

Протокол датчика температуры цифрового показывающего ДТЦП-24М. Версия 1.0

1. Оглавление

2.	Формат пакета.....	2
1.	Формат команды.....	2
2.	Формат ответа.....	2
3.	Контрольная сумма.....	3
3.	Коды команд.....	4
	Чтение одного или нескольких USHORT регистров.....	4
	Запись одного регистра.....	4
4.	Номера регистров.....	4

2. Формат пакета

Данные передаются с использованием стандартного интерфейса RS-485.

Скорость передачи: 9600

Тип паритета: нет

Стоповых бит: 1

Протокол связи – Modbus RTU.

Все читаемые и записываемые данные организован через USHORT регистры.

Задержка между двумя последовательными байтами больше длительности 1 байта считается началом нового пакета.

Все 16 битные и 32 битные значения передаются старшим байтом вперёд.

1. Формат команды

ShortAdr	CmdCode	Data...	CRC16
----------	---------	---------	-------

ShortAdr – сетевой адрес датчика, которому адресована команда.

Адрес 0 используется для ширококвещательных команд, на запросы с таким адресом отвечает любой датчик при этом в ответе будет стоять его текущий сетевой адрес.

CmdCode - код команды.

Data – данные специфичные к конкретной команде CmdCode.

CRC16 – контрольная сумма CRC16 всех байт пакета (младшим байтом вперёд)

2. Формат ответа

Формат ответа полностью повторяет формат запроса за исключением:

- поле ShortAdr в ответе содержит сетевой адрес отвечающего датчика

- в поле **CmdCode** ответа старший бит означает наличие ошибки, если он установлен – команда выполнена с ошибкой и поле данных содержит 1 байт расшифровки ошибки, если старший бит сброшен – команда выполнена успешно и поле данных содержит ответ на команду.

Младшие 7 бит копируются из запроса.

коды ошибок:

01 - ILLEGAL FUNCTION - если команда с указанным кодом не поддерживается датчиком

02 - ILLEGAL DATA ADDRESS - запрошен недопустимый номер регистра для чтения или для записи

03 - ILLEGAL DATA VALUE - передано недопустимое значение

04 - Failure In Associated Device

3. Контрольная сумма

Контрольная сумма вычисляется по стандартному алгоритму CRC-16-[IBM](#).

Пример реализации алгоритма взят с сайта

http://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BA%D0%BB%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%B7%D0%B1%D1%8B%D1%82%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4#CRC-16

«Пример программы табличного (быстрого) расчёта стандартного (ARC) CRC-16 на языке Си»

```
/*
Name   : CRC-16
Poly   : 0x8005      x^16 + x^15 + x^2 + 1
Init   : 0xFFFF
Revert : true
XorOut : 0x0000
Check  : 0x4B37 ("123456789")
MaxLen : 4095 байт (32767 бит) - обнаружение
        одинарных, двойных, тройных и всех нечетных ошибок
*/
const unsigned short Crc16Table[256] = {
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
    0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
    0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
    0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
    0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
    0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
    0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
    0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
    0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
    0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
    0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
    0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
    0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
    0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
    0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
    0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0xA5C0, 0x6480, 0xA441,
    0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
    0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
    0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
    0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
    0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
    0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
    0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
    0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
    0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x99C0, 0x5880, 0x9841,
    0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
    0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
    0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
    0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};

unsigned short Crc16(unsigned char * pcBlock, unsigned short len)
{
    unsigned short crc = 0xFFFF;
    while (len--)
        crc = (crc >> 8) ^ Crc16Table[(crc & 0xFF) ^ *pcBlock++];
    return crc;
}
```

Чтобы проверить что алгоритм расчёта CRC реализован правильно используйте пример:

```
для набора байт(HEX):   81 03 00
CRC16 будет:           21 18
а для набора байт:     81 03 00 21 18
CRC16 будет:           00 00
```

Таким образом контроль CRC упрощается до простой проверки на 0.

3. Коды команд

Чтение одного или нескольких USHORT регистров

Код: 3

Данные в пакете: 2 USHORT'a

HIBYTE(StartReg)	LOBYTE(StartReg)	HIBYTE(RegsCnt)	LOBYTE(RegsCnt)
------------------	------------------	-----------------	-----------------

StartReg – адрес первого читаемого регистра

RegsCnt – количество читаемых регистров

Ответ: $1 + 2 * \text{<количество регистров>}$ байт

количество данных в пакете (равно <RegsCnt>*2)	HIBYTE(Reg1)	LOBYTE(Reg1)	...
---	--------------	--------------	-----

Запись одного регистра

Код: 6

Описание: запись одиночного USHORT регистра. Используется для смены адреса.

Данные в пакете: : 2 USHORT'a

HIBYTE(RegAdr)	LOBYTE(RegAdr)	HIBYTE(RegsData)	LOBYTE(RegData)
----------------	----------------	------------------	-----------------

RegAdr – адрес записываемого регистра

RegData – записываемое значение

Ответ: содержит копию данных запроса, то есть те же 2 USHORT'a

4. Номера регистров

Регистр 0 (R/W): адрес устройства в младшем байте

Регистр 1 (R): тип устройства (для датчика температуры показывающего - 3)

Регистр 2 (R): показания температуры в формате signed short в единицах $1/256$ °C, то есть в старшем байте находится целое значение градусов, в младшем – дробная часть. Диапазон значений $[-128.0 +127.99]$ - верхняя граница = $127 + 254/255$.

В случае ошибки измерения температуры в этом регистре будет 0x7FFF, а в следующем – код ошибки чтения показаний.

Регистр 3 (R): код ошибки измерения температуры – этот регистр должен всегда читаться одновременно с регистром 2, чтобы гарантировать, что значения относятся к одному моменту времени.

код ошибки	Расшифровка
0	данные температуры верны
1	инициализация датчика после перезагрузки
2	нет ответа первичного датчика (висит 1 в линии, датчик не подключен)
3	нет ответа первичного датчика (висит 1 в линии, датчик подключен неверно)
4	нет ответа первичного датчика (висит 0 в линии, линия закорочена)
5	первичный датчик постоянно перезагружается, нестабильное питание
6	преобразование температуры идёт слишком долго, неисправность первичного датчика

Регистр 4 (R): Резервирован, должен всегда читаться 0.