АННОТАЦИЯ

В данном документе приведено краткое описание протокола MODBUS/RTU с передачей данных по последовательному интерфейсу.

ПРОТОКОЛ ПЕРЕДАЧИ ДАННЫХ MODBUS/RTU

Модель OSI протокола ModBus/RTU представлена на рисунке 1.

Уровень	Модель ISO/OSI	
7	Прикладной уровень	Прикладной уровень ModBus
6	Представительский уровень	---
5	Сеансовый уровень	---
4	Транспортный уровень	---
3	Сетевой уровень	---
2	Канальный уровень	Протокол последовательного интерфейса ModBus
1	Физический уровень	EIA/TIA-485 (RS-485) или EIA/TIA-232 (RS-232)

Рисунок 1 – Модель ISO/OSI протокола ModBus/RTU

1 Физический уровень

В качестве среды передачи данных используется двухпроводный (полудуплексный) или четырехпроводный (дуплексный) дифференциальный интерфейс TIA/EIA-485 (TIA/EIA-422). Требования к параметрам среды передачи данных приведены в стандарте ANSI/TIA/EIA-485-A-98.

2 Канальный уровень

Канальный уровень обеспечивает создание, передачу и прием кадров данных. Этот уровень обслуживает запросы сетевого уровня и использует сервис физического уровня для приема и передачи пакетов.

Протокол MobBus является протоколом типа “ведущий-ведомый”, т.е. в одно и то же время к шине подключено может быть только одно ведущее устройство (мастер) и один или несколько (до 247) ведомых устройств (слейвы). Передача данных инициируется всегда ведущим устройством. Ведомые устройства могут отвечать отвечают только на запросы ведущего.

Ведущее устройство единовременно может инициировать запросы к конкретному ведомому устройству (unicast mode) или всем ведомым устройствам (broadcast mode – широковещательный запрос). Ведомые устройства сети не отвечают на широковещательные запросы, а только принимают их. Для передачи широковещательных запросов используется адрес 0.

Протокол ModBus предполагает использование адресов ведомых устройств в диапазоне 1-247. Каждое устройство в сети должно иметь уникальный адрес.

Формат данных протокола ModBus/RTU представлен на рисунке 2.

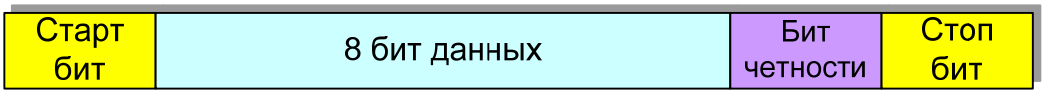

Рисунок 2 – Формат данных

Посылка каждого байта начинается со старт-бита, после которого следуют 8 бит данных, бит четности (even) и стоп бит. Таким образом, одна посылка данных состоит из 11 бит.

Для согласования со сторонними изделиями, возможна работа без бита четности, при этом должны использоваться два стоп-бита, как указано на рисунке 3.

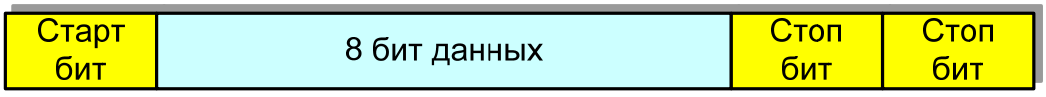

Рисунок 3 – Альтернативный формат данных

Обмен данными по протоколу производится фреймами пакетами (данных). Структура фрейма приведена на рисунке 4.

Рисунок 4 – Структура фрейма

Фрейм начинается с посылки адрес устройства, к которому отправляется запрос (или адрес устройства, которое формирует ответ). Диапазон возможных значений адресов: 0–247. Адрес 0 (нулевой) является широкополосным и предназначен для передачи информации всем устройствам в сети. Запрос с нулевым адресом устройства не предполагает ответа.

После передачи адреса следует байт функции, определяющий функциональную принадлежность запроса (ответа). Диапазон возможных значений: 0 – 255.

После передачи Функции следует передача данных. Передача данных осуществляется побайтно. Количество передаваемых байт – 0…252.

После передачи данных следует два байта контрольной суммы, предназначенных для проверки достоверности принимаемой информации.

В соответствии с протоколом ModBus/RTU, длина фрейма может быть переменной, не более 256 байт. Передача байт данных в пределах фрейма производится последовательно с промежутком времени между передачей не более 1,5 времени передачи одного байта данных.

В протоколе используется повременная синхронизация начала и завершения передачи.

Диаграмма передачи фреймов приведена на рисунке 5.

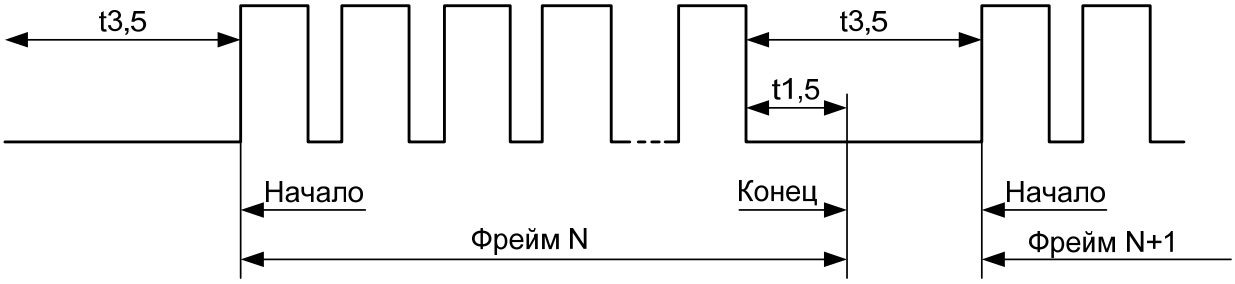

Рисунок 5 – Диаграмма передачи фреймов

Перед началом передачи очередного фрейма, необходима выдержка времени, соответствующая 3,5 временам передачи одного байта данных (t3,5) после завершения передачи предыдущего фрейма (или “ложной” передачи данных).

Завершение передачи фрейма является отсутствие передачи данных в течении 1,5 времени передачи одного байта данных (t1,5). Однако, если по истечении времени t1,5 в течение времени t3,5 возобновится передача данных, то фрейм считается недостоверным.

Все устройства в сети должны иметь один формат передачи данных и одну скорость передачи данных. Рекомендуемая скорость передачи данных - 19,2 кБит/с. Допускается передача данных на скоростях 9,6 кБит/с, 57,6 кБит/с, 115,2 кБит/с.

Для определения достоверности принимаемых данных используются:

- контроль бита четности при передаче каждого байта (аппаратная функция приемо-передатчика);

- подсчет и сравнение контрольной суммы CRC (Cyclical Redundancy Checking) при передаче фрейма.

Контрольная сумма состоит из 2-х байт в формате [MSB(старший байт)|LSB(младший байт)].

Контрольная сумма подсчитывается и добавляется в конец фрейма передающим устройством, и сравнивается принимающим устройством с контрольной суммой, подсчитанной им по принятым данным.

В подсчете контрольной суммы используются все байты фрейма, начиная с первого (адреса).

Подсчет контрольной суммы производится следующим образом:

1) Записывается в 16-ти битный регистр CRC число 0xFFFF.

2) Первому байту данных и регистру CRC применяется функция XOR, результат помещается в CRC регистр;

3) Регистр CRC сдвигается вправо на 1 бит, старший бит CRC регистра устанавливается в 0. Проверяется сдвинутый бит CRC регистра.

4) Если сдвинутый бит CRC регистра равен 1, то CRC регистру и полиномиальному числу (например 0xA001) применяется функция XOR;

5) Выполняются пункты 3,4, пока не будет выполнено 8 сдвигов CRC регистра;

6) Следующему байту данных и регистру CRC применяется функция XOR, результат помещается в CRC регистр;

6) Повторяются действия по пунктам 3-6 для оставшихся данных.

7) Контрольная сумма передается в фрейме младшим байтом вперед, т.е. в формате LSB|MSB.

Возможна также табличная форма подсчета контрольной суммы, что значительно ускоряет процесс подсчета.

Функция подсчета CRC16 на языке си имеет вид:

```
unsigned short CRC16 ( puchMsg, usDataLen )
unsigned char *puchMsg ;
unsigned short usDataLen ;
{
unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
unsigned uIndex ; /* will index into CRC lookup table */
while (usDataLen--) /* pass through message buffer */
{
uIndex = uchCRCLo ^ *puchMsgg++ ; /* calculate the CRC */
uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex} ;
uchCRCHi = auchCRCLo[uIndex] ;
}
return (uchCRCHi << 8 | uchCRCLo) ;
}
```

```
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80,
0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40} ;
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40};
```

3 Прикладной уровень

Перечень используемых функций протокола ModBus/RTU приведены в таблице 1. Весь перечень функций приведен в MODBUS Application Protocol Specification V1.1b3.

Тип адресации	Описание функции	Код функции (hex)
Битовая адресация	Чтение дискретных входов	0x2
	Чтение состояния релейных выходов	0x1
	Запись состояния одного релейного выхода	0x5
	Запись состояния нескольких релейных выходов	0xF
16-битная адресация	Чтение регистров данных	0x4
	Чтение регистров параметров	0x3
	Запись одного регистра параметров	0x6
	Запись одного нескольких регистров параметров	0x10
Работа с файловыми записями	Чтение данных файла	0x14
	Запись данных файла	0x15
Диагностические данные	Чтение ID (серийного номера)	0x11

0x2 - Чтение дискретных входов

Эта функция используется для чтения от 1 до 2000 дискретных входов. Нумерация дискретных входов начинается с нуля. Значения битов: 1 – соответствует замкнутому состоянию входа, 0 – разомкнутому.

Запрос

Функция	1 байт	0x2
Начальный адрес	2 байта	0x0-0xFFFF
Количество входов для чтения	2 байта	1-2000 (0x7D0)

Ответ

Функция	1 байт	0x2
Число байт	1 байт	N*
Байты состояния входов	N* x 1 байт	

Ошибка

Функция	1 байт	0x82
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

N* = [Количество входов для чтения]/8, если остаток > 0, то принимается N=N+1

Пример запроса состояния дискретных входов 197-218 устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x2	**Функция**	0x2
Начальный адрес (старший байт)	0x0	**Число байт**	0x3
Начальный адрес (младший байт)	0xC4	**Состояния входов 204-197**	0xAC

Количество входов (старший байт)	0x0	**Состояния входов 212-205**	0xDB
Количество входов (младший байт)	0x16	**Состояния входов 218-213**	0x35

0x1 - Чтение состояния релейных выходов

Эта функция используется для чтения от 1 до 2000 релейных выходов.

Нумерация релейных выходов начинается с нуля. Значения битов: 1 – соответствует включенному состоянию выхода, 0 – выключенному.

Запрос

Функция	1 байт	0x1
Начальный адрес	2 байта	0x0-0xFFFF
Количество выходов для чтения	2 байта	1-2000 (0x7D0)

Ответ

Функция	1 байт	0x1
Число байт	1 байт	N*
Байты состояния выходов	n x 1 байт	n=N* или n=N+1

Ошибка

Функция	1 байт	0x81
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

N* = [Количество выходов для чтения]/8, если остаток > 0, то принимается N=N+1

Пример запроса состояния релейных выходов 20-38 устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x1	**Функция**	0x1
Начальный адрес (старший байт)	0x0	**Число байт**	0x3
Начальный адрес (младший байт)	0x13	**Состояния выходов 27-20**	0xCD
Количество входов (старший байт)	0x0	**Состояния выходов 35-28**	0x6B
Количество входов (младший байт)	0x13	**Состояния выходов 38-36**	0x05

0x5 - Запись состояния одного релейного выхода

Эта функция используется для записи состояния одного релейного выхода по выбранному адресу.

Нумерация релейных выходов начинается с нуля. Значения регистров: 0xFF00 – соответствует включенному состоянию выхода, 0x00FF – выключенному.

При успешном выполнении команды в ответ устройство присылает копию запроса.

Запрос

Функция	1 байт	0x5
Адрес выхода	2 байта	0x0-0xFFFF
Значение для записи	2 байта	0xFF00 или 0x00FF

Ответ

Функция	1 байт	0x5
Адрес выхода	2 байта	0x0-0xFFFF
Значение для записи	2 байта	0xFF00 или 0x00FF

Ошибка

Функция	1 байт	0x85
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

Пример включения релейного выхода 173 устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x5	**Функция**	0x5
Адрес выхода (старший байт)	0x0	**Адрес выхода (старший байт)**	0x0
Адрес выхода (младший байт)	0xAC	**Адрес выхода (младший байт)**	0xAC
Значение выхода (старший байт)	0xFF	**Значение выхода (старший байт)**	0xFF
Значение выхода (младший байт)	0x00	**Значение выхода (младший байт)**	0x00

0xF - Запись состояния нескольких релейных выходов

Эта функция используется для записи состояния нескольких релейных выходов выбранного диапазона адресов.

Нумерация релейных выходов начинается с нуля. Значения битов: 1 – соответствует включенному состоянию выхода, 0 – выключенному.

При успешном выполнении команды ответ имеет формат: функция, начальный адрес записи, число записанных релейных выходов.

Запрос

Функция	1 байт	0xF
Начальный адрес	2 байта	0x0-0xFFFF
Число выходов для записи	2 байта	1-2000 (0x07B0)
Число байт	1 байт	N*
Значение для записи	N* x 1 байт	

N* = [Количество выходов для чтения]/8, если остаток > 0, то принимается N=N+1

Ответ

Функция	1 байт	0xF
Начальный адрес	2 байта	0x0-0xFFFF
Число записанных релейных выходов	2 байта	1-2000 (0x07B0)

Ошибка

Функция	1 байт	0x8F
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

Пример записи состояния релейных выходов 20-30 устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0xF	**Функция**	0xF
Начальный адрес (старший байт)	0x0	**Начальный адрес (старший байт)**	0x0
Начальный адрес (младший байт)	0x13	**Начальный адрес (младший байт)**	0x13
Количество выходов (старший байт)	0x0	**Количество выходов (старший байт)**	0x0
Количество выходов (младший байт)	0xA	**Количество выходов (младший байт)**	0xA
Число байт	0x2		
Значение (старший байт)	0xCD		
Значение (младший байт)	0x1		

0x4 - Чтение регистров данных

Эта функция используется для последовательного чтения от 1 до 125 регистров данных. Нумерация регистров начинается с нуля. Регистры 16-ти битные, беззнаковые или знаковые (в дополнительном коде). 32-х битные регистры разбиваются на два 16-ти битных слова в формате [HIword, LOWword]

Запрос

Функция	1 байт	0x4
Начальный адрес	2 байта	0x0-0xFFFF
Количество регистров для чтения	2 байта	1-125 (0x7D)

Ответ

Функция	1 байт	0x4
Число байт	1 байт	2 x N*
Байты регистров	N* x 2 байт	

N - Количество регистров для чтения

Ошибка

Функция	1 байт	0x84
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

Пример чтения регистра 9 устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x4	**Функция**	0x4
Начальный адрес (старший байт)	0x0	**Число байт**	0x2
Начальный адрес (младший байт)	0x8	**Значение регистра (старший байт)**	0x0

Количество регистров (старший байт)	0x0	**Значение регистра (младший байт)**	0xA
Количество регистров (младший байт)	0x1		

0x3 - Чтение регистров параметров

Эта функция используется для последовательного чтения от 1 до 125 регистров параметров. Нумерация регистров начинается с нуля. Регистры 16-ти битные, беззнаковые или знаковые (в дополнительном коде). 32-х битные регистры разбиваются на два 16-ти битных слова в формате [HIword, LOWword]

Запрос

Функция	1 байт	0x3
Начальный адрес	2 байта	0x0-0xFFFF
Количество регистров для чтения	2 байта	1-125 (0x7D)

Ответ

Функция	1 байт	0x3
Число байт	1 байт	2 x N*
Байты регистров	N* x 2 байт	

N - Количество регистров для чтения

Ошибка

Функция	1 байт	0x83
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

Пример чтения регистров 108-110 устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x3	**Функция**	0x3
Начальный адрес (старший байт)	0x0	**Число байт**	0x6
Начальный адрес (младший байт)	0x6B	**Значение регистра 108 (старший байт)**	0x2
Количество регистров (старший байт)	0x0	**Значение регистра 108 (младший байт)**	0x2B
Количество регистров (младший байт)	0x3	**Значение регистра 109 (старший байт)**	0x0
		Значение регистра 109 (младший байт)	0x0
		Значение регистра 110 (старший байт)	0x0
		Значение регистра 110 (младший байт)	0x64

0x6 – Запись одного регистра параметров

Эта функция используется для записи одного регистра параметров.

Нумерация регистров начинается с нуля. Регистры 16-ти битные, беззнаковые или знаковые (в дополнительном коде).

При успешном выполнении команды в ответ устройство присылает копию запроса.

Запрос

Функция	1 байт	0x6
Адрес регистра	2 байта	0x0-0xFFFF
Значение для записи	2 байта	0x0-0xFFFF

Ответ

Функция	1 байт	0x6
Адрес регистра	2 байта	0x0-0xFFFF
Значение для записи	2 байта	0x0-0xFFFF

Ошибка

Функция	1 байт	0x86
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

Пример записи числа 3 в регистр 2 устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x6	**Функция**	0x6
Адрес регистра (старший байт)	0x0	**Адрес регистра (старший байт)**	0x0
Адрес регистра (младший байт)	0x1	**Адрес регистра (младший байт)**	0x1
Значение для записи (старший байт)	0x0	**Значение для записи (старший байт)**	0x0
Значение для записи (младший байт)	0x3	**Значение для записи (младший байт)**	0x3

0x10 – Запись одного регистра параметров

Эта функция используется для записи 1-123 регистра параметров.

Нумерация регистров начинается с нуля. Регистры 16-ти битные, беззнаковые или знаковые (в дополнительном коде).

При успешном выполнении команды ответ имеет формат: функция, начальный адрес записи, число записанных регистров.

Запрос

Функция	1 байт	0x10
Адрес начала записи	2 байта	0x0-0xFFFF
Число регистров	2 байта	0x0-0x7B
Число байт	1 байт	2 x N*
Значения для записи	N* x 2 байт	0x0-0xFFFF…

Ответ

Функция	1 байт	0x10
Адрес начала записи	2 байта	0x0-0xFFFF
Число записанных регистров	2 байта	1-123(0x7B)

N - Количество регистров для записи

Ошибка

Функция	1 байт	0x90
Код ошибки	1 байт	0x1, 0x2, 0x3 или 0x4

Пример записи чисел 10 и 258 в два регистра, начиная с адреса 1, устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x10	**Функция**	0x10
Адрес начала записи (старший байт)	0x0	**Адрес начала записи (старший байт)**	0x0
Адрес начала записи (младший байт)	0x1	**Адрес начала записи (младший байт)**	0x1
Значение для записи (старший байт)	0x0	**Число записанных регистров (старший байт)**	0x0
Значение для записи (младший байт)	0xA	**Число записанных регистров (младший байт)**	0x2
Значение для записи (старший байт)	0x1		
Значение для записи (младший байт)	0x2		

0x14 – Чтение записи файла

Эта функция используется для чтение файлов.

Файл состоит из записей. Каждый файл может содержать до 1000 записей с адресацией от 0 до 9999.

Функция позволяет считывать несколько групп записей в одном запросе.

Длина считываемой записи должна быть выбрана такой, чтобы длина ответа не превысила 253 байта.

Запрос

Функция	1 байт	0x14
Число байт запроса	1 байт	0x7-0xF5
Группа 1. Тип запроса	1 байт	0x6
Группа 1. Номер файла	2 байта	0x1-0xFFFF
Группа 1. Номер записи	2 байта	0x0-0x270F
Группа 1. Длина записи	2 байта	N
Группа 2. Тип запроса	1 байт	0x6
Группа 2. Номер файла	2 байта	0x1-0xFFFF
Группа 2. Номер записи	2 байта	0x0-0x270F
Группа 2. Длина записи	2 байта	N

Ответ

Функция	1 байт	0x14
Число байт ответа	1 байт	0x7-0xF5
Группа 1. Длина записи	1 байт	0x7-0xF5
Группа 1. Тип запроса	1 байт	0x6
Группа 1. Данные	N x 2 байт	

Группа 2. Длина записи	1 байт	0x7-0xF5
Группа 2. Тип запроса	1 байт	0x6
Группа 2. Данные	N x 2 байт	

Ошибка

Функция	1 байт	0x94
Код ошибки	1 байта	0x1, 0x2, 0x3, 0x4 или 0x8

Пример чтения 2-х групп данных, устройства с адресом 0x1

- 2-х регистров из файла 4, начиная с регистра 1;
- 2-х регистров из файла 3, начиная с регистра 9:

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x14	**Функция**	0x14
Число байт	0xE	**Число байт**	0xC
Группа 1. Тип запроса	0x6	**Группа 1. Длина записи**	0x5
Группа 1. Номер файла (старший байт)	0x0	**Группа 1. Тип запроса**	0x6
Группа 1. Номер файла (младший байт)	0x4	**Группа 1. Данные 1 (старший байт)**	0xD
Группа 1. Номер записи (старший байт)	0x0	**Группа 1. Данные 1 (младший байт)**	0xFE
Группа 1. Номер записи (младший байт)	0x1	**Группа 1. Данные 2 (старший байт)**	0x0
Группа 1. Длина записи (старший байт)	0x0	**Группа 1. Данные 2 (младший байт)**	0x20
Группа 1. Длина записи (младший байт)	0x2	**Группа 1. Длина записи**	0x5
Группа 2. Тип запроса	0x6	**Группа 1. Тип запроса**	0x6
Группа 2. Номер файла (старший байт)	0x0	**Группа 1. Данные 1 (старший байт)**	0x33
Группа 2. Номер файла (младший байт)	0x3	**Группа 1. Данные 1 (младший байт)**	0xCD
Группа 1. Номер записи (старший байт)	0x0	**Группа 1. Данные 2 (старший байт)**	0x0
Группа 1. Номер записи (младший байт)	0x9	**Группа 1. Данные 2 (младший байт)**	0x40
Группа 1. Длина записи (старший байт)	0x0		
Группа 1. Длина записи (младший байт)	0x2		

0x15 – Запись записи файла

Эта функция используется для записи файлов.

Файл состоит из записей. Каждый файл может содержать до 1000 записей с адресацией от 0 до 9999.

Функция позволяет записывать несколько групп записей в одном запросе.

Длина записи должна быть выбрана такой, чтобы длина ответа не превысила 253 байта.

При успешном выполнении команды в ответ устройство присылает копию запроса.

Запрос

Функция	1 байт	0x15
Число байт запроса	1 байт	0x9-0xFB
Группа 1. Тип запроса	1 байт	0x6
Группа 1. Номер файла	2 байта	0x1-0xFFFF
Группа 1. Номер записи	2 байта	0x0-0x270F
Группа 1. Длина записи	2 байта	N
Группа 1. Данные	N x 2 байт	
Группа 2. Тип запроса	1 байт	0x6
Группа 2. Номер файла	2 байта	0x1-0xFFFF
Группа 2. Номер записи	2 байта	0x0-0x270F
Группа 2. Длина записи	2 байта	N
Группа 2. Данные	N x 2 байт	

Ответ

Функция	1 байт	0x15
Число байт запроса	1 байт	0x9-0xFB
Группа 1. Тип запроса	1 байт	0x6
Группа 1. Номер файла	2 байта	0x1-0xFFFF
Группа 1. Номер записи	2 байта	0x0-0x270F
Группа 1. Длина записи	2 байта	N
Группа 1. Данные	N x 2 байт	
Группа 2. Тип запроса	1 байт	0x6
Группа 2. Номер файла	2 байта	0x1-0xFFFF
Группа 2. Номер записи	2 байта	0x0-0x270F
Группа 2. Длина записи	2 байта	N
Группа 2. Данные	N x 2 байт	

Ошибка

Функция	1 байт	0x95
Код ошибки	1 байта	0x1, 0x2, 0x3, 0x4 или 0x8

Пример записи группы данных, устройства с адресом 0x1

- 3-х регистров в файле 4, начиная с регистра 7;

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x15	**Функция**	0x15
Число байт	0xD	**Число байт**	0xD
Тип запроса	0x6	**Тип запроса**	0x6
Номер файла	0x0	**Номер файла**	0x0

(старший байт)		**(старший байт)**	
Номер файла (младший байт)	0x4	**Номер файла (младший байт)**	0x4
Номер записи (старший байт)	0x0	**Номер записи (старший байт)**	0x0
Номер записи (младший байт)	0x7	**Номер записи (младший байт)**	0x7
Длина записи (старший байт)	0x0	**Длина записи (старший байт)**	0x0
Длина записи (младший байт)	0x3	**Длина записи (младший байт)**	0x3
Данные 1 (старший байт)	0x6	**Данные 1 (старший байт)**	0x6
Данные 1 (младший байт)	0xAF	**Данные 1 (младший байт)**	0xAF
Данные 2 (старший байт)	0x4	**Данные 2 (старший байт)**	0x4
Данные 2 (младший байт)	0xBE	**Данные 2 (младший байт)**	0xBE
Данные 3 (старший байт)	0x10	**Данные 3 (старший байт)**	0x10
Данные 3 (младший байт)	0xD	**Данные 3 (младший байт)**	0xD

0x11 - Чтение ID (серийного номера)

Эта функция используется для чтения серийного номера изделия

Запрос

Функция	1 байт	0x11

Ответ

Функция	1 байт	0x11
Число байт	1 байт	
Байт ID устройства	2 байта	(базовый адрес в формате строки)
Байт статуса	1 байт	0xFF
Байты номера (гг nnnn)	6 байт	(номер в формате строки)

Ошибка

Функция	1 байт	0x91
Код ошибки	1 байта	0x1 или 0x4

Пример чтения ID SVA-35D, устройства с адресом 0x1

Запрос		**Ответ**	
Адрес	0x1	**Адрес**	0x1
Функция	0x11	**Функция**	0x11
		Число байт	0x8
		Байт ID устройства	0x36

		Байт ID устройства	0x30
		Байт статуса	0xFF
		Байт номера 1	0x31
		Байт номера 2	0x33
		Байт номера 3	0x30
		Байт номера 4	0x30
		Байт номера 5	0x30
		Байт номера 6	0x31

Коды ошибок:

0x1 – ошибка функции

0x2 – ошибка адреса данных

0x3 – ошибка значения данных

0x4 – ошибка обработки данных устройством

0x8 – ошибка четности данных файла (для функций 0x14, 0x15)