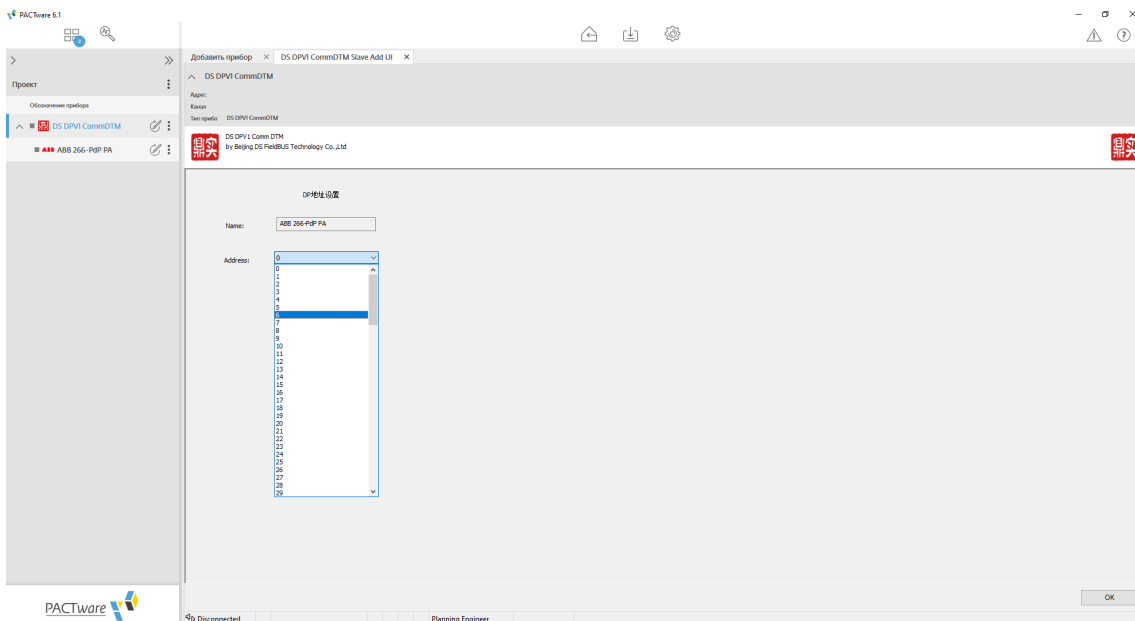


Рис. 10-8. Окно настройки адреса обмена данными

Настройте IP-адрес интерфейса Ethernet шлюза PBM-ETH-3.0: кликните по коммуникационному DTM шлюзу и в открывшемся окне настройки IP-адреса, показанном на Рис. 10-9, измените адрес на используемый в данный момент.

Задав IP-адрес, кликните по кнопке «SAVE».

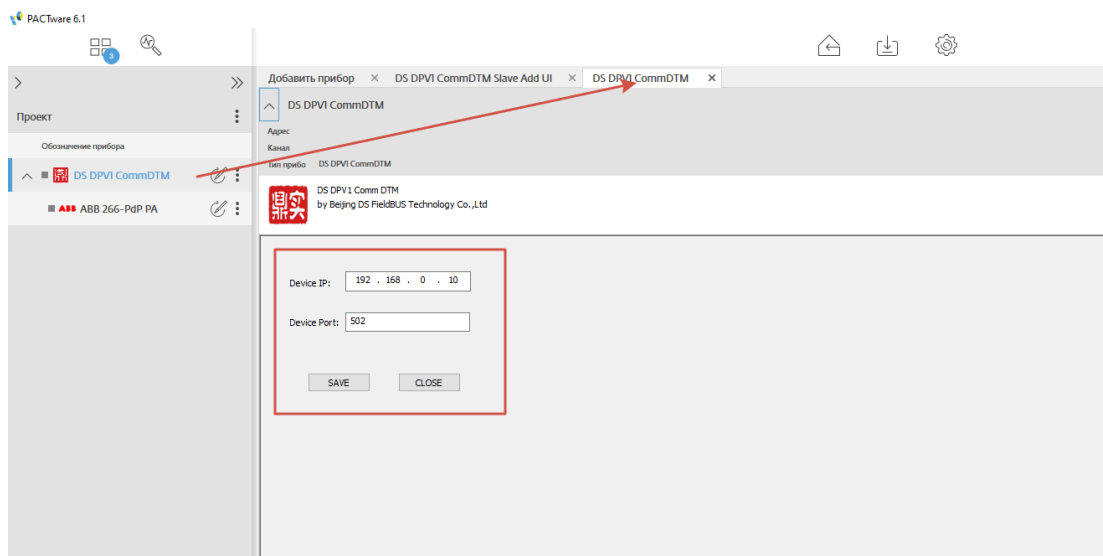


Рис. 8-9. Окно настройки IP-адреса

Итоговый результат настройки показан на Рис. 10-10 ниже:

- DS DPV1 CommDTM — коммуникационный DTM шлюза версии 3.0
- XI XXX — DTM устройства PROFIBUS PA (ABB 2020TG)

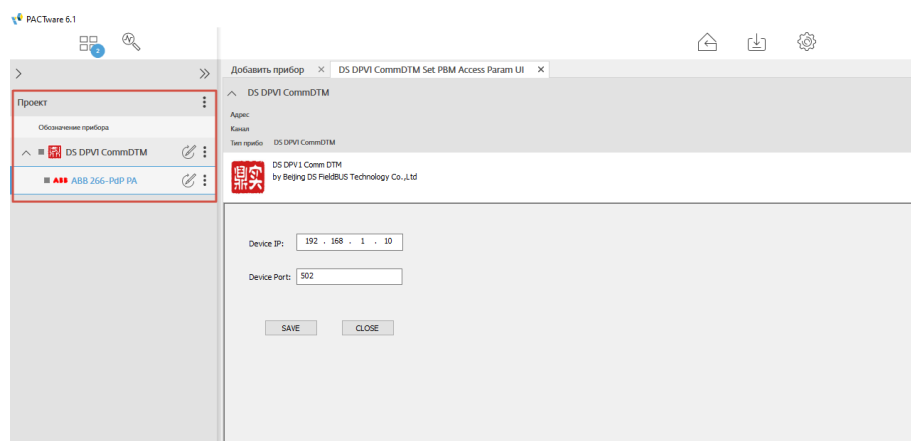


Рис. 10-10. Итоговая конфигурация

Шаг 5: Дважды кликните по DTM устройства PROFIBUS PA, чтобы настроить параметры и инициировать обмен данными DPV1 в сети PROFIBUS. Окно работы с прибором PROFIBUS PA (266 PdP PA) показано на Рис. 10-11 ниже.

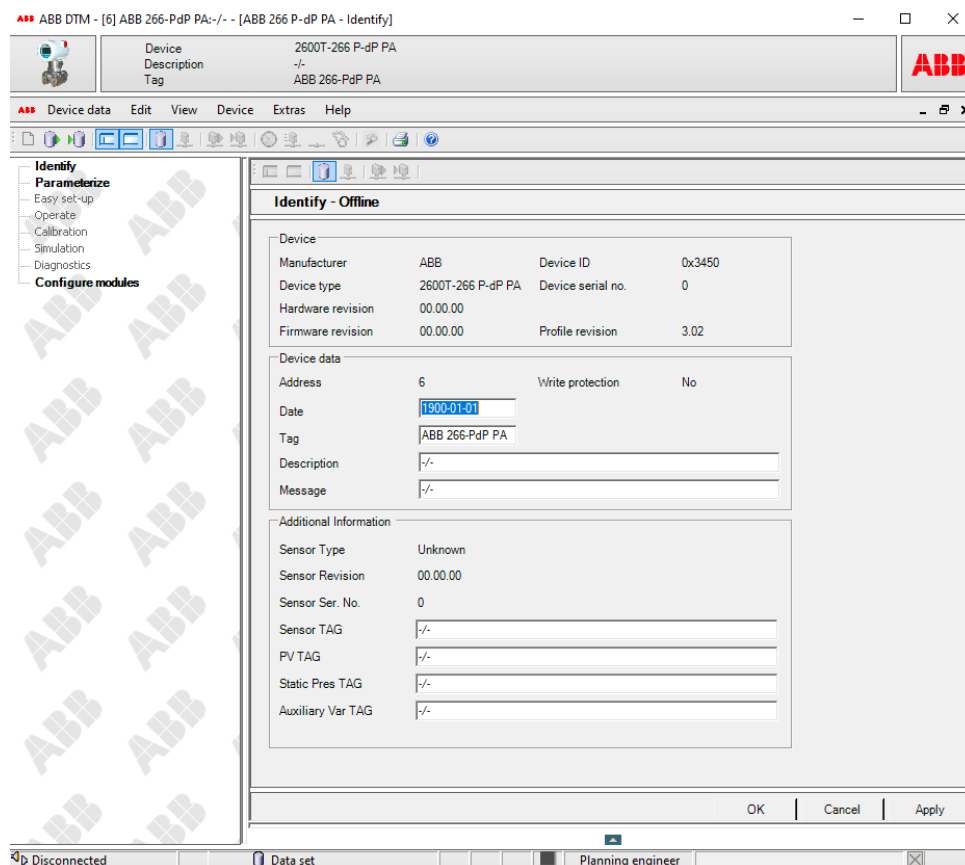


Рис. 10-11. Окно работы с прибором PROFIBUS PA (266 PdP PA)

Приложение 1. Термины и определения

Режим коммутатора с двумя сетевыми портами (Switch Dual-Port Mode):

Режим работы, при котором встроенный коммутационный чип шлюза PBM-ETH обеспечивает передачу данных как для самого шлюза, так и между устройствами, подключёнными к двум его сетевым портам. В этом режиме можно строить сеть по топологии «гирляндная цепочка» (daisy chain) без использования внешнего коммутатора.

Режим независимых сетевых портов (Independent Dual-Port Mode):

Режим работы, при котором два сетевых интерфейса шлюза PBM-ETH имеют собственные IP-адреса, находятся в разных подсетях и не пересылают телеграммы между собой. Режим независимых сетевых портов может использоваться для построения резервированной сети Ethernet. Связь со шлюзом по Modbus-TCP может осуществляться одновременно через IP-адреса обоих сетевых портов.

Нормальный рабочий режим шлюза (Normal Operation Mode):

Режим работы шлюза PBM-ETH, противопоставляемый режиму обновления встроенного ПО. В этом режиме шлюз выполняет свои штатные функции. Обновление встроенного ПО в данном режиме невозможно.

Режим обновления встроенного ПО шлюза (Firmware Upgrade Mode):

Режим работы шлюза, используемый при обновлении встроенного ПО. В этом режиме возможно только обновление прошивки; функции преобразования протоколов Modbus-TCP ↔ PROFIBUS недоступны.

Безопасные выходные данные PROFIBUS (PROFIBUS Safety Output Data):

Данные нулевой длины или данные, состоящие из одних нулей, отправляемые шлюзом Ведомым устройствам PROFIBUS DP.

Состояние «RUN» Ведущего устройства:

Нормальное рабочее состояние Ведущего устройства. В этом состоянии Выходные данные из области Modbus (Modbus 4х) передаются на выходы Ведомых устройств DP, а данные со входов Ведомых устройств DP обновляются во входной области Modbus (Modbus 3х).

Состояние «STOP» Ведущего устройства:

Состояние безопасных выходов Ведущего устройства. В этом состоянии Ведомым устройствам DP отправляются безопасные выходные данные PROFIBUS (данные нулевой длины или все нули) вместо данных из выходной области Modbus. Входные данные от Ведомых устройств DP при этом продолжают обновляться во входной области Modbus.

Состояние OFFLINE (Автономный режим) Ведущего устройства:

Автономное состояние Ведущего устройства, при котором оно не отправляет телеграммы по шине PROFIBUS DP и не принимает никаких телеграмм от шины PROFIBUS DP. Ведущее устройство находится в этом состоянии, если шлюз PBM-ETH не имеет корректной конфигурации или DIP-переключатель установлен в положение «Автономный режим» (OFFLINE).

Автоматический режим работы шлюза:

Режим, в котором шлюз PBM-ETH работает при включённой опции «AUTO RUN» в конфигурации Ведущего устройства (по умолчанию эта опция включена в программе PBConf). При подаче питания Ведущее устройство автоматически переходит в состояние RUN. Если опция «AUTO STOP» включена и выполняется условие её срабатывания, шлюз переходит в состояние STOP; когда условие AUTO STOP перестаёт действовать, шлюз автоматически возвращается в состояние RUN.

Ручной режим работы шлюза (Manual Operation Mode):

Режим, в котором шлюз PBM-ETH работает при отключённой опции «AUTO RUN» в конфигурации Ведущего устройства. При подаче питания Ведущее устройство запускается и останавливается в состоянии STOP; для перехода в состояние RUN требуется ручное переключение пользователем. Если опция «AUTO STOP» включена и срабатывает условие её активации, Ведущее устройство переходит в состояние STOP. При этом, даже когда условие

«AUTO STOP» перестаёт действовать, Ведущее устройство остаётся в состоянии STOP — возврат в RUN должен быть выполнен пользователем вручную. Данный режим обычно применяется в системах с повышенными требованиями к уровню безопасности.

Циклический обмен данными PROFIBUS DPV0:

Периодический (циклический) обмен данными между Ведущим и Вedomыми устройствами PROFIBUS DP.

Входные данные DPV0 (DPV0 Input Data):

Данные, периодически передаваемые Вedomым устройством PROFIBUS DP в шлюз.

Выходные данные DPV0 (DPV0 Output Data):

Данные, периодически отправляемые шлюзом Вedomым устройствам PROFIBUS DP.

Ациклический обмен данными PROFIBUS DPV1:

Ациклический обмен данными между Ведущим устройством и Вedomыми устройствами в сети PROFIBUS DP.

Данные чтения DPV1 (DPV1 Read Data):

Ациклические данные, считываемые шлюзом из Вedomого устройства PROFIBUS DP.

Данные записи DPV1 (DPV1 Write Data):

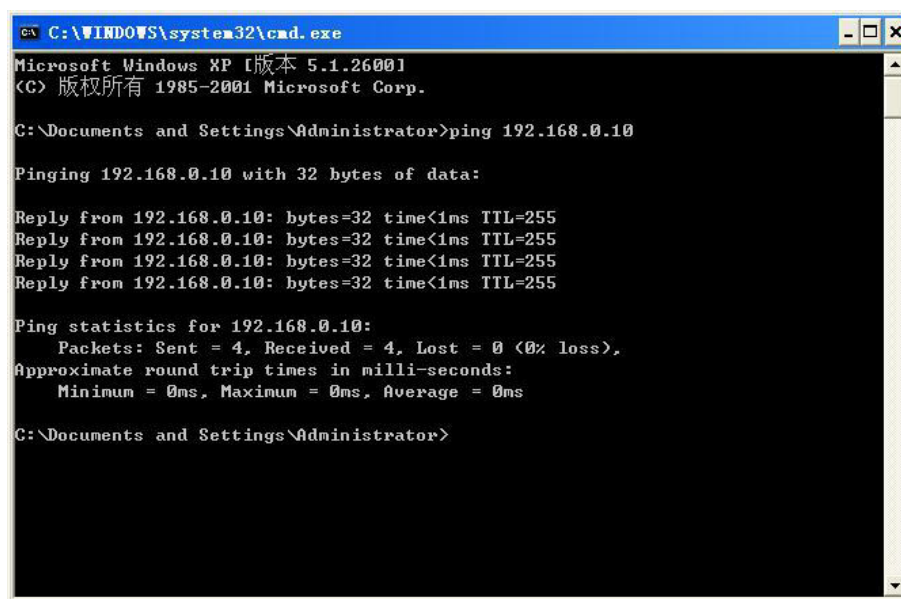
Ациклические данные, отправляемые шлюзом Вedomому устройству PROFIBUS DP.

Приложение 2. Поиск и устранение типовых неисправностей (дополняется)

Сетевые операции

Диагностика связи с устройствами при помощи ICMP ping

РВМ-ЕТН-3.0 поддерживает протокол ICMP. Команда «ping» позволяет проверить сетевую связь со шлюзом. Например, в ОС Windows команда ping выполняется из окна командной строки (cmd). С её помощью определяется, есть ли связь со шлюзом. Если связь отсутствует — проверьте исправность подключения к сети.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:

Reply from 192.168.0.10: bytes=32 time<1ms TTL=255
Reply from 192.168.0.10: bytes=32 time<1ms TTL=255
Reply from 192.168.0.10: bytes=32 time<1ms TTL=255
Reply from 192.168.0.10: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator>
```

Внимание: для защиты от сетевых атак шлюз отклоняет запросы ping, если они отправляются с высокой частотой (с короткими интервалами). В этом случае инструмент ping показывает тайм-аут ответа, что не является признаком нарушения сетевой связи. При диагностике сети с помощью команды «ping» рекомендуется выполнять проверку с интервалом не менее 1 с.

Обнаружение конфликта IP-адресов шлюза

Шлюз поддерживает механизм обнаружения конфликта IP-адресов. При подаче питания шлюз РВМ-ЕТН сначала проверяет, не используется ли уже в текущей сети IP-адрес, который он намерен занять. Если такой адрес уже существует, фиксируется конфликт IP-адресов. При этом индикатор ЕТН соответствующего сетевого порта на передней панели устройства начинает мигать красным цветом; устройство при этом продолжает нормальную работу. Шлюз, зафиксировавший конфликт IP-адресов, повторяет проверку каждые 10 секунд. Если конфликт IP-адресов исчез, индикатор ЕТН переключается на зелёный цвет; в противном случае состояние конфликта IP-адресов сохраняется.

Приложение 3. Краткое описание спецификации обмена данными по Modbus-TCP

Примечание: далее идёт общее описание протокола Modbus. Если вы уже знакомы с данным протоколом, то можно перейти к практическому описанию Modbus в п. 6 на стр. 88

1. Общие сведения

Modbus-TCP — это простой протокол, независимый от производителя, производный от семейства коммуникационных протоколов Modbus, предназначенный для управления и контроля устройств автоматизации. Он охватывает применение телеграмм Modbus в средах Intranet и Интернет, использующих протокол TCP/IP. Наиболее распространённое применение протокола — обслуживание таких устройств, как ПЛК, модули Ввода-Вывода и шлюзы, подключаемые к другим простым полевым шинам или модулям Ввода-Вывода.

Протокол Modbus-TCP опубликован как (фактический) стандарт автоматизации. Поскольку Modbus уже широко известен, данное Приложение содержит лишь небольшой объём информации, не вошедший в другие документы. Тем не менее, настоящий раздел призван прояснить, какие функции Modbus представляют ценность для обеспечения совместимости типовых устройств автоматизации, а какие части Modbus являются «избыточными», унаследованными от применения протокола в качестве программируемого протокола для ПЛК.

1.1 Ориентация на Соединение

В Modbus обработка данных традиционно не устанавливает границ Соединений (логический канал связи, устанавливаемый по протоколу TCP между Клиентом (Client/Master) и Сервером (Server/Slave) Modbus-TCP), что обеспечивает высокую устойчивость к помехам и требует минимального объёма служебной информации с каждой стороны.

Программные операции, напротив, предполагают подход, ориентированный на Соединение. Для простых переменных это реализуется через единый символ «входа», а для переменных Modbus Plus — через явное указание «пути программы», который поддерживает двустороннее Соединение до тех пор, пока оно не будет явно разорвано.

Modbus-TCP поддерживает оба случая. Соединение легко идентифицируется на уровне сетевого протокола, и одно Соединение может поддерживать несколько независимых транзакций. Кроме того, протокол TCP допускает большое количество одновременных Соединений, поэтому во многих случаях инициатор сам решает: переустанавливать Соединение по мере необходимости либо повторно использовать одно длительное Соединение.

Разработчиков, знакомых с Modbus, может удивить, почему протокол TCP, ориентированный на Соединение, применяется гораздо шире, чем UDP, ориентированный на дейтаграммы. Основная причина заключается в том, что отдельные «транзакции» инкапсулируются в Соединение, которое можно идентифицировать, контролировать и разрывать без специальных действий со стороны Клиента и Сервера. Это позволяет процессу адаптироваться к изменению характеристик сети и упрощает внедрение средств безопасности, таких как межсетевые экраны и прокси-серверы.

Аналогичным подходом руководствовались разработчики первоначальной Всемирной паутины, выбрав TCP и порт 80 для реализации минимального веб-запроса, выполняемого как единая транзакция.

1.2 Кодирование данных

Modbus использует порядок байтов «big-endian» для представления адресов и объектов данных. Это означает, что при передаче числового значения, размер которого превышает один байт, первым отправляется старший (наиболее значимый) байт. Примеры:

- 16-битное значение 0x1234 → первым идёт 0x12, затем 0x34
- 32-битное значение 0x12345678L → порядок передачи: 0x12, 0x34, 0x56, 0x78

1.3 Пояснения к справочным номерам

Прокол Modbus строит свою модель данных на основе ряда областей данных с различными характеристиками. Четыре базовые области данных перечислены ниже:

- Область дискретных входов (Discrete Inputs) — однобитовые значения, предоставляемые системой ввода-вывода; только чтение.
- Область дискретных выходов (Coils) — однобитовые значения, изменяемые прикладной программой; чтение и запись.
- Область входных регистров (Input Registers) — 16-битовые числовые значения, предоставляемые системой ввода-вывода; только чтение.
- Область регистров хранения (Holding Registers) — 16-битовые числовые значения, изменяемые прикладной программой; чтение и запись.

Различие между входом и выходом, а также между объектами данных с битовой и регистровой адресацией не подразумевает какого-либо различия в поведении приложения. Вполне обычно и совершенно допустимо считать, что все четыре базовые области данных накладываются друг на друга, если это является наиболее естественной интерпретацией для целевого механизма.

Для каждой базовой области данных протокол позволяет индивидуально обращаться к любому из 65536 объектов данных, а операции чтения и записи могут охватывать несколько последовательных объектов данных вплоть до ограничения размера данных, определяемого функциональным кодом транзакции.

Здесь нет предположения о том, что объекты данных представляют собой действительно непрерывный массив данных — хотя именно так устроено большинство простых ПЛК.

Функциональный код «Read/Write General Reference» определён как переносящий 32-битное ссылочное значение и позволяющий напрямую обращаться к объектам данных в «очень» большом адресном пространстве. В настоящее время устройства ПЛК, использующие эту возможность, отсутствуют.

Потенциальным источником путаницы является соотношение между ссылочными значениями функций Modbus и «номерами регистров» ПЛК Modicon. По историческим причинам пользовательские ссылочные значения представлены десятичными числами, начинающимися с 1. Modbus же использует более общую форму — беззнаковые целые числа, начинающиеся с 0, при анализе и организации данных программным обеспечением.

Таким образом, сообщение Modbus, запрашивающее чтение регистра 0, возвращает значение, хранящееся в приложении по адресу регистра 4:00001 (тип памяти 4 = выходной регистр, ссылочное значение 00001).

1.4 Основной принцип неявной длины

Все запросы и ответы Modbus спроектированы таким образом, чтобы получатель мог удостовериться в целостности сообщения. Для функциональных кодов с запросами и ответами фиксированной длины достаточно отправить только сам функциональный код. Для функциональных кодов, переносящих данные переменной длины в запросе или ответе, перед полем данных добавляется счётчик байтов (один байт).

При передаче Modbus поверх протокола TCP в префикс включается дополнительная информация о длине, что позволяет получателю определять границы сообщения даже при его разбиении на несколько пакетов. Наличие явных и неявных признаков длины, а также использование кода обнаружения ошибок CRC-32 (на уровне Ethernet) сводит к бесконечно малой величине вероятность появления нераспознанных ошибок в запросах и ответах.

2. Обзор классов совместимости (Conformance Classes)

При разработке нового протокола «с нуля» можно обеспечить жёсткое единообразие способов кодирования и интерпретации. Однако Modbus, благодаря своим широким возможностям, уже реализован во множестве устройств, и необходимо избегать нарушения работы существующих реализаций.

По этой причине существующий набор типов обработки был разделён на классы совместимости: класс 0 объединяет общеупотребительные и в целом единообразно реализованные функции; класс 2 включает полезные функции, но обладающие некоторыми особенностями. Функции существующих устройств, не пригодные для обеспечения совместимости, также были идентифицированы.

Следует отметить, что в будущих расширениях данного стандарта будут определены дополнительные функциональные коды для ситуаций, не охваченных существующим стандартом де-факто. Однако приведение подробностей предлагаемых расширений в настоящем документе ввело бы читателя в заблуждение. Всегда имеется возможность определить, поддерживает ли конкретное целевое устройство тот или иной функциональный код, путём «пробной» отправки кода или даже путём анализа типа возвращаемого исключительного ответа. Такой подход гарантирует сохранение совместимости с существующими устройствами Modbus при внедрении подобных расширений в будущем. Фактически именно на этом основан принцип разделения функциональных кодов по классам в настоящее время.

2.1 Функции Класcа 0

Это минимальный набор полезных функций — как для Ведущего (Master), так и для Ведомого (Slave) устройства.

- Чтение нескольких регистров хранения (функциональный код 03)
- Запись нескольких регистров хранения (функциональный код 16)

2.2 Функции Класcа 1

Это дополнительный набор общепринятых и широко используемых функций для совместного применения. Как отмечалось ранее, многие Ведомые устройства обрабатывают входы, выходы, дискретные значения и значения регистров как равнозначные.

- Чтение дискретных выходов (Coils) — функциональный код 01
- Чтение дискретных входов (Discrete Inputs) — функциональный код 02
- Чтение входных регистров (Input Registers) — функциональный код 04
- Запись одного дискретного выхода (Single Coil) — функциональный код 05
- Запись одного регистра хранения (Single Register) — функциональный код 06
- Чтение слова состояния исключений (Exception Status) — функциональный код 07 (данная функция, очевидно, имеет различный смысл для каждой серии Ведомых устройств).

2.3 Функции Класcа 2

Функции передачи данных, необходимые для рутинных операций, таких как человеко-машинный интерфейс (HMI) и управление.

- Запись нескольких дискретных выходов (Force Multiple Coils) — функциональный код 15
- Чтение общей ссылки (Read General Reference) — функциональный код 20

Данная функция может обрабатывать несколько одновременных запросов и принимать 32-битные ссылочные значения. Она лучше всего подходит для расширения с целью работы с большим регистровым пространством и объектами данных, не имеющими ссылок типа «непривязанные переменные».

- Запись общей ссылки (Write General Reference) — функциональный код 21

Данная функция может обрабатывать несколько одновременных запросов и принимать 32-битные ссылочные значения. Она лучше всего подходит для расширения с целью работы с большим регистровым пространством и объектами данных, не имеющими ссылок типа «непривязанные переменные».

- Маскированная запись регистра (Mask Write Register) — функциональный код 22
- Чтение/запись нескольких регистров (Read/Write Multiple Registers) — функциональный код 23

Данная функция обрабатывает диапазон входных и выходных регистров как единую транзакцию. Это наилучший способ реализации обмена образом состояния с модулями ввода-вывода по правилам Modbus.

Таким образом, высокопроизводительное универсальное устройство сбора данных может реализовать функции 03, 16 и 23, сочетая быстрый регламентный обмен данными (23) с возможностью выполнения запросов или обновления отдельных объектов данных (03 и 16).

- Чтение очереди FIFO (Read FIFO Queue) — функциональный код 24

Довольно специализированная функция, предназначенная для передачи данных табличной структуры в хост-устройство по принципу FIFO. Полезна для определённого программного обеспечения регистрации событий.

2.4 Специальные функции устройства/производителя/сети

Все перечисленные ниже функции, хотя и упоминаются в руководстве по протоколу Modbus, обладают сильной аппаратной зависимостью и поэтому не пригодны для целей обеспечения совместимости.

- Диагностика (Diagnostics) — функциональный код 08
- Программирование (484) — функциональный код 09
- Опрос (484) — функциональный код 10
- Получение значения счётчика коммуникационных событий (Modbus) — функциональный код 11
- Получение журнала коммуникационных событий (Modbus) — функциональный код 12
- Программирование (584/984) — функциональный код 13
- Опрос (584/984) — функциональный код 14
- Сообщение идентификатора Ведомого устройства (Report Slave ID) — функциональный код 17
- Программирование (884/u84) — функциональный код 18
- Восстановление связи (884/u84) — функциональный код 19
- Программирование (принципиальное) — функциональный код 40
- Замена встроенного ПО (Firmware Replacement) — функциональный код 125
- Программирование (584/984) — функциональный код 126
- Сообщение локального адреса (Modbus) — функциональный код 127

3. Структура протокола

В данном разделе описывается общий формат инкапсуляции запроса и/или ответа Modbus при передаче по сети Modbus-TCP. Следует отметить, что структура тела запроса и ответа (от функционального кода до конца поля данных) имеет точно такой же формат и значение, что и в других вариантах Modbus, а именно:

- Modbus на основе последовательного интерфейса — кодировка ASCII
- Modbus на основе последовательного интерфейса — кодировка RTU (двоичная)
- Сеть Modbus Plus — канал данных

В этих вариантах различаются лишь формат кадра, режим обнаружения ошибок и описание адреса. Все запросы протокола Modbus-TCP отправляются по протоколу TCP через зарегистрированный порт 502.

Запросы обычно передаются в полудуплексном режиме в рамках одного соединения. Это означает, что пока соединение занято ответом, другие запросы отправлять нельзя. Некоторые устройства используют несколько TCP-соединений для поддержания высокой пропускной способности.

Поле «адрес Ведомого устройства» Modbus заменено однобайтовым полем «Идентификатор устройства» (Unit Identifier) и используется для связи через такие устройства, как мосты и шлюзы, которые с одним IP-адресом поддерживают несколько независимых оконечных устройств.

Запрос и ответ содержат шестибайтовый префикс следующего формата:

- Байт 0: Идентификатор транзакции — копируется сервером — обычно 0
- Байт 1: Идентификатор транзакции — копируется сервером — обычно 0
- Байт 2: Идентификатор протокола = 0
- Байт 3: Идентификатор протокола = 0
- Байт 4: Поле длины (старший байт) = 0 (все сообщения короче 256)
- Байт 5: Поле длины (младший байт) = количество последующих байт
- Байт 6: Идентификатор устройства (бывший «адрес Ведомого устройства»)
- Байт 7: Функциональный код Modbus
- Байт 8 и далее: Требуемые данные

Таким образом, для примера «прочитать 1 регистр со смещением 4 от устройства с UI 9, возвращающего значение 5», запрос и ответ будут выглядеть так:

Запрос: 00 00 00 00 00 06 09 03 00 04 00 01

Ответ: 00 00 00 00 00 05 09 03 02 00 05

Можно заметить, что в Modbus-TCP не требуются поля контроля «CRC-16» или «LRC». Вместо этого для проверки целостности пакетов используются механизмы контрольных сумм TCP/IP и канального уровня (Ethernet).

4. Ссылочные значения протокола для классов совместимости

Обратите внимание, что в примерах запрос и ответ приведены начиная с байта функционального кода. Как отмечалось ранее, в случае Modbus-TCP имеется транспортно-зависимый префикс из 7 байт:

ref ref 00 00 00 len unit

Первые два байта «ref ref» не имеют определённого значения для сервера и просто побайтно копируются из запроса в ответ для удобства клиента. Отдельный клиент обычно устанавливает это значение в 0.

В данном примере формат запроса и ответа следующий (пример запроса «чтение регистров», подробно описанный в последующих разделах):

03 00 00 00 01 → 03 02 12 34

Это означает, что при добавлении префикса путём шестнадцатеричной конкатенации байтов полное сообщение в TCP-соединении будет иметь следующий вид (при условии, что идентификатор устройства равен 09):

Запрос: 00 00 00 00 00 06 09 03 00 00 00 01

Ответ: 00 00 00 00 00 05 09 03 02 12 34

4.1 Подробное описание команд Класса 0

4.1.1 Чтение нескольких регистров хранения (FC 03)

Запрос:

Байт 0: FC = 03

Байты 1–2: Ссылочное значение

Байты 3–4: Количество регистров (1–125)

Ответ:

Байт 0: FC = 03

Байт 1: Количество байт в ответе ($B = 2 \times \text{количество регистров}$)

Байты 2–(B+1): Значения регистров

Исключение:

Байт 0: FC = 83 (hex)

Байт 1: Код исключения = 01 или 02

Пример: Чтение 1 регистра по ссылочному значению 0 (40001 в Modicon 984), возвращается значение 1234 (hex):

03 00 00 00 01 → 03 02 12 34

4.1.2 Запись нескольких регистров хранения (FC 16)

Запрос:

Байт 0: FC = 10 (hex)

Байты 1–2: Ссылочное значение

Байты 3–4: Количество регистров (1–100)

Байт 5: Количество байт ($B = 2 \times \text{количество регистров}$)

Байты 6–(B+5): Значения регистров

Ответ:

Байт 0: FC = 10 (hex)

Байты 1–2: Ссылочное значение

Байты 3–4: Количество регистров

Исключение:

Байт 0: FC = 90 (hex)

Байт 1: Код исключения = 01 или 02

Пример: Запись 1 регистра по ссылочному значению 0 (40001 в Modicon 984) значением 1234 (hex):

10 00 00 00 01 02 12 34 → 10 00 00 00 01

4.2 Подробное описание команд Класа 1

4.2.1 Чтение дискретных выходов (FC 01)

Запрос:

Байт 0: FC = 01

Байты 1–2: Ссылочное значение

Байты 3–4: Количество бит (1–2000)

Ответ:

Байт 0: FC = 01

Байт 1: Количество байт в ответе ($B = (\text{количество бит} + 7) / 8$)

Байты 2–(B+1): Значения бит (**младший значимый бит соответствует первому дискретному выходу**)

Исключение:

Байт 0: FC = 81 (hex)

Байт 1: Код исключения = 01 или 02

Пример: Чтение 1 дискретного выхода по ссылочному значению 0 (00001 в Modicon 984), возвращается значение 1:

01 00 00 00 01 → 01 01 01

Примечание: формат возвращаемых данных отличается от архитектуры big-endian. Кроме того, если данный запрос подразумевает обращения к словам, не выровненным по 16-битной границе, его обработка в Ведомом устройстве потребует повышенных вычислительных затрат.

4.2.2 Чтение дискретных входов (FC 02)

Запрос:

Байт 0: FC = 02

Байты 1–2: Ссылочное значение

Байты 3–4: Количество бит (1–2000)

Ответ:

Байт 0: FC = 02

Байт 1: Количество байт в ответе ($B = (\text{количество бит} + 7) / 8$)

Байты 2–(B+1): Значения бит (**младший значимый бит соответствует первому дискретному входу**)

Исключение:

Байт 0: FC = 82 (hex)

Байт 1: Код исключения = 01 или 02

Пример: Чтение 1 дискретного входа по ссылочному значению 0 (10001 в Modicon 984), возвращается значение 1:

02 00 00 00 01 → 02 01 01

Примечание: формат возвращаемых данных отличается от архитектуры big-endian. Кроме того, если данный запрос подразумевает обращения к словам, не выровненным по 16-битной границе, его обработка в Ведомом устройстве потребует повышенных вычислительных затрат.

4.2.3 Чтение входных регистров (FC 04)

Запрос:

Байт 0: FC = 04

Байты 1–2: Ссылочное значение

Байты 3–4: Количество регистров (1–125)

Ответ:

Байт 0: FC = 04

Байт 1: Количество байт в ответе ($B = 2 \times \text{количество регистров}$)

Байты 2–(B+1): Значения регистров

Исключение:

Байт 0: FC = 84 (hex)

Байт 1: Код исключения = 01 или 02

Пример: чтение 1 входного регистра по ссылочному значению 0 (30001 в Modicon 984), возвращается значение 1234 (hex):

04 00 00 00 01 → 04 02 12 34

4.2.4 Запись одного дискретного выхода (FC 05)

Запрос:

Байт 0: FC = 05

Байты 1–2: Ссылочное значение

Байт 3: = FF — включить дискретный выход, = 00 — выключить

Байт 4: = 00

Ответ:

Байт 0: FC = 05

Байты 1–2: Ссылочное значение

Байт 3: = FF — включить дискретный выход, = 00 — выключить (эхо)

Байт 4: = 00

Исключение:

Байт 0: FC = 85 (hex)

Байт 1: Код исключения = 01 или 02

Пример: запись значения 1 в 1 дискретный выход по ссылочному значению 0 (00001 в Modicon 984):

05 00 00 FF 00 → 05 00 00 FF 00

4.2.5 Запись одного регистра хранения (FC 06)

Запрос:

Байт 0: FC = 06

Байты 1–2: Ссылочное значение

Байты 3–4: Значение регистра

Ответ:

Байт 0: FC = 06

Байты 1–2: Ссылочное значение

Байты 3–4: Значение регистра

Исключение:

Байт 0: FC = 86 (hex)

Байт 1: Код исключения = 01 или 02

Пример: Запись значения 1234 (hex) в 1 регистр по ссылочному значению 0 (40001 в Modicon 984):

06 00 00 12 34 → 06 00 00 12 34

4.2.6 Чтение слова состояния исключений (FC 07)

Примечание: «Слово состояния исключений» не связано с «исключительным ответом». Сообщение «Чтение слова состояния исключений» было предусмотрено в ранних версиях Modbus для обеспечения максимальной скорости ответа при опросе многоточечных сетей на низких скоростях передачи. ПЛК обычно специально выделяли диапазон из 8 дискретных выходов (Coils), опрашиваемых данным сообщением.

Запрос:

Байт 0: FC = 07

Ответ:

Байт 0: FC = 07

Байт 1: Слово состояния исключений (обычно соответствует заранее определённый диапазону из 8 дискретных выходов)

Исключение:

Байт 0: FC = 87 (hex)

Байт 1: Код исключения = 01 или 02

Пример: Чтение слова состояния исключений, возвращается значение 34 (hex):

07 → 07 34

4.3 Подробное описание команд Класа 2

4.3.1 Запись нескольких дискретных выходов (FC 15)

Запрос:

Байт 0: FC = 0F (hex)

Байты 1–2: Ссылочное значение

Байты 3–4: Количество бит (1–800)

Байт 5: Количество байт ($B = (\text{количество бит} + 7) / 8$)

Байты 6–(B+5): Записываемые данные (младший значимый бит = первый дискретный выход)

Ответ:

Байт 0: FC = 0F (hex)

Байты 1–2: Ссылочное значение

Байты 3–4: Количество бит

Исключение:

Байт 0: FC = 8F (hex)

Байт 1: Код исключения = 01 или 02

Пример: Запись значений 0, 0, 1 в 3 дискретных выхода по ссылочному значению 0 (00001 в Modicon 984):

0F 00 00 00 03 01 04 → 0F 00 00 00 03

Примечание: формат возвращаемых данных отличается от архитектуры big-endian. Кроме того, если данный запрос подразумевает обращения к словам, не выровненным по 16-битной границе, его обработка в Ведомом устройстве потребует повышенных вычислительных затрат.

4.3.2 Чтение общей ссылки (FC 20)

Запрос:

Байт 0: FC = 14 (hex)

Байт 1: Количество байт остатка запроса ($= 7 \times \text{количество групп}$)

Байт 2: Тип ссылочного значения для первой группы = 06 для расширенной регистровой памяти 6xxxx

Байты 3–6: Ссылочное значение для первой группы = смещение в памяти для внешней памяти 6xxxx = 32-битное ссылочное значение для регистров 4xxxx

Байты 7–8: Количество регистров в первой группе

Байты 9–15: (повтор со байта 2 для второй группы)

...

Ответ:

Байт 0: FC = 14 (hex)

Байт 1: Общее количество байт в ответе ($= \text{количество групп} + \text{общее количество байт данных групп}$)

Байт 2: Количество байт в первой группе ($B1 = 1 + (2 \times \text{количество регистров})$)

Байт 3: Тип ссылки для первой группы

Байты 4–(B1+2): Значения регистров первой группы

Байт (B1+3): Количество байт во второй группе ($B2 = 1 + (2 \times \text{количество регистров})$)

Байт (B1+4): Тип ссылки для второй группы

Байты (B1+5)–(B1+B2+2): Значения регистров второй группы

...

Исключение:

Байт 0: FC = 94 (hex)

Байт 1: Код исключения = 01, или 02, или 03, или 04

Пример 1: Чтение 1 расширенного регистра по ссылочному значению 1:2 (в Modicon 984 — внешняя память 1, смещение 2), возвращается значение 1234 (hex):

14 07 06 00 01 00 02 00 01 → 14 04 03 06 12 34

Пример 2 (перспективный): Чтение 1 регистра по ссылочному значению 0 (возвращается 1234 hex) и 2 регистров по ссылочному значению 5 (возвращаются 5678 и 9ABC hex):

14 0E 04 00 00 00 00 00 01 04 00 00 00 05 00 02 → 14 0A 03 04 12 34 05 04 56 78 9A BC

Примечание: ограничения размера передачи трудно определить точной математической формулой. В общем случае размер сообщения запроса и ответа ограничен 256 байтами ввиду ограничений размера буфера и с учётом общей длины каждого кадра данных запроса и ответа. Если Ведомое устройство отказывается отправлять сообщение из-за слишком большого размера ответа, генерируется исключение типа 04.

4.3.3 Запись общей ссылки (FC 21)

Запрос:

Байт 0: FC = 15 (hex)

Байт 1: Количество байт остатка запроса

Байт 2: Тип ссылочного значения для первой группы = 06 для расширенной регистровой памяти 6xxxx

Байты 3–6: Ссылочное значение для первой группы = смещение в памяти для внешней памяти 6xxxx = 32-битное ссылочное значение для регистров 4xxxx

Байты 7–8: Количество регистров в первой группе (W1)

Байты 9–(8 + 2 × W1): Данные регистров первой группы

(повтор структуры группы, начиная с байта 2, для других групп)

...

Ответ: Ответ является прямым отражением запроса.

Байт 0: FC = 15 (hex)

Байт 1: Количество байт остатка запроса

Байт 2: Тип ссылочного значения для первой группы = 06 для расширенной регистровой памяти 6xxxx

Байты 3–6: Ссылочное значение для первой группы = смещение в памяти для внешней памяти 6xxxx = 32-битное ссылочное значение для регистров 4xxxx

Байты 7–8: Количество регистров в первой группе (W1)

Байты 9–(8 + 2 × W1): Данные регистров первой группы

(повтор структуры группы, начиная с байта 2, для других групп)

...

Исключение:

Байт 0: FC = 95 (hex)

Байт 1: Код исключения = 01, или 02, или 03, или 04

Пример 1: Запись 1 расширенного регистра по ссылочному значению 1:2 (в Modicon 984 — внешняя память 1, смещение 2) значением 1234 (hex):

15 09 06 00 01 00 02 00 01 12 34 → 15 09 06 00 01 00 02 00 01 12 34

Пример 2 (перспективный): Запись 1 регистра по ссылочному значению 0 значением 1234 (hex) и 2 регистров по ссылочному значению 5 значениями 5678 и 9ABC (hex):

15 14 04 00 00 00 00 00 01 12 34 04 00 00 00 05 00 02 56 78 9A BC

→ 15 14 04 00 00 00 00 00 01 12 34 04 00 00 00 05 00 02 56 78 9A BC

Примечание: ограничения размера передачи трудно определить точной математической формулой. В общем случае размер сообщения запроса и ответа ограничен 256 байтами ввиду ограничений размера буфера и с учётом общей длины каждого кадра данных запроса и ответа. Если Ведомое устройство отказывается отправлять сообщение из-за слишком большого размера ответа, генерируется исключение типа 04.

4.3.4 Маскированная запись регистра (FC 22)

Запрос:

Байт 0: FC = 16 (hex)

Байты 1–2: Ссылочное значение

Байты 3–4: Маска AND для регистра

Байты 5–6: Маска OR для регистра

Ответ:

Байт 0: FC = 16 (hex)

Байты 1–2: Ссылочное значение

Байты 3–4: Маска AND для регистра

Байты 5–6: Маска OR для регистра

Исключение:

Байт 0: FC = 96 (hex)

Байт 1: Код исключения = 01 или 02

Пример: изменение битового поля 0–3 регистра на значение 4 (hex) по ссылочному значению 0 (40001 в Modicon 984) (AND с 000F, OR с 0004):

16 00 00 00 0F 00 04 → 16 00 00 00 0F 00 04

4.3.5 Чтение/запись нескольких регистров (FC 23)

Запрос:

Байт 0: FC = 17 (hex)

Байты 1–2: Ссылочное значение для чтения

Байты 3–4: Количество регистров для чтения (1–125)

Байты 5–6: Ссылочное значение для записи

Байты 7–8: Количество регистров для записи (1–100)

Байт 9: Количество байт ($B = 2 \times \text{количество регистров для записи}$)

Байты 10–(B+9): Значения регистров

Ответ:

Байт 0: FC = 17 (hex)

Байт 1: Количество байт ($B = 2 \times \text{количество регистров для чтения}$)

Байты 2–(B+1): Значения регистров

Исключение:

Байт 0: FC = 97 (hex)

Байт 1: Код исключения = 01 или 02

Пример: Запись 1 регистра по ссылочному значению 3 (40004 в Modicon 984) значением 0123 (hex), чтение 2 регистров по ссылочному значению 0 с возвратом значений 0004 и 5678 (hex):

17 00 00 00 02 00 03 00 01 02 01 23 → 17 04 00 04 56 78

Примечание: если регистры попеременно участвуют в операциях чтения и записи, результат не определён. Часть устройств выполняет сначала запись, затем чтение; другая часть — сначала чтение, затем запись.

4.3.6 Чтение очереди FIFO (FC 24)

Запрос:

Байт 0: FC = 18 (hex)

Байты 1–2: Ссылочное значение

Ответ:

Байт 0: FC = 18 (hex)

Байты 1–2: Количество байт ($B = 2 + \text{количество регистров}$) (макс. 64)

Байты 3–4: Количество регистров (количество регистров, накопленных в FIFO) (макс. 31)

Байты 5–(B+2): Данные регистров, начиная с начала FIFO

Исключение:

Байт 0: FC = 98 (hex)

Байт 1: Код исключения = 01, или 02, или 03

Пример: чтение содержимого секции FIFO, начиная со ссылочного значения 0005 (40006 в Modicon 984), содержит 2 регистра со значениями 1234 и 5678 (hex):

18 00 05 → 18 00 06 00 02 12 34 56 78

Примечание: реализация данной функции на устройствах 984 весьма ограничена по универсальности — предполагается, что данная секция регистров содержит счётчик со значениями от 0 до 31, за которым следуют данные — до 31 регистра. После выполнения функции счётчик регистров не сбрасывается в 0, как можно было бы ожидать от операции FIFO.

В целом данную функцию можно рассматривать как ограниченное подмножество функции 03 — Чтение нескольких регистров хранения, поскольку последняя способна выполнить необходимую функциональность.

5. Коды исключений

При возникновении ошибки Ведомое устройство возвращает набор определённых кодов исключений. Следует отметить, что Ведущее устройство может «пробно» отправлять команды и по полученному успешному или исключительному ответу определять, какие функции Modbus поддерживает подчинённое устройство, а также размер различных доступных областей данных Ведомого устройства.

Все исключения обозначаются добавлением 0x80 к функциональному коду запроса; за этим байтом следует единственный байт с кодом причины, как показано в примере ниже:

03 12 34 00 01 → 83 02

Запрос чтения 1 регистра по индексу 0x1234 вернул исключение типа 2 — «Недопустимый адрес данных».

Перечень кодов исключений:

01 — Недопустимая функция (*Illegal Function*)

Функциональный код, полученный в запросе, не разрешён для данного Ведомого устройства. Это может означать, что данный функциональный код применим только к более новым контроллерам и не поддерживается выбранным устройством. Также может указывать, что

Ведомое устройство находится в таком состоянии, при котором обработка данного запроса невозможна — например, запрос на возврат значений регистров, когда устройство ещё не сконфигурировано.

02 — Недопустимый адрес данных (Illegal Data Address)

Адрес данных, полученный в запросе, не является допустимым для данного Ведомого устройства. Точнее, комбинация ссылочного значения и длины передачи некорректна. Например, для контроллера со 100 регистрами запрос со смещением 96 и длиной 4 будет успешен, тогда как запрос со смещением 96 и длиной 5 вызовет исключение 02.

03 — Недопустимое значение данных (Illegal Data Value)

Значение, содержащееся в поле данных запроса, недопустимо для данного Ведомого устройства. Это указывает на структурную ошибку в остатке сложного запроса — например, неверно указана неявная длина. Поскольку протокол Modbus не интерпретирует особые значения специальных регистров, данный код не означает, что объект данных, записываемый в регистр, имеет значение, выходящее за пределы ожидаемого приложением.

04 — Недопустимая длина ответа (Illegal Response Length)

Указывает, что запрос сформировал бы ответ, размер которого превышает доступный размер данных Modbus. Используется только для функций, порождающих многосоставной ответ, таких как функции 20 и 21.

05 — Подтверждение (Acknowledge)

Зарезервировано для команд, связанных с программированием.

06 — Ведомое устройство занято (Slave Device Busy)

Зарезервировано для команд, связанных с программированием.

07 — Отказ (Negative Acknowledge)

Зарезервировано для команд, связанных с программированием.

08 — Ошибка контроля чётности памяти (Memory Parity Error)

Зарезервировано для функций 20 и 21. Указывает, что расширенная область файлов не прошла проверку целостности.

0A — Путь шлюза недоступен (Gateway Path Unavailable)

Зарезервировано для шлюзов Modbus Plus. Указывает, что шлюз не смог выделить путь Modbus Plus для обработки запроса. Обычно означает ошибку конфигурирования шлюза.

0B — Устройство за шлюзом не отвечает (Gateway Target Device Failed to Respond)

Зарезервировано для шлюзов Modbus Plus. Указывает, что ответ от целевого устройства не был получен. Обычно означает, что устройство не подключено к сети.

6. Области памяти Modbus

Области памяти контроллера (или устройства Modbus), рассматриваемые в протоколе Modbus, обозначаются как 3xxxx и 4xxxx.

Обозначение области памяти	Наименование	Тип	Вид операции	Адрес ячейки памяти
3xxxx	Входные регистры (Input Registers)	Слово	Только чтение	30001–3xxxx, xxxx: зависит от устройства
4xxxx	Регистры хранения / выходные регистры (Holding/Output Registers)	Слово	Чтение и запись	40001–4xxxx, xxxx: зависит от устройства

7. Функции Modbus

Уровень приложения Modbus определяет формат телеграмм Modbus и сервисные функции.

Протокол Modbus определяет простой протокольный блок данных (PDU), независимый от нижележащего коммуникационного уровня. Отображение протокола Modbus на конкретную шину или сеть может вводить дополнительные поля в прикладной блок данных (ADU).

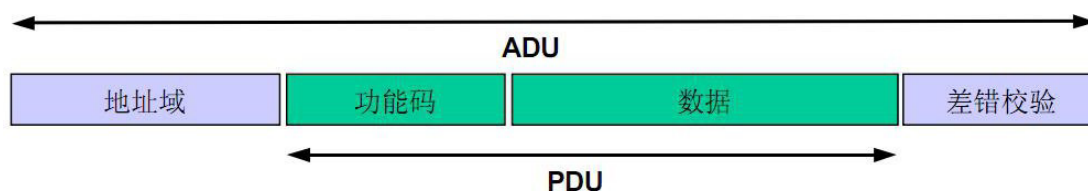


Рис. 3-7-1. Универсальный кадр Modbus (общий формат кадра Modbus)

Универсальный кадр Modbus показан на Рис. 3-7-1 выше. Инкапсуляция запроса или ответа Modbus в сети Modbus-TCP IP показана на Рис. 3-7-2 ниже:

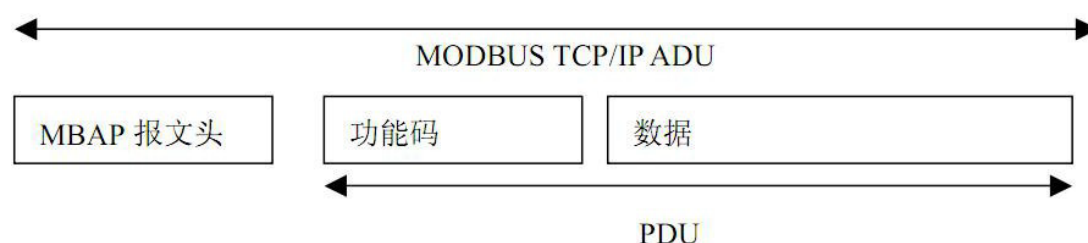


Рис. 3-7-2. Запрос/ответ Modbus поверх TCP/IP

Как показано на Рис. 3-7-2 выше, для идентификации прикладного блока данных Modbus (ADU) при передаче поверх TCP/IP используется специальный заголовок. Этот заголовок называется MBAP-заголовком (Modbus Application Protocol Header — заголовок протокола приложения Modbus).

MBAP-заголовок включает следующие поля:

Поле	Длина	Описание	Клиент	Сервер
Идентификатор транзакции (Transaction Identifier)	2 байта	Идентификатор транзакции «запрос–ответ» Modbus	Иницируется клиентом	Копируется сервером из принятого запроса
Идентификатор протокола (Protocol Identifier)	2 байта	0 = протокол Modbus	Иницируется клиентом	Копируется сервером из принятого запроса
Длина (Length)	2 байта	Количество последующих байт	Иницируется клиентом (запрос)	Иницируется сервером (ответ)
Идентификатор устройства (Unit Identifier)	1 байт	Идентификатор удалённого Вedomого устройства, подключённого по последовательной линии или другой шине	Иницируется клиентом	Копируется сервером из принятого запроса

Заголовок имеет длину 7 байт.

Идентификатор транзакции (Transaction Identifier) — используется для сопоставления запроса и ответа. В ответе Modbus-сервер копирует идентификатор транзакции из запроса.

Идентификатор протокола (Protocol Identifier) — используется для мультиплексирования в рамках системы. Значение 0 идентифицирует протокол Modbus.

Длина (Length) — поле длины содержит количество байт следующих далее полей, включая идентификатор устройства и поле данных.

Идентификатор устройства (Unit Identifier) — используется для маршрутизации в системе. Специально предназначен для связи через шлюзы между сетью Ethernet TCP-IP и последовательными линиями Modbus или Modbus+ с Ведомыми устройствами на этих линиях. Modbus-клиент задаёт это поле в запросе, а сервер обязан вернуть его с тем же значением в ответе.

Ниже описаны форматы PDU для нескольких функциональных кодов, поддерживаемых сообщениями Modbus TCP/IP.

7.1 Чтение регистров хранения (Read Holding Registers)

Функциональный код: 03H

Формат запроса Ведущего устройства (МВАР-заголовок + PDU):

Заголовок МВАР (Modbus Application Protocol header):

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	06	01

PDU (блок данных протокола):

Функциональный код	Старший байт начального адреса регистра	Младший байт начального адреса регистра	Старший байт количества регистров	Младший байт количества регистров
03	00	6B(107)	00	03

Функция: чтение значений регистров хранения Ведомого устройства (4XXXXX).

Примечание: начальный адрес регистра 00000 в телеграмме соответствует адресу 40001 в устройстве; остальные адреса смещаются аналогично.

В данном примере: начальный адрес = 006BH = 107, что соответствует адресу 40108; количество регистров = 0003; конечный адрес = 40108 + 3 – 1 = 40110. Таким образом, данный запрос выполняет чтение трёх регистров хранения (40108–40110) Ведомого устройства с идентификатором устройства 01.

Формат ответа Ведомого устройства (МВАР-заголовок + PDU):

Заголовок МВАР:

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	09	01

Часть PDU:

Функц. код	Счётчик байт	Регистр 40108 (ст. байт)	Регистр 40108 (м.л. байт)	Регистр 40109 (ст. байт)	Регистр 40109 (м.л. байт)	Регистр 40110 (ст. байт)	Регистр 40110 (м.л. байт)
03	6	02	2B	01	06	2A	64

Ведомое устройство возвращает значения регистров хранения 40108–40110: (40108) = 022BH, (40109) = 0106H, (40110) = 2A64H.

7.2 Чтение входных регистров (Read Input Registers)

Функциональный код: 04H

Формат запроса Ведущего устройства (МВАР-заголовок + PDU)

Заголовок МВАР:

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	06	01

Часть PDU:

Функц. код	Старший байт начального адреса регистра	Младший байт начального адреса регистра	Старший байт количества регистров	Младший байт количества регистров
04	00	08	00	01

Функция: чтение значений входных регистров Ведомого устройства (3XXXX).

Примечание: начальный адрес регистра 00000 в телеграмме соответствует адресу 30001 в устройстве; остальные адреса смещаются аналогично.

В данном примере: чтение значения входного регистра Ведомого устройства с идентификатором 01. Начальный адрес = 0008H = 8, что соответствует адресу 30009; количество регистров = 0001; конечный адрес = 30009.

Таким образом, данный запрос выполняет чтение одного регистра хранения 30009 Ведомого устройства с идентификатором устройства 01.

Формат ответа Ведомого устройства (МВАР-заголовок + PDU):

Заголовок МВАР:

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	05	01

Часть PDU:

Функц. код	Счётчик байт	Младший байт регистра 30009	Старший байт регистра 30009
04	2	01	01

Ведомое устройство возвращает значение входного регистра 30009: (30009) = 0101H.

7.3 Запись одного регистра (Write Single Register)

Функциональный код: 06H

Формат запроса Ведущего устройства (МВАР-заголовок + PDU)

Заголовок МВАР:

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	06	01

Часть PDU:

Функц. код	Старший байт адреса регистра	Младший байт адреса регистра	Старший байт данных	Младший байт данных
06	00	08	00	17

Функция: запись значения в один регистр хранения Ведомого устройства (4XXXXX).

Примечание: начальный адрес регистра хранения 40000 в телеграмме соответствует адресу 40001 в устройстве; остальные адреса смещаются аналогично.

В данном примере: запись значения в один регистр хранения Ведомого устройства с идентификатором 01. Адрес регистра = 0008H = 8, что соответствует адресу 40009. Таким образом, данный запрос выполняет запись значения 0017H в регистр 40009 Ведомого устройства с идентификатором 01.

Формат ответа (МВАР-заголовок + PDU):

Заголовок МВАР:

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	06	01

Часть PDU:

Функц. код	Старший байт адреса регистра	Младший байт адреса регистра	Старший байт данных	Младший байт данных
06	00	08	00	17

7.4 Запись нескольких регистров (Write Multiple Registers)

Функциональный код: 10H

Формат запроса Ведущего устройства (МВАР-заголовок + PDU)

Заголовок МВАР:

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	0B	01

Часть PDU:

Функц. код	Ст. байт адреса регистра	Мл. байт адреса регистра	Ст. байт кол-ва регистров	Мл. байт кол-ва регистров	Счётчик байт	Данные (ст. байт)	Данные (мл. байт)	Данные (ст. байт)	Данные (мл. байт)
10	00	87	00	02	04	01	05	0A	10

Функция: запись значений в несколько регистров хранения Ведомого устройства (4XXXXX).

Примечание: начальный адрес регистра хранения 40000 в телеграмме соответствует адресу 40001 в устройстве; остальные адреса смещаются аналогично.

В данном примере: запись значений в несколько регистров хранения Ведомого устройства с идентификатором 01. Начальный адрес регистра = 0087H = 135, что соответствует адресу 40135; количество регистров = 0002H = 2; конечный адрес = 40135 + 2 – 1 = 40136.

Таким образом, данный запрос выполняет запись двух значений в регистры хранения Ведомого устройства с идентификатором 01: 0105H → 40135; 0A10H → 40136.

Формат ответа (МВАР-заголовок + PDU):

Заголовок МВАР:

Идентификатор транзакции		Идентификатор протокола		Длина		Идентификатор устройства
E3	76	00	00	00	06	01

Часть PDU:

Функц. код	Старший байт адреса регистра	Младший байт адреса регистра	Старший байт данных	Младший байт данных
10	00	87	00	02

С полный список оборудования компании можно ознакомиться на нашем сайте:

www.industrialnets.ru

Если у вас возникли вопросы, напишите или позвоните нам:

e-mail: sales@i-nets.ru, support@i-nets.ru

Телефон: 8 800 222-32-80