

Описание
протокола
обмена данными
стандарта
МЭК-870-5-1-95
формата FT3
МС1201

ОГЛАВЛЕНИЕ

Общие принципы передачи данных по стандарту МЭК-870-5-1-95 формат кадра FT3.....	3
Передача в сети.....	3
Фрейм.....	3
Формат кадра запроса.....	3
Формат кадра ответа.....	3
Система команд устройства дискретного вывода MC1201.....	5
Список команд MC1201.....	5
0x01 Подготовка к записи данных.....	5
0x02 Запись адреса.....	5
0x03 Чтение адреса.....	5
0x08 Получить типизацию устройства.....	6
0x15 Установка скорости обмена данными.....	6
0x50 Установить дискретный выход.....	6
0x51 Получить дискретный выход.....	7
0x52 Установить конфигурацию дискретного выхода.....	7
0x53 Получить конфигурацию дискретного выхода.....	8
0x54 Установить время удержания дискретного выхода.....	8
0x55 Получить время удержания дискретного выхода.....	8
0x58 Получить байт состояния устройства.....	9
0x59 Сбросить байт состояния устройства.....	9
Приложение А.....	9
Приложение Б. Пример программы расчета CRC.....	11

ОБЩИЕ ПРИНЦИПЫ ПЕРЕДАЧИ ДАННЫХ ПО СТАНДАРТУ МЭК-870-5-1-95 ФОРМАТ КАДРА FTЗ.

ПЕРЕДАЧА В СЕТИ

Устройства в сети отвечают на запросы главного контроллера. Байты идут непрерывным потоком. Запрос – ответ. Начало кадра запроса и ответа идентифицируется маркером (двумя специальными байтами). МС1201 начинает отвечать через 2 мс после получения последнего байта запроса.

ФРЕЙМ

Назначение битов: 1 стартовый бит; 8 бит данных, младшим значащим разрядом вперед, паритет отсутствует; 1 стоповый бит

старт	1	2	3	4	5	6	7	8	стоп
-------	---	---	---	---	---	---	---	---	------

ФОРМАТ КАДРА ЗАПРОСА

Кадр запроса содержит следующие поля:

Наименование	Описание	Размер	Значение
Head	Стартовая последовательность	2 байта	0x05 0x64
DataLen	Длина данных	1 байт	0x00
ControlByte	Контрольный байт	1 байт	0x00
Address	Адрес	2 байта, младший байт передается первым	0x0000– 0xFFFF*
Command	Команда для устройства	1 байт	
Parameters	Параметры команды	9 байт	
CRC	Контрольная сумма	2 байта, старший байт передается первым	

Кадр запроса состоит из стартовой последовательности длиной 2 байта, одного блока данных длиной 14 байт и двух байт CRC в конце. CRC рассчитывается для 14 байт, начиная с длины.

*Address = 0x00FF - широковещательный адрес. При совместном использовании с протоколом MODBUS помнить, что в MODBUS допустимый адрес устройства ограничен значением 0x01 – 0xF7.

См. структуру [PKTSEND](#).

ФОРМАТ КАДРА ОТВЕТА

Кадр ответа состоит из стартовой последовательности длиной 2 байта и одного или нескольких блоков данных.

Если число передаваемых данных не более 10 байт, то кадр ответа содержит 1 блок данных, фиксированной длины - 16 байт (из них 4 байта – заголовочная часть, 2 байта - CRC). В поле длины DataLen, независимо от количества байт данных в блоке, передается 14. Содержимое незадействованных байт данных может быть произвольным, CRC считается для всех 14 байт, начиная с поля длины.

Кадр ответа с одним блоком данных имеет вид:

Наименование	Описание	Размер	Значение
Head	Стартовая последовательность	2 байта	0x05 0x64
DataLen	Длина данных	1 байт	0x0E
ControlByte	Контрольный байт	1 байт	0x00
Address	Адрес	2 байта, младший байт передается первым	0x0000– 0xFFFF
Data	Данные	10 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	

Кадр ответа из нескольких блоков содержит следующие поля:

Наименование	Описание	Размер	Значение
Head	Стартовая последовательность	2 байта	0x05 0x64
DataLen	Длина данных в кадре	1 байт	0x0F – 0xFF
ControlByte	Контрольный байт	1 байт	0x00
Address	Адрес	2 байта, младший байт передается первым	0x0000– 0xFFFF
Data	Данные	10 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	
Data	Данные	14 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	
...
Data	Данные	1-14 байт, младший байт передается первым	
CRC	Контрольная сумма	2 байта, старший байт передается первым	

Если число передаваемых данных более 10 байт, то кадр ответа содержит несколько блоков данных. Каждый блок данных заканчивается двумя байтами CRC. Первый блок данных также имеет заголовочную часть (4 байта), которая является заголовочной частью для всего кадра (последующие блоки не содержат заголовочной части). В поле длины DataLen указывается количество байт данных в кадре (без стартовой последовательности и CRC).

Длина первого блока всегда 16 байт (с учетом заголовочной части и 2 байт CRC), длина последнего блока определяется количеством байт данных в нем и может находиться в пределах от 3 (1 байт данных, 2 байта CRC) до 16, все промежуточные блоки имеют длину 16 байт (14 байт данных, 2 байта CRC).

В поле DataLen указывается длина данных Data плюс 4 байта, учитывающие размер полей DataLen, ControlByte и Address. Длина кадра ответа (исключая поле Head и поля CRC) не должна превышать 255 байт.

См. структуры [PKTHEAD](#), [PKTREADHEAD](#), [PKTREADDATA](#).

СПИСОК КОМАНД MC1201

Код команды	Наименование
0x01	Подготовка к записи данных
0x02	Изменение адреса устройства
0x03	Чтение адреса устройства
0x08	Получить типизацию устройства.
0x15	Установка скорости обмена данными
0x50	Установить дискретный выход
0x51	Получить дискретный выход
0x52	Установить конфигурацию дискретного выхода
0x53	Получить конфигурацию дискретного выхода
0x54	Установить время удержания дискретного выхода
0x55	Получить время удержания дискретного выхода
0x58	Получить байт состояния устройства
0x59	Сбросить байт состояния устройства

0X01 ПОДГОТОВКА К ЗАПИСИ ДАННЫХ

Команда 0x01 «Подготовка к записи данных» является предварительной для любой команды, изменяющей внутренние данные энергонезависимой памяти MC1201.

Параметры	Байты структуры PARAMETRS
Признак команды	P1=0xA5:

Возвращаемые данные: нет

0X02 ЗАПИСЬ АДРЕСА

Установить адрес в сети RS-485.

Предварительная команда: 0x01: «Подготовка к записи данных во флэш-память устройства»

Параметры	Байты структуры PARAMETRS
Старый адрес	P1, P2
Новый адрес	P3, P4

Младший байт передается первым.

Возвращаемые данные: нет

0X03 ЧТЕНИЕ АДРЕСА

Параметры	нет
-----------	-----

Возвращаемые данные: (см. поле Data в разделе «[Формат кадра ответа](#)»)

- 0-1 байт Data: адрес; младший байт передается первым.

0X08 ПОЛУЧИТЬ ТИПИЗАЦИЮ УСТРОЙСТВА

Параметры	нет
-----------	-----

Возвращаемые данные: структура [TUsotype](#)

Значение поля Model передается в шестнадцатеричном виде. Для правильной интерпретации модели устройства необходимо поменять байты местами.

Для устройства MC1201 значение поля Model 0x0112.

0X15 УСТАНОВКА СКОРОСТИ ОБМЕНА ДАННЫМИ

Установить скорость RS-485.

Предварительная команда: 0x01: "Подготовка к записи данных во флэш-память устройства"

Параметры	Байты структуры PARAMETRS
Код скорости	P1

Код скорости	Скорость
1	19200
2	9600 при первом включении
3	4800
4	2400
5	1200

Возвращаемые данные: нет

0X50 УСТАНОВИТЬ ДИСКРЕТНЫЙ ВЫХОД

Установить дискретный выход по маске значением. Команда устанавливает выходы и начинает цикл их удержания, если это необходимо.

Параметры	Байты структуры PARAMETRS
Код маски	P1
8 бит дискретного выхода	P2
Пароль	P3=0x9C, P4=0x39

Код маски	Маска
0	нет маски
1	OR
2	XOR
3	AND
4	NOT

Младший бит байта дискретного выхода (P2) соответствует младшему pin выхода.

ПРИМЕР1:

Код маски	1
8 бит дискретного выхода	01000000
выход до подачи команды	00010000
выход после подачи команды	01010000

ПРИМЕР2:

Код маски	0
8 бит дискретного выхода	01000000
выход	01000000

Возвращаемые данные: нет

0X51 ПОЛУЧИТЬ ДИСКРЕТНЫЙ ВЫХОД

Параметры	Байты структуры PARAMETRS
Флаг сброса	P9

Если флаг сброса равен 1, то байт состояния сбрасывается после передачи, т.е. обнуляются все его биты.

Возвращаемые данные: (см. поле Data в разделе “[Формат кадра ответа](#)”)

- 0 байт Data – 8 бит дискретного выхода;
- 9 байт Data – байт состояния устройства (см. “[Байт состояния устройства MC1201](#)”).

0X52 УСТАНОВИТЬ КОНФИГУРАЦИЮ ДИСКРЕТНОГО ВЫХОДА

Установить тип времени удержания и дискретизацию выходов.

Предварительная команда: 0x01: “[Подготовка к записи данных во флэш-память устройства](#)”

Параметры	Байты структуры PARAMETRS
Код типа времени	P1
Дискретизация	P2

Код типа времени	Тип времени
0	миллисекунда
1	секунда

Дискретизация не может быть 255 или 0 (устанавливается в 1). При вычислении реального времени удержания дискретизация умножается на время удержания выхода. Если дискретизация равна 0 или 255, то она не участвует в формировании реального времени удержания выхода, то есть в этом случае дискретизация устанавливается равной 1.

Возвращаемые данные: нет

0X53 ПОЛУЧИТЬ КОНФИГУРАЦИЮ ДИСКРЕТНОГО ВЫХОДА

Параметры	Байты структуры PARAMETRS
Тип конфигурации	P1

Код типа конфигурации	Тип конфигурации
0	текущая
1	та, которая будет использована в следующем цикле удержания

Примечание: устанавливаемая конфигурация выхода записывается в память устройства, но не становится текущей. Она станет текущей, как только будет выдана команда 0x50 “Установить дискретный выход” и начнется цикл удержания выхода.

Возвращаемые данные: (см. поле Data в разделе “Формат кадра ответа”)

- 0 байт Data – код типа времени (0 – миллисекунда; 1 – секунда);
- 1 байт Data – дискретизация.

0X54 УСТАНОВИТЬ ВРЕМЯ УДЕРЖАНИЯ ДИСКРЕТНОГО ВЫХОДА

Предварительная команда: 0x01: “Подготовка к записи данных во флэш-память устройства”

Параметры	Байты структуры PARAMETRS
Время удержания 0-7 выхода	P1-P8

Примечание: при значении параметров P1-P8 равных 0 устанавливается бесконечное время удержания соответствующего выхода.

Реальное время удержания выхода	время удержания выхода *дискретизацию сек/мсек
Минимальное время удержания	1 миллисекунда
Максимальное время удержания	17,9 час

Возвращаемые данные: нет

0X55 ПОЛУЧИТЬ ВРЕМЯ УДЕРЖАНИЯ ДИСКРЕТНОГО ВЫХОДА

Параметры	Байты структуры PARAMETRS
Тип времени удержания	P1

Код типа времени удержания	Тип времени
0	текущее
1	то, которое будет использовано в следующем цикле удержания

Примечание: устанавливаемое время удержания выхода записывается в память устройства, но не становится текущим. Оно станет текущим, как только будет выдана команда 0x50 “Установить дискретный выход” и начнется цикл удержания выхода.

дискретный выход” и начнется цикл удержания выхода. Текущее время удержания выхода по окончании цикла удержания сбрасывается в 0.

Возвращаемые данные: (поле Data см. в разделе протокола “[Формат кадра ответа](#)”)

- 0 – 7 байт Data – время удержания 0 – 7 выхода.

0X58 ПОЛУЧИТЬ БАЙТ СОСТОЯНИЯ УСТРОЙСТВА

Параметры	Байты структуры PARAMETRS
Флаг сброса	P1

Если P1=1, то байт состояния сбрасывается после передачи, т.е. обнуляются все его биты.

Возвращаемые данные: (см. поле Data в разделе “[Формат кадра ответа](#)”)

- 0 байт Data – байт состояния.

Формат байта состояния см. “[Байт состояния устройства MC1201](#)”

0X59 СБРОСИТЬ БАЙТ СОСТОЯНИЯ УСТРОЙСТВА

Сбросить байт состояния.

Параметры	Параметры
	нет

Возвращаемые данные: нет

ПРИЛОЖЕНИЕ А

PARAMETRS

Параметры пакета передачи

```
typedef struct _PARAMETRS
{
    unsigned char P1;    //Параметр N1
    unsigned char P2;    //Параметр N2
    unsigned char P3;    //Параметр N3
    unsigned char P4;    //Параметр N4
    unsigned char P5;    //Параметр N5
    unsigned char P6;    //Параметр N6
    unsigned char P7;    //Параметр N7
    unsigned char P8;    //Параметр N8
    unsigned char P9;    //Параметр N9
} PARAMETRS;
```

PKTHEAD

Заголовок пакета.

```
typedef struct _PKTHEAD
{
    unsigned char HeadByte1; //Сигнатура заголовка: байт N1 = 0x05
```

```
unsigned char HeadByte2; //Сигнатура заголовка: байт N2 = 0x64
}PKTHEAD;
```

PKTREADHEAD

Стартовый пакет приема

```
typedef struct _PKTREADHEAD
{
    unsigned char DataLen; //Длина данных
    unsigned char ControlByte; //Контрольный байт
    unsigned short Address; //Адрес устройства
    unsigned char Data[10]; //Данные
    unsigned short CRC; //Контрольная сумма
} PKTREADHEAD;
```

PKTREADDATA

Пакет приема данных

```
typedef struct _PKTREADDATA
{
    unsigned char Data[14]; //Данные
    unsigned short CRC; //Контрольная сумма
} PKTREADDATA;
```

Примечание: длина поля Data в зависимости от размера кадра может варьироваться от 1 до 14.

PKTSEND

Пакет для передачи

```
typedef struct _PKTSEND
{
    PKTHEAD Head; //Заголовок пакета
    unsigned char DataLen; //Длина данных
    unsigned char ControlByte; //Контрольный байт
    unsigned short Address; //Адрес устройства
    unsigned char Command; //Команда для устройства
    PARAMETRS P1P9; //Параметры
    unsigned short CRC; //Контрольная сумма
} PKTSEND;
```

TUSOTYPE

```
typedef struct {
    unsigned short Model; // Модель устройства
    unsigned char HardwareVersion; // Аппаратная версия
}
```

```

unsigned char SoftwareVersion;    // Программная версия
unsigned char Reserv4;           // Не используется
unsigned char Reserv5;           // Не используется
unsigned char Reserv6;           // Не используется
unsigned char Reserv7;           // Не используется
unsigned short SerialNumber;     // Фабричный серийный номер
}TUsotype;

```

БАЙТ СОСТОЯНИЯ УСТРОЙСТВА MC1201

Формат байта состояния

```

typedef struct{
    unsigned char ProcReset    :1;    // Сброс процессора
    unsigned char ErrFlash    :1;    // Ошибка: флэш-память не отвечает
    unsigned char ErrFlashCRC :1;    // Ошибка CRC флэш
    unsigned char ErrPocketCRC :1;    // Ошибка CRC пакета RS-485
    unsigned char b4           :1;
    unsigned char b5           :1;
    unsigned char b6           :1;
    unsigned char TimeKeepingActive:1;
}TUsotype1201State;

```

Комментарии:

0-бит	Нет
1-бит	Ошибка флэш-памяти: флэш-память не отвечает. Если эта ошибка появилась, значит физически испорчена флэш-память. MC не работоспособно.
2-бит	Ошибка CRC флэш. Эта ошибка появляется, когда считывание из флэш проходит успешно, но считанные данные не проходят тест на контрольную сумму. Здесь возможны два варианта: <ul style="list-style-type: none"> • А: Испортился один или несколько битов флэш. MC не работоспособно. • В: Данные во флэш не установлены. Переустановите данные и работайте дальше.
3-бит	Ошибка CRC пакета RS-485. Принятый пакет не прошел тест на контрольную сумму
7-бит	Цикл времени удержания выходов активен. Этот бит нельзя сбросить.

ПРИЛОЖЕНИЕ Б. ПРИМЕР ПРОГРАММЫ РАСЧЕТА CRC

```

const unsigned short crctable_ft3[256] = {
0x0000, 0x9EB3, 0xA3D5, 0x3D66, 0xD919, 0x47AA, 0x7ACC, 0xE47F,
0x2C81, 0xB232, 0x8F54, 0x11E7, 0xF598, 0x6B2B, 0x564D, 0xC8FE,
0x5902, 0xC7B1, 0xFAD7, 0x6464, 0x801B, 0x1EA8, 0x23CE, 0xBD7D,
0x7583, 0xEB30, 0xD656, 0x48E5, 0xAC9A, 0x3229, 0x0F4F, 0x91FC,
0xB204, 0x2CB7, 0x11D1, 0x8F62, 0x6B1D, 0xF5AE, 0xC8C8, 0x567B,

```

0x9E85, 0x0036, 0x3D50, 0xA3E3, 0x479C, 0xD92F, 0xE449, 0x7AFA,
0xEB06, 0x75B5, 0x48D3, 0xD660, 0x321F, 0xACAC, 0x91CA, 0x0F79,
0xC787, 0x5934, 0x6452, 0xFAE1, 0x1E9E, 0x802D, 0xBD4B, 0x23F8,
0xFABB, 0x6408, 0x596E, 0xC7DD, 0x23A2, 0xBD11, 0x8077, 0x1EC4,
0xD63A, 0x4889, 0x75EF, 0xEB5C, 0x0F23, 0x9190, 0xACF6, 0x3245,
0xA3B9, 0x3D0A, 0x006C, 0x9EDF, 0x7AA0, 0xE413, 0xD975, 0x47C6,
0x8F38, 0x118B, 0x2CED, 0xB25E, 0x5621, 0xC892, 0xF5F4, 0x6B47,
0x48BF, 0xD60C, 0xEB6A, 0x75D9, 0x91A6, 0x0F15, 0x3273, 0xACC0,
0x643E, 0xFA8D, 0xC7EB, 0x5958, 0xBD27, 0x2394, 0x1EF2, 0x8041,
0x11BD, 0x8F0E, 0xB268, 0x2CDB, 0xC8A4, 0x5617, 0x6B71, 0xF5C2,
0x3D3C, 0xA38F, 0x9EE9, 0x005A, 0xE425, 0x7A96, 0x47F0, 0xD943,
0x6BC5, 0xF576, 0xC810, 0x56A3, 0xB2DC, 0x2C6F, 0x1109, 0x8FBA,
0x4744, 0xD9F7, 0xE491, 0x7A22, 0x9E5D, 0x00EE, 0x3D88, 0xA33B,
0x32C7, 0xAC74, 0x9112, 0x0FA1, 0xEBDE, 0x756D, 0x480B, 0xD6B8,
0x1E46, 0x80F5, 0xBD93, 0x2320, 0xC75F, 0x59EC, 0x648A, 0xFA39,
0xD9C1, 0x4772, 0x7A14, 0xE4A7, 0x00D8, 0x9E6B, 0xA30D, 0x3DBE,
0xF540, 0x6BF3, 0x5695, 0xC826, 0x2C59, 0xB2EA, 0x8F8C, 0x113F,
0x80C3, 0x1E70, 0x2316, 0xBDA5, 0x59DA, 0xC769, 0xFA0F, 0x64BC,
0xAC42, 0x32F1, 0x0F97, 0x9124, 0x755B, 0xEBE8, 0xD68E, 0x483D,
0x917E, 0x0FCD, 0x32AB, 0xAC18, 0x4867, 0xD6D4, 0xEBB2, 0x7501,
0xBDFF, 0x234C, 0x1E2A, 0x8099, 0x64E6, 0xFA55, 0xC733, 0x5980,
0xC87C, 0x56CF, 0x6BA9, 0xF51A, 0x1165, 0x8FD6, 0xB2B0, 0x2C03,
0xE4FD, 0x7A4E, 0x4728, 0xD99B, 0x3DE4, 0xA357, 0x9E31, 0x0082,
0x237A, 0xBDC9, 0x80AF, 0x1E1C, 0xFA63, 0x64D0, 0x59B6, 0xC705,
0x0FFB, 0x9148, 0xAC2E, 0x329D, 0xD6E2, 0x4851, 0x7537, 0xEB84,
0x7A78, 0xE4CB, 0xD9AD, 0x471E, 0xA361, 0x3DD2, 0x00B4, 0x9E07,
0x56F9, 0xC84A, 0xF52C, 0x6B9F, 0x8FE0, 0x1153, 0x2C35, 0xB286};

unsigned short crc_ft3(unsigned char *Data, unsigned char DataLen)

```
{  
    unsigned short crc = 0;  
    unsigned char uIndex;  
  
    while (DataLen--)  
    {  
        uIndex= ((crc>>8) ^ *Data++);  
        crc<<=8;  
        crc ^= crctable_ft3[uIndex];  
    }  
    return (crc>>8)|(crc<<8);  
}
```