

РЕПЕЙ



ПРОТОКОЛ ОБМЕНА
MODBUS RTU

КОМПЛЕКС ПОЭЛЕМЕНТНОГО КОНТРОЛЯ И ПРЕДИКТИВНОЙ ДИАГНОСТИКИ
АККУМУЛЯТОРНОЙ БАТАРЕИ
РЕПЕЙ

ПРОТОКОЛ ОБМЕНА MODBUS RTU

ОПИСАНИЕ ПРОТОКОЛА

МТ.РЕПЕЙ.01.01.ОП от 09.12.2020

Оглавление

1. Общие сведения	4
2. Физический уровень.....	4
3. Канальный уровень	4
4. Прикладной уровень	4
4.1. Функция 65	5
4.1.1. Чтение текущей даты и времени (подфункция 4).....	7
4.1.2. Установка текущей даты и времени (подфункция 5)	7
4.1.3. Чтение параметров местного времени (подфункция 6)	8
4.1.4. Запись параметров местного времени (подфункция 7).....	9
4.1.5. Запись уставок (подфункция 11).....	9
4.1.6. Чтение текущего состояния дискретных выходов (подфункция 17)	10
4.1.7. Чтение текущего состояния двоичных выходных сигналов (подфункция 19)	10
4.1.8. Чтение списка двоичных сигналов (подфункция 20).....	11
4.1.9. Чтение количества осциллограмм (подфункция 21)	12
4.1.10. Чтение информации об осциллограммах (подфункция 22)	12
4.1.11. Чтение информации о флеш-памяти (подфункция 23)	13
4.1.12. Чтение записей журнала (подфункция 25).....	15
4.1.13. Формирование файла, содержащего записи журнала за указанный период (подфункция 27).....	19
4.1.14. Открытие файла (подфункция 31)	19
4.1.15. Чтение файла (подфункция 33).....	20
4.1.16. Закрытие файла (подфункция 34)	21
4.1.17. Передача длинной команды (подфункция 239)	21
4.1.18. Получение информации о буфере сервера (подфункция 240).....	22
4.1.19. Чтение данных из буфера сервера (подфункция 241).....	22
4.1.20. Уведомление сервера о завершении чтения из буфера (подфункция 242).....	23
5. Приложение 1. Пример использования запроса передачи длинной команды (подфункция 239)	24
6. Литература.....	28

1. Общие сведения

При обмене данными Репей всегда является сервером, Конфигуратор – клиентом. Конфигуратора может обмениваться данными с Репей только напрямую (рис. 1).

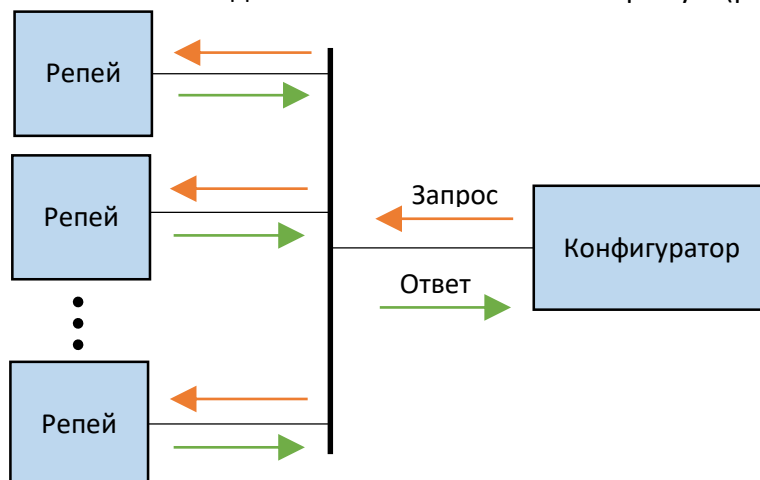


Рисунок 1. Обмен данными между Конфигуратором и Репей

2. Физический уровень

На физическом уровне применяется протокол ISO/IEC 8482:1993 (RS-485) [3].

3. Канальный уровень

На канальном уровне используется протокол MODBUS RTU [1], передающий данные в кадрах следующей структуры.

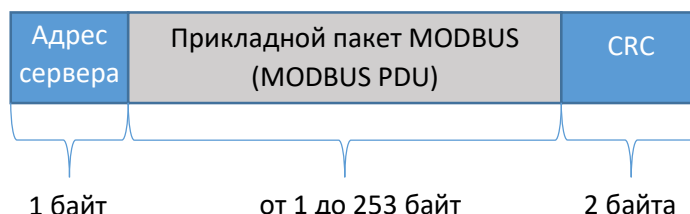


Рисунок 2. Структура кадра MODBUS RTU

4. Прикладной уровень

На прикладном уровне протокола обмена данными используется прикладной уровень протокола MODBUS [2]. Пакеты прикладного уровня подразделяются на запросы, которые клиент посылает серверу и ответы, которые сервер отправляет клиенту.

В дополнение к стандартным функциям MODBUS применяются также прикладные пакеты MODBUS с кодом функции 65 (рис. 3). Кроме кода функции эти пакеты содержат код подфункции, который определяет тип запроса, порядковый номер запроса, длину данных и сами данные. Порядковый номер запроса служит для сопоставления запросов и ответов. Этот номер назначается клиентом при создании запроса и возвращается сервером в ответе на запрос. Поле длины данных указывает количество байтов в следующем за ним поле данных. Если поле данных отсутствует, то длина данных равна 0. Во всех запросах и ответах с кодом функции 65 все многобайтные числа, содержащиеся в MODBUS PDU, передаются, начиная со старшего байта (big-endian).

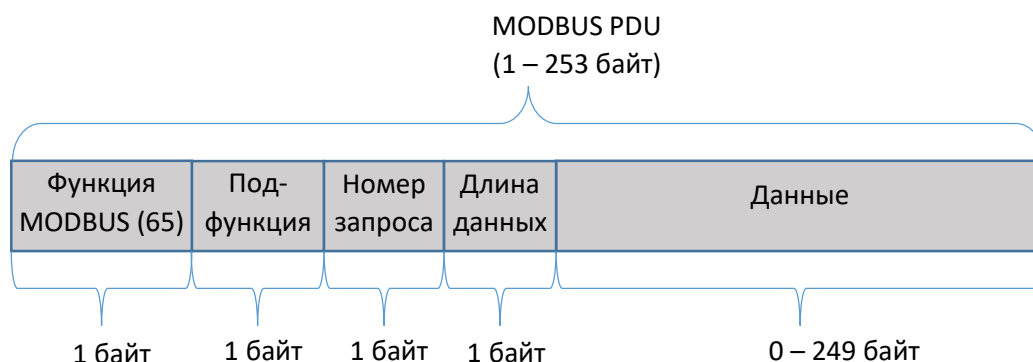


Рисунок 3. Структура прикладного пакета MODBUS (MODBUS PDU)

4.1. Функция 65

Запросы с кодом функции 65 и ответы на эти запросы имеют заголовки одинаковой структуры (рис. 3). Структура поля данных зависит от кода подфункции. Предусмотрена специальная подфункция ([239](#)), предназначенная для передачи команд, длина которых превышает объём стандартного прикладного пакета MODBUS (MODBUS PDU). Клиент, посылая серию запросов с этой подфункцией, может частями передать серверу команду такой же структуры, как MODBUS PDU обычного запроса (рис. 3). Единственное отличие команды от MODBUS PDU заключается в том, что её длина может быть больше максимального размера MODBUS PDU, и поэтому поле длины данных в команде указывает действительный объём поля данных команды только тогда, когда он не превышает 255 байт. Сервер, получая запросы с подфункцией 239, накапливает у себя фрагменты команды до тех пор, пока не получит её полностью, а затем исполняет эту команду и внутри ответа на запрос 239 отправляет клиенту результат выполнения команды. Этот результат оформляется как MODBUS PDU, содержащий ответ на запрос с тем кодом функции, который указан в первом байте команды.

Ответ на запрос с кодом функции 65 в поле данных MODBUS PDU содержит тип ответа (0 – короткий, 1 – длинный), код ответа (табл. 1) и, в некоторых случаях, поле данных ответа.

Ответы могут быть положительными и отрицательными. Положительный ответ означает, что запрос выполнен успешно. Отрицательный ответ означает, что сервер не смог выполнить запрос, при этом код ответа указывает причину, по которой запрос не удалось выполнить. Отрицательные ответы отправляются в тех случаях, когда ошибка, возникшая при выполнении запроса, не может быть описана кодами исключений MODBUS. Кроме того, как положительные, так и отрицательные ответы могут быть короткими и длинными. Короткий ответ (рис. 4, 5) целиком помещается в MODBUS PDU.



Рисунок 4. Структура поля данных PDU короткого ответа

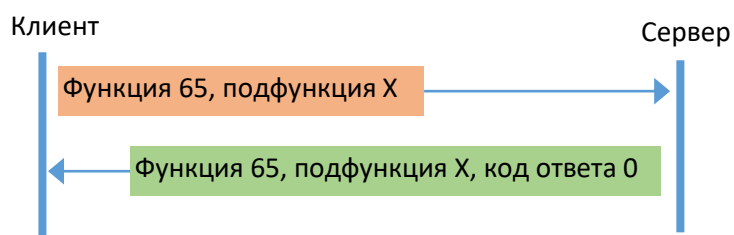


Рисунок 5. Обмен данными при получении короткого ответа

Длинный ответ (рис. 6, 7) указывает общее количество байтов данных ответа, которые передаются через специальный буфер сервера, и содержит первый фрагмент этих данных, начиная со смещения 0. Для чтения данных из буфера сервера используются подфункции [241](#) и [242](#).

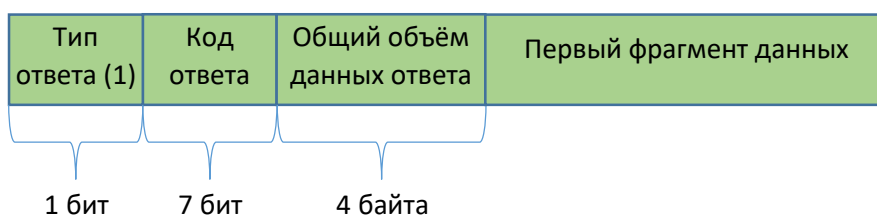


Рисунок 6. Структура поля данных PDU длинного ответа

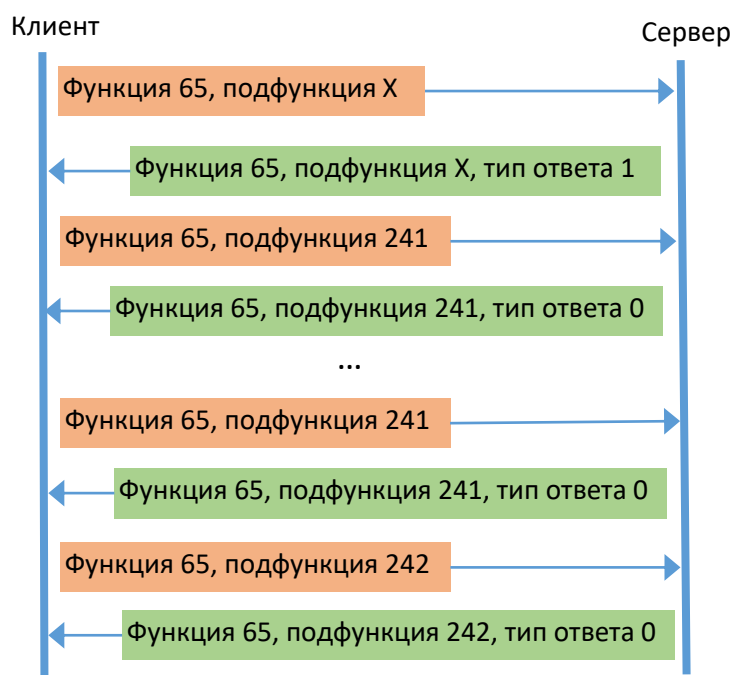


Рисунок 7. Обмен данными при получении длинного ответа

При ответе на один и тот же запрос длинные и короткие ответы с одним и тем же кодом ответа имеют одинаковую структуру поля данных.

Таблица 1. Коды ответов

Код ответа	Описание
0	Положительный ответ (запрос выполнен успешно).
1	Длина запроса не соответствует типу запроса.
2	Некорректные параметры запроса.
3	Ошибка в сервере при выполнении запроса.
4	Нет прав доступа к ресурсу.
5	Неправильный пароль.
6	Сеанс связи закрыт.
7	Истекло время хранения данных ответа в буфере сервера.
8	Данные ответа не помещаются в буфер сервера.
9	Ошибка при открытии файла.
10	Ошибка при чтении файла.
11	Ошибка при записи файла.
12	Ошибка при записи уставок.
13	Данные запроса не помещаются в буфер сервера.
14	Недопустимый идентификатор файла.
15	Недопустимый режим открытия файла.
16	Операция с файлом не соответствует режиму открытия файла.
17	Неизвестный код подфункции.
18	Длина поля данных запроса не соответствует длине, указанной в заголовке запроса.

4.1.1. Чтение текущей даты и времени (подфункция 4)

В запросе поле данных отсутствует.

Поле данных положительного ответа на запрос текущей даты и времени содержит метку времени с точностью до миллисекунд (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC) и смещение местного времени от UTC в минутах с учётом часового пояса и летнего времени (т.е. разность текущего местного времени и UTC в минутах, представленная как 16-битовое целое число со знаком). Положительное смещение означает, что местное время опережает UTC.



Рисунок 8. Поле данных положительного ответа на запрос текущей даты и времени

4.1.2. Установка текущей даты и времени (подфункция 5)

Поле данных запроса содержит метку времени с точностью до миллисекунд (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC).



Рисунок 9. Структура запроса установки текущей даты и времени

В положительном ответе на этот запрос поле данных отсутствует.

4.1.3. Чтение параметров местного времени (подфункция 6)

В запросе поле данных отсутствует.

Поле данных положительного ответа на запрос чтения параметров местного времени (рис. 10) содержит местный часовой пояс (разность между стандартным местным временем и UTC в минутах), момент перехода на летнее время, момент возврата к стандартному времени и разность летнего и стандартного времени в минутах. Моменты перехода и возврата указываются по местному времени и имеют структуру, описанную в табл. 2. Разность указывается как 16-битовое целое число со знаком, при этом положительная величина означает, что в момент перехода на летнее время часы переводятся на указанное количество минут вперёд, а отрицательная величина – назад. Если разность летнего и стандартного времени равна 0, то летнее время не применяется, и в течение всего года действует стандартное время, соответствующее часовому поясу; в этом случае моменты перехода на летнее время и возврата к стандартному времени не имеют смысла и могут содержать некорректные значения.

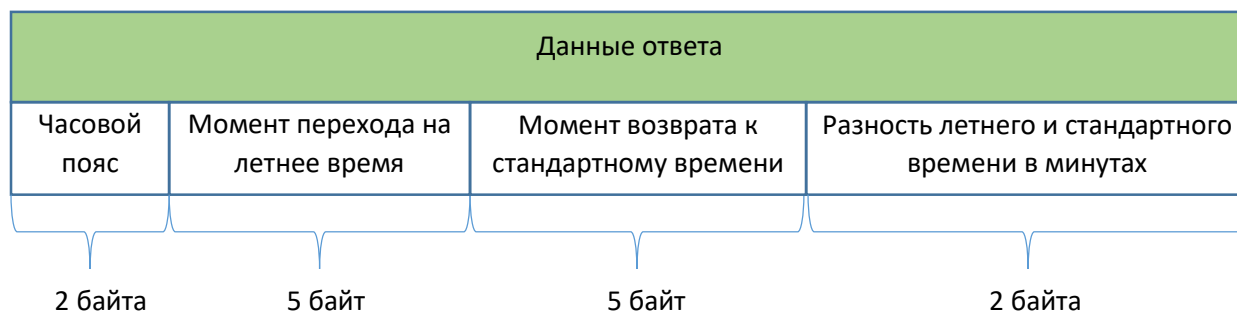


Рисунок 10. Поле данных положительного ответа на запрос чтения параметров летнего времени

Таблица 2. Структура момента времени

Смещение, байт	Размер, байт	Условное обозначение	Описание
0	1	Month	Номер месяца (от 1 до 12).
1	1	DayOfWeek	Код дня недели (табл. 3).
2	1	DayOfMonth	Порядковый номер дня в месяце. Если код дня недели равен 0, то параметр DayOfMonth содержит номер дня месяца, при этом 0 означает последний день месяца. При остальных значениях кода дня недели параметр DayOfMonth содержит порядковый номер указанного дня недели в месяце, при этом 0 означает последний день недели в месяце.
3	1	Hour	Час (от 0 до 23).
4	1	Minute	Минута (от 0 до 59).

Таблица 3. Коды дней недели

Код	Описание
0	Любой день недели
1	Понедельник
2	Вторник
3	Среда
4	Четверг
5	Пятница
6	Суббота
7	Воскресенье

Примеры моментов времени:

1. 2, 0, 21, 1, 15 – 21-е февраля, 1 час 15 минут.
2. 12, 7, 0, 20, 0 – последнее воскресенье декабря, 20 часов 0 минут.
3. 4, 1, 2, 2, 0 – второй понедельник апреля, 2 часа 0 минут.

4.1.4. Запись параметров местного времени (подфункция 7)

Поле данных запроса записи параметров местного времени имеет такую же структуру, как и поле данных ответа на запрос чтения этих параметров (см. п. 4.1.3).

В положительном ответе на этот запрос поле данных отсутствует.

4.1.5. Запись уставок (подфункция 11)

Поле данных запроса содержит список программ уставок. Каждый элемент списка программ уставок состоит из номера программы уставок, количества уставок для этой программы и списка уставок. Каждый элемент списка уставок состоит из идентификатора уставки и её значения. Значения всех уставок указываются как 32-битовые целые числа со знаком.

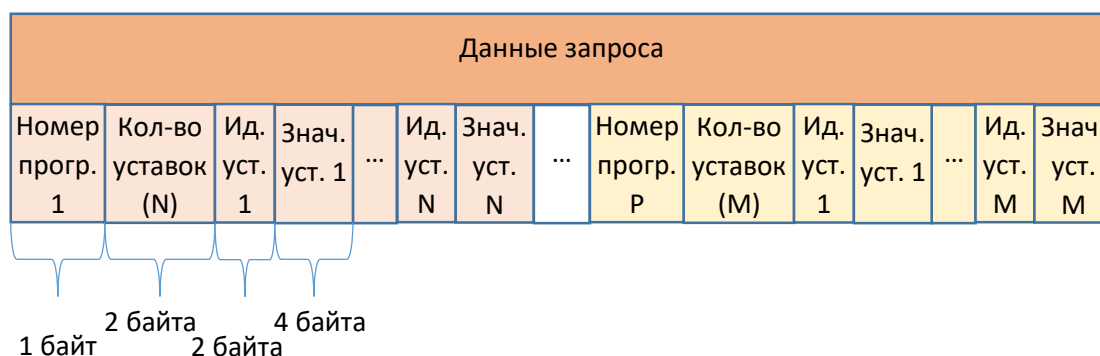


Рисунок 11. Структура запроса записи уставок

Нумерация программ уставок начинается с 0 (0 – все программы уставок, 1 – программа №1, 2 – программа №2 и т. д.)

В положительном ответе на этот запрос поле данных отсутствует.

4.1.6. Чтение текущего состояния дискретных выходов (подфункция 17)

В запросе поле данных отсутствует.

Поле данных положительного ответа на этот запрос содержит количество дискретных выходов и, собственно, состояния дискретных выходов. В поле состояния каждый бит отображает состояние соответствующего дискретного выхода.



Рисунок 12. Поле данных положительного ответа на запрос чтения состояния дискретных выходов

4.1.7. Чтение текущего состояния двоичных выходных сигналов (подфункция 19)

В запросе поле данных отсутствует.

Поле данных положительного ответа на этот запрос содержит количество двоичных выходных сигналов и состояния этих сигналов. В поле состояния каждый бит отображает состояние соответствующего сигнала.

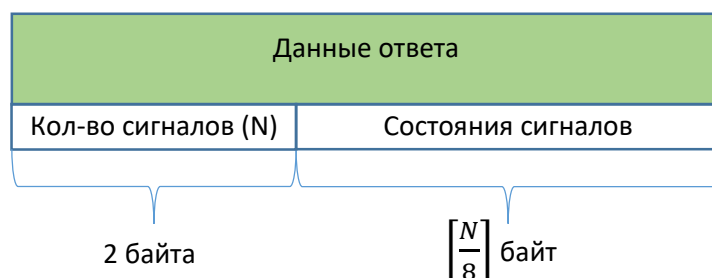


Рисунок 13. Поле данных положительного ответа на запрос чтения состояния двоичных выходных сигналов

4.1.8. Чтение списка двоичных сигналов (подфункция 20)

Поле данных запроса (рис. 14) содержит код типа двоичного сигнала (табл. 4), состояние запрашиваемых сигналов (0 или 1), и код структуры элемента списка сигналов (табл. 5).



Рисунок 14. Структура запроса списка двоичных сигналов

Поле данных положительного ответа на этот запрос повторяет данные запроса (код типа сигнала, состояние сигналов и код структуры элемента списка), а также содержит количество элементов списка (может быть равно 0) и сам список сигналов (если количество элементов списка не равно 0).

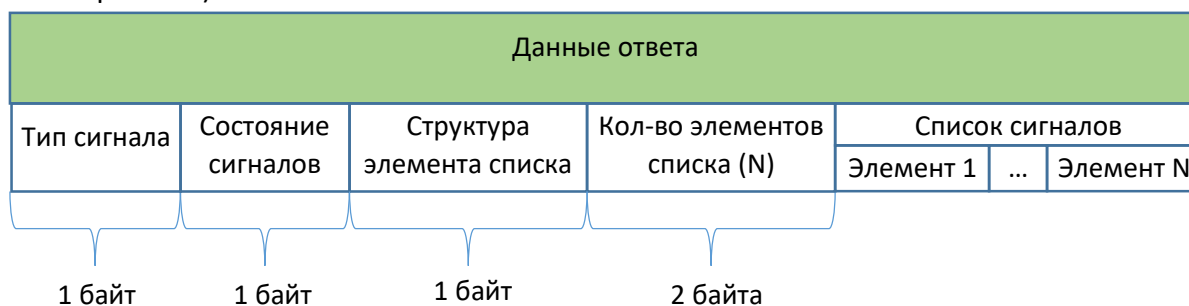


Рисунок 15. Поле данных положительного ответа на запрос чтения списка двоичных сигналов, находящихся в определённом состоянии.

Таблица 4. Типы двоичных сигналов

Код	Название
1	Двоичные выходы
3	Двоичные выходные сигналы
4	Сигналы пусковых органов

Таблица 5. Типы структур элемента списка сигналов

Код	Название
0	Краткое название сигнала (строка UTF-8, оканчивающаяся нулём)
1	Идентификатор сигнала (16-битовое число)
2	Идентификатор и краткое название сигнала (см. табл. 6)

Таблица 6. Структура элемента списка

Смещение, байт	Размер, байт	Условное обозначение	Описание
0	2	ID	Идентификатор сигнала.
2		ShortName	Краткое название сигнала (строка UTF-8, оканчивающаяся нулём)

4.1.9. Чтение количества осциллограмм (подфункция 21)

В запросе поле данных отсутствует.

Поле данных положительного ответа на этот запрос (см. рис. 16) содержит количество осциллограмм.

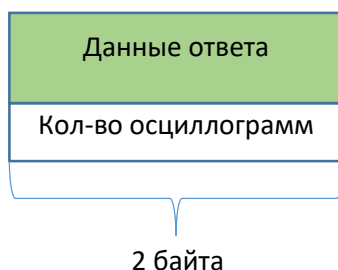


Рисунок 16. Поле данных положительного ответа на запрос количества осциллограмм

4.1.10. Чтение информации об осциллограммах (подфункция 22)

Поле данных запроса содержит порядковый номер начальной осциллограммы, (нумерация начинается с нуля, ноль соответствует самой новой осциллограмме) и количество осциллограмм, информацию о которых нужно прочитать.



Рисунок 17. Структура запроса чтения информации об осциллограммах

Поле данных положительного ответа содержит массив структур, описывающих осциллограммы, указанные в запросе. Каждая структура информации об осциллограмме содержит время создания осциллограммы (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC), длительность осциллограммы в миллисекундах, размер файла осциллограмма, тип (см. табл. 7 и табл. 8) и идентификатор сигнала, по которому была создана осциллограмма.

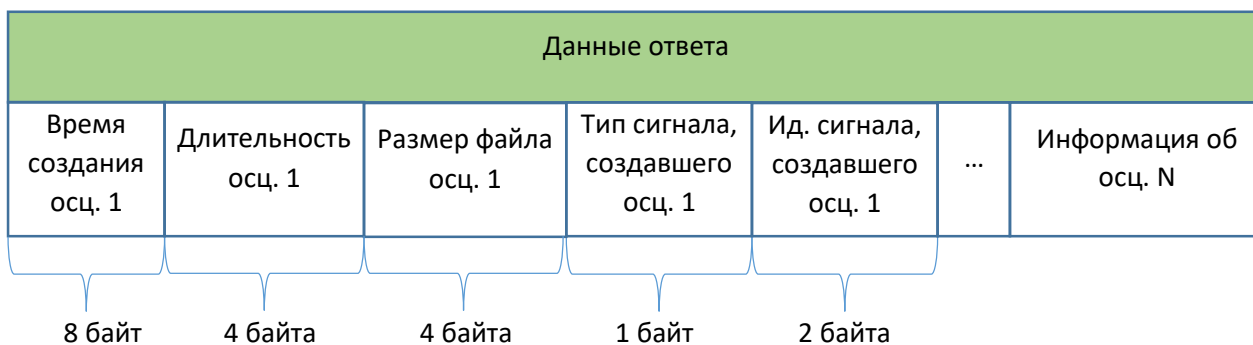


Рисунок 18. Поле данных положительного ответа на запрос информации об осциллограммах

Таблица 7. Типы сигналов, по которым была создана осциллограмма

Код	Название
1	Двоичные выходные сигналы
2	Программные сигналы (табл. 8)

Таблица 8. Программные сигналы, по которым была создана осциллограмма

Код	Название
3	Пуск по команде от конфигулятора
4	Пуск по команде от АСУ

4.1.11. Чтение информации о флеш-памяти (подфункция 23)

Поле данных запроса содержит код вида информации (см. табл. 9).



Рисунок 19. Структура запроса чтения информации о флеш-памяти

Таблица 9. Виды информации

Код	Вид информации
0	Вся информация
1	Информация о журнале сигнализации
2	Информация о системном журнале
4	Информация об осциллограммах
5	Информация о заполнении флеш-памяти
6	Информация о журнале аналоговых величин

Для всех видов информации, относящихся к журналам, поле данных положительного ответа (см. рис. 20) содержит номер версии структуры журнала, номер версии ПО, создавшего журнал, размер записи журнала, максимально допустимое количество записей, текущее количество записей, уникальный идентификатор последней записи (сквозной порядковый номер записи во всех журналах), метка времени последней записи (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC).

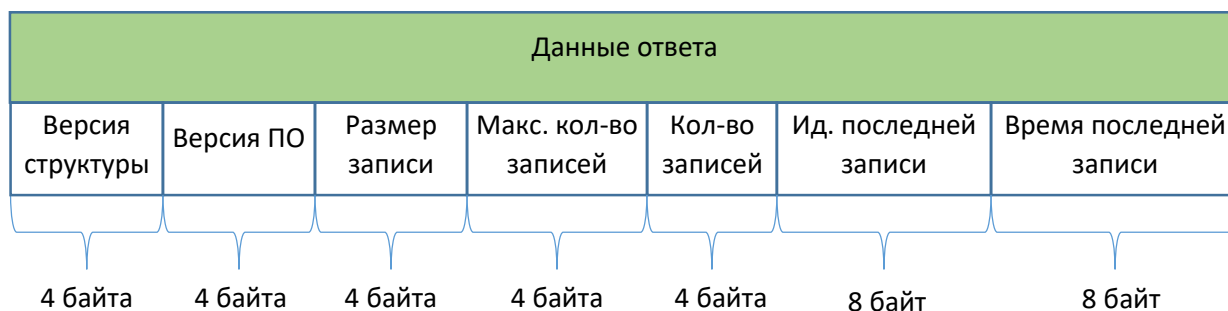


Рисунок 20. Поле данных положительного ответа на запрос чтения информации о конкретном журнале

Для информации об осциллограммах поле данных положительного ответа (см. рис. 21) содержит общее количество осциллограмм, а также время создания, длительность (в миллисекундах), размер файла и причину (тип и идентификатор сигнала) создания последней осциллограммы.



Рисунок 21. Поле данных положительного ответа на запрос информации об осциллограммах

Для информации о заполнении флеш-памяти поле данных положительного ответа (см. рис. 21) содержит общий объем флеш-памяти устройства, и объем свободной памяти в байтах.

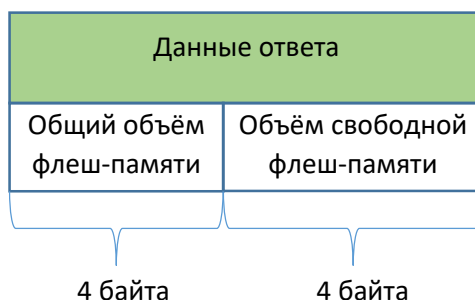


Рисунок 22. Поле данных положительного ответа на запрос информации о заполнении флеш-памяти

Для вида информации, обозначающего всю информацию, поле данных положительного ответа (см. рис. 23) содержит информацию об осциллограммах и обо всех журналах, при этом структура информации об осциллограммах и о каждом журнале такая же, как в ответе на запрос конкретного вида информации (см. рис. 20 – 22).

Данные ответа					
Инф. о журнале сигнализации	Инф. о системном журнале	Инф. о журнале изменений уставок	Инф. об осциллограммах	Инф. о заполнении флеш-памяти	Инф. о журнале аналоговых величин

Рисунок 23. Поле данных положительного ответа на запрос чтения всех видов информации

4.1.12. Чтение записей журнала (подфункция 25)

Поле данных запроса содержит код типа журнала (см. табл. 9), номер начальной записи и количество записей. Нумерация записей начинается с нуля, запись с номером 0 – самая новая запись в журнале. В качестве кода типа журнала можно использовать любое значение из табл. 9, кроме того, которое обозначает все журналы.



Рисунок 24. Структура запроса чтения записей журнала

Поле данных положительного ответа на этот запрос содержит номер начальной записи, количество записей и последовательность записей. Каждая запись в этой последовательности имеет структуру, определяется типом журнала и версией структуры (см. табл. 10, 14, 16, 18).

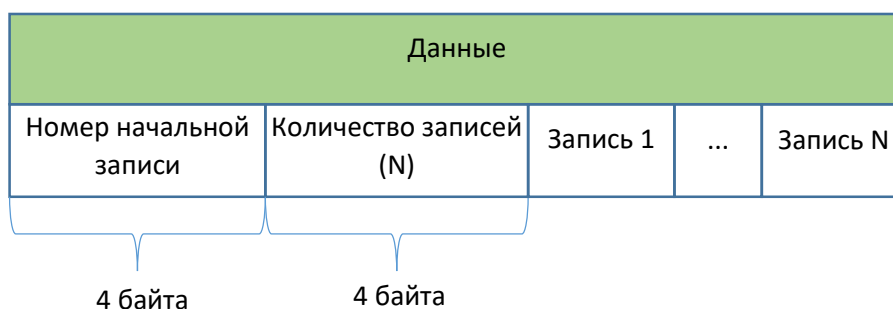


Рисунок 25. Поле данных положительного ответа на запрос чтения записей журнала

Таблица 10. Структура записи журнала сигнализации

Смещение, байт	Размер, байт	Условное обозначение	Описание
0	8	RecordNumber	Сквозной порядковый номер записи во всех журналах (нумерация начинается с 0).
8	8	Timestamp	Метка времени с точностью до миллисекунд (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC).
16	12	EmergencySignals	Битовый массив состояний аварийных сигналов (см. табл. 11).
28	12	PreventiveSignals	Битовый массив состояний предупредительных сигналов (см. табл. 11).
40	6	FaultSignals	Битовый массив состояний сигналов о неисправностях (см. табл. 12).
46	268	Parameters	Дополнительные параметры (см. табл. 13). Значения параметров хранятся в формате float32 (IEEE 754).

Таблица 11. Соответствие битов сигналам

Номер бита	Сигнал
0	АБ: глубокий разряд
1	АБ: высокое напряжение
2	АБ: высокие пульсации U
3	АБ: отсутствие термокомпенсации
4	АБ: высокий ток заряда
5	АБ: низкий ток заряда
6	АБ: высокие пульсации I
7	Шкаф: низкая температура
8	Шкаф: высокая температура 1 ст.
9	Шкаф: высокая температура 2 ст.
10	Помещение: низкая температура
11	Помещение: высокая температура
12 – 31	Акм №i: ускоренный износ (бит 12 соответствует 1-му аккумулятору)
32 – 51	Акм №i: деградация (бит 32 соответствует 1-му аккумулятору)
52 – 71	Акм №i: короткое замыкание (бит 52 соответствует 1-му аккумулятору)
72 – 91	Акм №i: тепловой разгон (бит 72 соответствует 1-му аккумулятору)
92 – 95	Зарезервированы

Таблица 12. Соответствие битов сигналам о неисправностях

Номер бита	Сигнал
0 – 19	Акм №i: потеря связи (бит 0 соответствует 1-му аккумулятору)
20 – 39	Акм №i: неисправность датчика (бит 20 соответствует 1-му аккумулятору)
40	РЕПЕЙ: неисправность
41 – 47	Зарезервированы

Таблица 13. Дополнительные параметры

Смещение, байт	Размер, байт	Параметр
0	4	Uаб
4	4	Iаб
8	4	кп Uаб
12	4	кп Iаб
16	4	Uпод
20	4	Tшкаф
24	4	Tпомещ
28	4 * 20	Uа (значения для 20 аккумуляторов)
108	4 * 20	Tа (значения для 20 аккумуляторов)

Таблица 14. Структура записи системного журнала

Смещение, байт	Размер, байт	Условное обозначение	Описание
0	8	RecordNumber	Сквозной порядковый номер записи во всех журналах (нумерация начинается с 0).
8	8	Timestamp	Метка времени с точностью до миллисекунд (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC).
16	1	SCADAAccessLevelID	Идентификатор уровня доступа от АСУ (см. табл. 15).
17	1	KIWIAccessLevelID	Идентификатор уровня доступа от Конфигуратора (см. табл. 15).
18	1	CPAccessLevelID	Идентификатор уровня доступа от пульта (см. табл. 15).
19	Размер записи - 19	Text	Текст сообщения, оканчивающийся нулём.

Таблица 15. Идентификаторы уровней доступа пользователя в журналах

Код	Описание
30	Полный доступ
60	Инженер РЗА
90	Наладчик РЗА
120	Инженер АСУ
150	Управление из АСУ
180	Настройка из АСУ

Таблица 16. Структура записи журнала изменений уставок

Смещение, байт	Размер, байт	Условное обозначение	Описание
0	8	RecordNumber	Сквозной порядковый номер записи во всех журналах (нумерация начинается с 0).
8	8	Timestamp	Метка времени с точностью до миллисекунд (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC).
16	2	SettingID	Идентификатор изменённой уставки.
18	1	AccessLevelID	Идентификатор уровня доступа пользователя, изменившего уставку (см. табл. 15).
19	1	SourceID	Идентификатор источника изменения уставки (см. табл. 17).
20	4	OldValue	Старое значение уставки (целое число со знаком).
24	4	NewValue	Новое значение уставки (целое число со знаком).
28	1	Program	Номер программы уставок (0 означает, что значение уставки не зависит от выбранной программы уставок).

Таблица 17. Идентификаторы источников изменений уставок и сигналов

Код	Описание
0	Базовая станция «Репей»
2	Конфигуратор
3	АСУ
4	Двоичный вход (только для сигналов)

Каждая запись журнала аналоговых величин (табл. 18) содержит значения определённых величин в указанные моменты времени.

Таблица 18. Структура записи журнала аналоговых величин

Смещение, байт	Размер, байт	Условное обозначение	Описание
0	8	RecordNumber	Сквозной порядковый номер записи во всех журналах (нумерация начинается с 0).
8	8	Timestamp	Метка времени с точностью до миллисекунд (уточнённое время UNIX – количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC).
16	256	Values	Массив значений аналоговых величин (см. табл. 19). Значения величин хранятся в формате float32 (IEEE 754).

Таблица 19. Массив значений аналоговых величин

Смещение, байт	Размер, байт	Описание
0	4	Uаб, напряжение на АБ
4	4	Iаб, ток АБ
8	4	Tшкаф, температура внутри шкафа
12	4	Tпомещ, температура внутри помещения
16	4	Ta ср, средняя температура аккумуляторов
20	4 * 20	Uа, напряжение аккумулятора (значения для 20 аккумуляторов)
100	4 * 20	Ta, температура аккумулятора (значения для 20 аккумуляторов)

4.1.13. Формирование файла, содержащего записи журнала за указанный период (подфункция 27)

Поле данных запроса содержит код типа журнала (см. табл. 9), начало и конец периода как количество полных миллисекунд, прошедших с 00:00:00 01.01.1970 UTC. В качестве кода типа журнала можно использовать любое значение из табл. 9, кроме того, которое обозначает все журналы.



Рисунок 26. Структура запроса формирования файла, содержащего записи журнала за указанный период

Поле данных положительного ответа содержит имя сформированного файла. Этот файл имеет такую же структуру, как и файл соответствующего журнала в устройстве. Записи в этом файле расположены в хронологическом порядке, т.е. в порядке возрастания сквозного порядкового номера записи (RecordNumber).

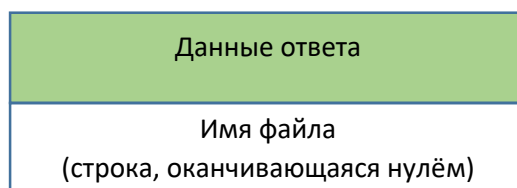


Рисунок 27. Поле данных положительного ответа на запрос формирования файла, содержащего записи журнала за указанный период

4.1.14. Открытие файла (подфункция 31)

Поле данных запроса содержит путь к файлу на сервере и способ открытия файла (табл. 20).

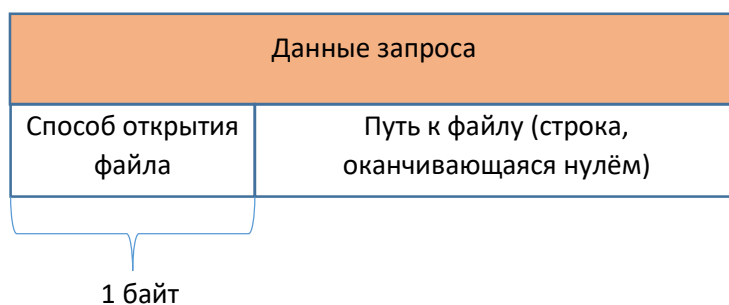


Рисунок 28. Структура запроса открытия файла

Таблица 20. Коды способов открытия файла

Код способа	Описание
0	Создать файл, если его нет или открыть для записи и удалить содержимое, если файл есть.
1	Создать файл, если его нет или открыть для записи и чтения, если он есть.
2	Открыть файл только для чтения.

В положительном ответе на этот запрос клиенту возвращается идентификатор открытого файла, размер и время изменения файла. Идентификатор должен использоваться при отправке других запросов для доступа к этому файлу. Время изменения представляет собой уточнённое время UNIX с точностью до миллисекунд.

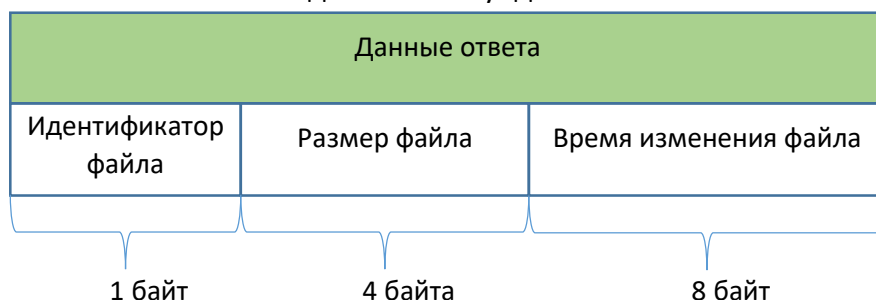


Рисунок 29. Поле данных положительного ответа на запрос открытия файла

4.1.15. Чтение файла (подфункция 33)

Поле данных запроса содержит идентификатор файла, полученный при его открытии, байтовое смещение читаемого фрагмента файла и длину фрагмента.



Рисунок 30. Структура запроса чтения файла

В поле данных положительного ответа находится идентификатор файла, смещение прочитанного фрагмента файла и сам фрагмент.

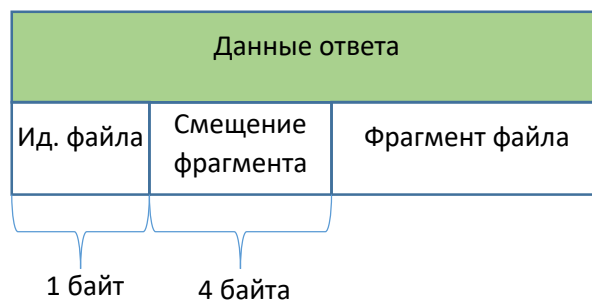


Рисунок 31. Поле данных положительного ответа на запрос чтения файла

4.1.16. Заккрытие файла (подфункция 34)

Поле данных запроса содержит идентификатор файла.

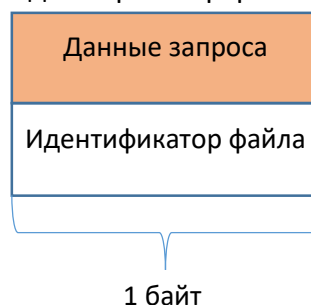


Рисунок 32. Структура запроса закрытия файла

В положительном ответе на этот запрос поле данных отсутствует.

4.1.17. Передача длинной команды (подфункция 239)

Эта подфункция предназначена для передачи серверу команд, длина которых превышает стандартный размер MODBUS PDU.

Поле данных запроса содержит идентификатор команды, назначенный клиентом, общую длину команды, байтовое смещение очередного фрагмента команды и сам фрагмент команды, начинающийся с указанного смещения.

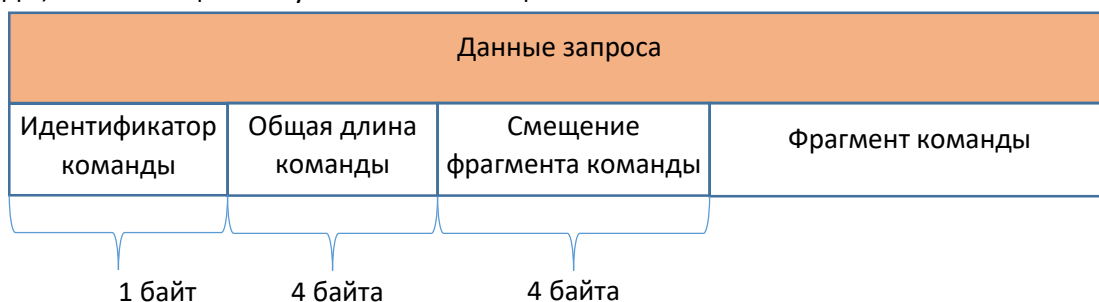


Рисунок 33. Структура запроса передачи длинной команды

В поле данных положительного ответа всегда содержится идентификатор полученной команды и тип результата (0 – подтверждение очередного фрагмента команды, 1 – результат исполнения команды). Если поле «Тип результата» содержит 0 (рис. 34), то за ним следует 32-битовое смещение следующего фрагмента команды. Если же поле «Тип результата» содержит 1 (рис. 35), то за ним следует результат исполнения команды, оформленный как ответ на запрос (MODBUS PDU).

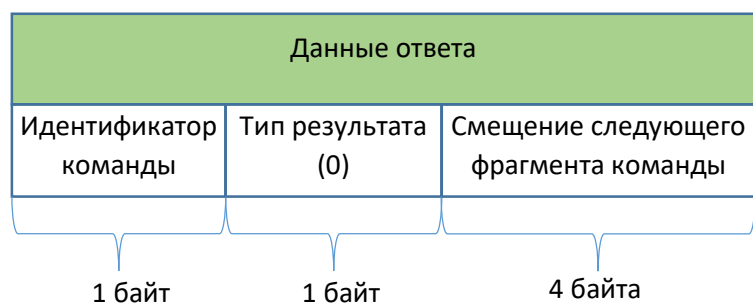


Рисунок 34. Поле данных положительного ответа на запрос передачи длинного запроса с типом результата 0

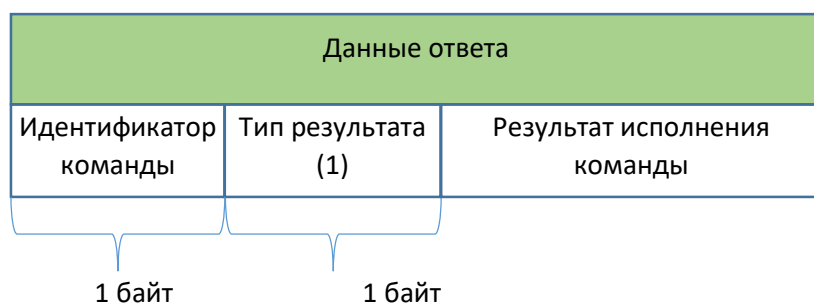


Рисунок 35. Поле данных положительного ответа на запрос передачи длинной команды с типом результата 1

4.1.18. Получение информации о буфере сервера (подфункция 240)

В запросе поле данных отсутствует.

Положительные ответы на этот запрос всегда короткие (т.е. тип ответа равен 0). Поле данных короткого положительного ответа на этот запрос содержит объём данных, находящихся в буфере сервера.

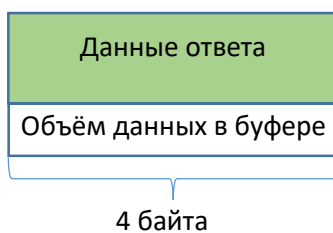


Рисунок 36. Поле данных положительного ответа на запрос информации о буфере сервера

4.1.19. Чтение данных из буфера сервера (подфункция 241)

Поле данных запроса содержит байтовое смещение от начала буфера и количество байтов для чтения.

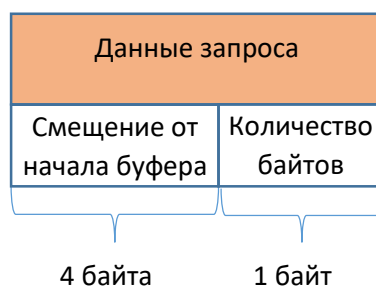


Рисунок 37. Структура запроса чтения данных из буфера сервера

Положительные ответы на этот запрос всегда короткие (т.е. тип ответа равен 0). Поле данных короткого положительного ответа содержит данные, прочитанные из буфера сервера.

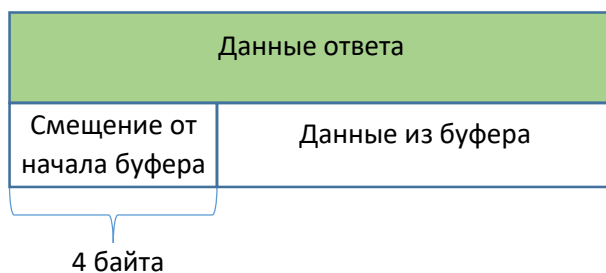


Рисунок 38. Поле данных положительного ответа на запрос чтения данных из буфера сервера

4.1.20. Уведомление сервера о завершении чтения из буфера (подфункция 242)

В запросе поле данных отсутствует.

В положительном ответе на этот запрос поле данных отсутствует.

5. Приложение 1. Пример использования запроса передачи длинной команды (подфункция 239)

Рассмотрим использование запроса передачи длинной команды на примере записи уставок. Предположим, что нужно записать значения 5 уставок с идентификаторами от 1 до 5 для двух программ. Для записи уставок используется подфункция 11 функции 65, поэтому команда записи уставок будет иметь вид, показанный в табл. 21. Структура поля данных этой команды (байты, начиная со смещения 4 в табл. 21) такая же, как и структура данных запроса записи уставок (см. рис. 1111). В данном случае размер поля данных команды составляет 66 байт, общая длина команды – 70 байт.

Таблица 21. Пример команды записи уставок

Смещение, байт	Размер, байт	Значение (шестн.)	Описание
0	1	41	Код функции (65)
1	1	0B	Код подфункции (11)
2	1	1E	Номер запроса (30)
3	1	42	Длина данных (66)
4	1	01	Номер программы уставок (1)
5	2	00 05	Количество уставок (5)
7	2	00 01	Идентификатор уставки (1)
9	4	00 00 02 01	Значение уставки (513)
13	2	00 02	Идентификатор уставки (2)
15	4	00 00 00 21	Значение уставки (33)
19	2	00 03	Идентификатор уставки (3)
21	4	00 00 00 22	Значение уставки (34)
25	2	00 04	Идентификатор уставки (4)
27	4	00 00 00 00	Значение уставки (0)
31	2	00 05	Идентификатор уставки (5)
33	4	FF FF FF FE	Значение уставки (-2)
37	1	02	Номер программы уставок (2)
38	2	00 05	Количество уставок (5)
40	2	00 01	Идентификатор уставки (1)
42	4	00 00 02 00	Значение уставки (512)
46	2	00 02	Идентификатор уставки (2)
48	4	00 00 00 20	Значение уставки (32)
52	2	00 03	Идентификатор уставки (3)
54	4	00 00 00 14	Значение уставки (20)
58	2	00 04	Идентификатор уставки (4)
60	4	00 00 00 01	Значение уставки (1)
64	2	00 05	Идентификатор уставки (5)
66	4	FF FF FF FD	Значение уставки (-3)

Для простоты предположим, что максимально допустимый размер MODBUS PDU составляет не 253, а 63 байта, тогда для передачи этой команды записи уставок потребуется 2 запроса с подфункцией 239. В первом запросе (табл. 22) передаются первые 50 байт команды, начиная со смещения 0. Положительный ответ на первый запрос (табл. 23) имеет тип результата 0 и указывает смещение следующего фрагмента команды (50). Во втором запросе (табл. 24) передаются последние 20 байт команды. Если при получении запроса

возникла ошибка, то сервер отправляет отрицательный ответ на подфункцию 239 (табл. 25). Если все фрагменты команды получены успешно, то сразу же после получения последнего фрагмента сервер исполняет команду и, если она исполнена успешно, то сервер отправляет положительный ответ на подфункцию 239 с типом результата 1 и положительным ответом на команду (табл. 26). Если при исполнении команды возникла ошибка, то отправляется положительный ответ на подфункцию 239 с типом результата 1 и отрицательным ответом на команду (табл. 27).

Таблица 22. MODBUS PDU 1-го запроса, передающего команду записи уставок

Смещение, байт	Размер, байт	Значение (шестн.)	Описание
0	1	41	Код функции (65)
1	1	EF	Код подфункции (239)
2	1	11	Номер запроса (17)
3	1	3B	Длина данных (59)
4	1	2B	Идентификатор команды (43)
5	4	00 00 00 46	Общая длина команды (70)
9	4	00 00 00 00	Смещение фрагмента команды (0)
13	1	41	Код функции (65)
14	1	0B	Код подфункции (11)
15	1	1E	Номер запроса (30)
16	1	42	Длина данных (66)
17	1	01	Номер программы уставок (1)
18	2	00 05	Количество уставок (5)
20	2	00 01	Идентификатор уставки (1)
22	4	00 00 02 01	Значение уставки (513)
26	2	00 02	Идентификатор уставки (2)
28	4	00 00 00 21	Значение уставки (33)
32	2	00 03	Идентификатор уставки (3)
34	4	00 00 00 22	Значение уставки (34)
38	2	00 04	Идентификатор уставки (4)
40	4	00 00 00 00	Значение уставки (0)
44	2	00 05	Идентификатор уставки (5)
46	4	FF FF FF FE	Значение уставки (-2)
50	1	02	Номер программы уставок (2)
51	2	00 05	Количество уставок (5)
53	2	00 01	Идентификатор уставки (1)
55	4	00 00 02 00	Значение уставки (512)
59	2	00 02	Идентификатор уставки (2)
61	2	00 00	Два старших байта значения уставки

Таблица 23. MODBUS PDU положительного ответа на 1-й запрос

Смещение, байт	Размер, байт	Значение (шестн.)	Описание
0	1	41	Функция (65)
1	1	EF	Подфункция (239)
2	1	11	Номер запроса (17)
3	1	07	Длина данных (7)
4	1	00	Тип ответа (0), код ответа (0)
5	1	2B	Идентификатор команды (43)
6	1	00	Тип результата (0)
7	4	00 00 00 32	Смещение следующего фрагмента команды (50)

Таблица 24. MODBUS PDU 2-го запроса, передающего команду записи уставок

Смещение, байт	Размер, байт	Значение (шестн.)	Описание
0	1	41	Код функции (65)
1	1	EF	Код подфункции (239)
2	1	12	Номер запроса (18)
3	1	1D	Длина данных (29)
4	1	2B	Идентификатор команды (43)
5	4	00 00 00 46	Общая длина команды (70)
9	4	00 00 00 32	Смещение фрагмента команды (50)
13	2	00 20	Два младших байта значения уставки
15	2	00 03	Идентификатор уставки (3)
17	4	00 00 00 14	Значение уставки (20)
21	2	00 04	Идентификатор уставки (4)
23	4	00 00 00 01	Значение уставки (1)
27	2	00 05	Идентификатор уставки (5)
29	4	FF FF FF FD	Значение уставки (-3)

Таблица 25. MODBUS PDU отрицательного ответа на 2-й запрос

Смещение, байт	Размер, байт	Значение (шестн.)	Описание
0	1	41	Функция (65)
1	1	EF	Подфункция (239)
2	1	11	Номер запроса (18)
3	1	01	Длина данных (1)
4	1	0D	Тип ответа (0), код ответа (13)

Таблица 26. MODBUS PDU положительного ответа на 2-й запрос и положительным ответом на команду

Смещение, байт	Размер, байт	Значение (шестн.)	Описание
0	1	41	Функция (65)
1	1	EF	Подфункция (239)
2	1	11	Номер запроса (18)
3	1	08	Длина данных (8)
4	1	00	Тип ответа (0), код ответа (0)
5	1	2B	Идентификатор команды (43)
6	1	01	Тип результата (1)
7	1	41	Код функции (65)
8	1	0B	Код подфункции (11)
9	1	1E	Номер запроса (30)
10	1	01	Длина данных (1)
11	1	00	Тип ответа (0), код ответа (0)

Таблица 27. MODBUS PDU положительного ответа на 2-й запрос и отрицательным ответом на команду

Смещение, байт	Размер, байт	Значение (шестн.)	Описание
0	1	41	Функция (65)
1	1	EF	Подфункция (239)
2	1	11	Номер запроса (18)
3	1	08	Длина данных (8)
4	1	00	Тип ответа (0), код ответа (0)
5	1	2B	Идентификатор команды (43)
6	1	01	Тип результата (1)
7	1	41	Код функции (65)
8	1	0B	Код подфункции (11)
9	1	1E	Номер запроса (30)
10	1	01	Длина данных (1)
11	1	0C	Тип ответа (0), код ответа (12)

6. Литература

1. MODBUS over Serial Line Specification and Implementation Guide V1.02; www.modbus.org; December 20, 2006.
2. MODBUS Application Protocol Specification V1.1b3; www.modbus.org; April 26, 2012.
3. ISO/IEC 8482:1993. Information technology – Telecommunications and information exchange between systems – Twisted pair multipoint interconnections.