

**Регулятор температуры “Струмень”
РТМ-02
РТМ-03**

**РУКОВОДСТВО
ПО РАБОТЕ С КАНАЛОМ СВЯЗИ**

СОДЕРЖАНИЕ

| | |
|--|--|
| 1. Введение..... | |
| 2. Каналы связи..... | |
| 3. Протокол обмена..... | |
| 3.1 Чтение температур..... | |
| 3.2 Установка режима..... | |
| 3.3 Установка типа контура | |
| 3.4 Установка режима ДИСТ..... | |
| 3.5 Установка режима RESTART..... | |
| 3.6 Чтение памяти | |
| 3.7 Запись памяти | |
| 3.8 Чтение архива | |
| 4. Циклический избыточный код (CRC)..... | |

1 Введение.

Настоящий документ предназначен для подключения регулятора температуры РТМ-03 (в дальнейшем регулятор) к ПЭВМ и содержит техническое описание и протокол обмена по работе с каналом связи.

2 Каналы связи.

Для связи с ПЭВМ регулятор имеет последовательный канал связи типа RS232 или RS485.

Параметры интерфейсов RS232 и RS485:

| | |
|--------------------------|------------|
| скорость обмена | -9600 бод. |
| тип паритета | -нет. |
| число информационных бит | -8. |
| число стоповых бит | -1. |

3 Протокол обмена.

Диалог между компьютером и регулятором основан на принципе “главный-подчинённый”. Роль “главного” всегда играет компьютер, а регулятор “подчинённый”.

Основные положения протокола приведены ниже:

- протокол предусматривает одно главное и до 255 подчинённых устройств.
- сообщения, которыми обмениваются между собой главное и подчинённое. устройства помещаются в пакеты. Структура пакетов представлена на рисунке 1.
- каждое подчинённое устройство, включённое в сеть, имеет свой уникальный адрес.
- для повышения надёжности передачи данных используется избыточный циклический код (CRC).
- Если задержка между байтами будет превышать 0,02 сек, регулятор будет считать, что это конец посылки и следующий байт воспримет как начало следующей посылки.

Рис1. Структура пакета.

| | | | |
|----------|--------------|-----------|--------------|
| Байт1 | Байт2 | | Байт n и n+1 |
| Адр.устр | Код команды. | Сообщение | Сум. |

Адр. устр – физический адрес регулятора. Адрес устанавливается в регуляторе. (см. тех. описание РТМ-03).

Код команды – коды команд приведены в таблице 1.

№Блока – равен “0” предназначен для дальнейшего расширения возможностей системы.

Сум - контрольная сумма (CRC-подробно рассмотрена в описании циклического избыточного кода).

Таблица 1.

| Коды команд | |
|--|------|
| _RD_TEMP - чтение температуры | 10H |
| _RD_RAM - чтение памяти | 51H |
| _RD_IRAM - чтение памяти | 52H |
| _RD_EEPROM - чтение памяти | 53H |
| _SET_SER_REJ - установка режима работы регулятора | 80H |
| _RESTART - перезапуск | 81H |
| _WR_EEPROM - запись памяти | 90H |
| _WR_IRAM - запись памяти | 91H |
| _SET_MODE_DIST установка режима ДИСТ | 82H |
| _SET_SER_TIP - установка типа контура регулирования | 83H |
| Коды ответов | |
| _KV_OK - ответ положительный | 0C0H |
| _KV_ER - ответ отрицательный | 0C1H |
| Коды ошибок | |
| _ER_COM - ошибка параметра Код команд | 2 |
| _ER_PAR - ошибка параметров | 4 |
| _ER_WR - ошибка параметра Запись памяти | 6 |
| _ER_RD - ошибка параметра Чтение памяти | 7 |
| _ER_MODE_R - один из контуров в регуляторе установлен в ручной режим | 8 |
| _ER_MODE_P - не установлен режим ДИСТ | 9 |
| _ER_BLOCK - ошибка параметра №Блока | 5 |

После того, как регулятор принял пакет и проверил правильность параметров, он формирует ответ. В случае, если пакет сформирован верно, регулятор возвращает положительный ответ (Таблица 1) либо пакет несущий в себе какую то информацию (чтение температур, чтение памяти). В противном случае регулятор возвращает ошибку и её код(Таблица 1).

Случаи, когда регулятор не отвечает на пакет:

- не верно указан Адр. Устр.;
- ошибка CRC (контрольной суммы);
- регулятор не отвечает на команду RESTART;

3.1 Чтение температур

Рис2. Структура пакета команды <<чтение температуры>>.

| | | | | | |
|-----------|-------------|--------|--------|------------|------------|
| Адр.устр. | Код команд. | №Блока | №датч. | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|--------|------------|------------|

Рис3. Структура пакета команды <<ответ на чтение температуры>>.

| | | | | | | |
|-----------|-------------|--------|--------|------------|------------|------------|
| Адр.устр. | Код команд. | №Блока | №датч. | Код темпер | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|--------|------------|------------|------------|

№ датч – физический номер опрашиваемого датчика температуры. Принимает значения от 1 до 8.

Код темпер - код температуры представляет число с плавающей точкой .

число с плавающей точкой представляет собой трехбайтное число (первый байт - порядок (характеристика) данного числа, а следующие два байта –мантисса. Формат трехбайтного числа с плавающей точкой состоит из пяти полей:

- знак порядка;
- порядок;
- знак мантиссы;
- переполнение;
- мантисса.

Поле знака порядка –однобитное и содержит единицу, если порядок отрицательный, нуль, если порядок положительный.

Поле порядка - семибитное. Оно указывает число битовых позиций, на которое нужно сдвинуть мантиссу, чтобы представить его в форме "0.пппп ". Если знак порядка содержит нуль, то мантисса сдвигается влево на порядок. если же знак порядка содержит единицу, то мантисса сдвигается вправо на порядок.

Поле знака мантиссы - однобитное. Оно содержит единицу, если мантисса (число) отрицательное, и нуль, если число положительное.

Поле переполнения -однобитное. Оно предназначено для фиксирования переполнения мантиссы, полученной в результате выполнения различных операций.

Поле мантиссы -14-битное. Оно содержит мантиссу числа.

Примеры представления чисел с плавающей точкой:

| десятичное число | число с плавающей точкой (HEX) |
|------------------|--------------------------------|
| 0 | 00 00 00 |
| 1 | 01 20 00 |
| -1 | 01 a0 00 |
| 0.5 | 00 20 00 |
| -8 | 04 a0 00 |

Примечание. Максимальное число, которое может быть преобразовано в число с плавающей точкой без потери точности: $16383 - 0.1638 \cdot 10^{**5}$, где ****** - обозначает степень.

Двоичное значение числа может быть вычислено по формуле:

$$d = (-1)^s \cdot (2^e \cdot (-1)^p) \cdot (f),$$

где *s* - знак мантиссы;

p - знак порядка;

(*s* и *p* равны нулю, если число положительное, и единице, если число отрицательное);

e - значение порядка;






f - значение мантиссы.

3.2 Установка режима

Рис4. Структура пакета команды <<установка режима>>.

| Адр.устр. | Код команд. | №Блока | №Контур | Режим работы | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|---------|--------------|------------|------------|
|-----------|-------------|--------|---------|--------------|------------|------------|

Регулятор имеет пять режимов работы:

1. Ручной режим  (режим можно включить только на регуляторе). В случае, если в регуляторе один из контуров работает в ручном режиме, программирование его будет невозможным (ошибка `_ER_MODE_R`).
2. Стоп . (Код режима работы равен "1")
3. Постоянно <<нормальный>> . (Код режима работы равен "2")
4. Постоянно <<пониженный>> . (Код режима работы равен "3")
5. Программный . (Код режима работы равен "4")

№Контур принимает значения 0 или 1.

0-первый контур.

1-второй контур.

Описание режимов работы регулятора и контуров изложено в "Регулятор температуры РТМ-03 "Струмень"". Паспорт.

3.3 Установка типа контура.

Рис5. Структура пакета команды <<установка типа контура>>.

| Адр.устр. | Код команд. | №Блока | №Контур | Тип контура | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|---------|-------------|------------|------------|
|-----------|-------------|--------|---------|-------------|------------|------------|

Тип контура регулирования может принимать следующие значения:

0-нет контура регулирования;

1-регулятор горячей воды;

2-регулятор температуры теплоносителя по температурному графику;

3-регулятор температуры теплоносителя по температуре в помещении;

4-регулятор температуры теплоносителя по температурному графику с

коррекцией по температуре в помещении;

Для корректной установки типа контура регулирования необходимо:

1. Установить режим ДИСТ.
2. Установить тип контура регулирования.
3. Перегрузить регулятор подав команду RESTART.

3.4 Установка режима ДИСТ.

Рис6. Структура пакета команды <<установка режима ДИСТ>>.

| | | | | |
|-----------|-------------|--------|------------|------------|
| Адр.устр. | Код команд. | №Блока | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|------------|------------|

3.5 Установка режима RESTART.

Рис7. Структура пакета команды <<RESTART>>.

| | | | | |
|-----------|-------------|--------|------------|------------|
| Адр.устр. | Код команд. | №Блока | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|------------|------------|

Команда RESTART приводит к полному перезапуску регулятора.

3.6 Чтение памяти.

Рис8. Структура пакета команды <<чтение памяти>>.

| | | | | | | | |
|----------|-------------|--------|------------|-------------|------------|------------|------------|
| Адр.устр | Код команд. | №Блока | Адр (мл.б) | Адр (ст.ба) | Кол-во б-т | Сум.(ст.б) | Сум.(мл.б) |
|----------|-------------|--------|------------|-------------|------------|------------|------------|

Кол-во б-т – Количество байт, значения которых нужно прочесть.

В случае, если количество байт будет превышать 6, регулятор выдаст ошибку параметров.

Рис9. Структура пакета команды <<ответ на чтение памяти>>.

| | | | | | | | | |
|-----------|-------------|--------|-----------------|----------------|------------|----------|------------|------------|
| Адр.устр. | Код команд. | №Блока | Адр б-та (мл.б) | Адр (ст.байта) | Кол-во б-т | Значение | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|-----------------|----------------|------------|----------|------------|------------|

Адр. ст/мл байта указывает адрес в памяти, по которому находятся те или иные коэффициенты и функции регулятора.

Значение – информация, которую присылает регулятор. Количество байт информации равно **Кол-во б-т**.

Адреса коэффициентов и функций указаны в док. “**Адреса коэффициентов и функций**” на конкретное исполнение регулятора.

3.7 Запись памяти.

Рис10. Структура пакета команды <<запись памяти>>.

| | | | | | | | | |
|-----------|-------------|--------|-----------------|-----------------|------------|----------|------------|------------|
| Адр.устр. | Код команд. | №Блока | Адр б-та (мл.б) | Адр б-та (ст.б) | Кол-во б-т | Значение | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|-----------------|-----------------|------------|----------|------------|------------|

Кол-во б-т – Количество байт, значения которых нужно записать.

Значение – информация, которую нужно записать в регулятор. Количество байт информации равно **Кол-во б-т**.

Примечание: В случае если **Кол-во б-т** будет превышать 6, регулятор не ответит на посылку.

3.8 Чтение архива.

Чтение архива выполняется после установки режима чтение архива.

Рис 11. Структура пакета команды <<Чтение архива>>.

| Адр.устр. | Код команд. | №Блока | Адр б-та (мл.б) | Адр б-та (ст.б) | Кол-во б-т | Сум.(ст.б) | Сум.(мл.б) |
|-----------|-------------|--------|-----------------|-----------------|------------|------------|------------|
|-----------|-------------|--------|-----------------|-----------------|------------|------------|------------|

Кол-во б-т – Количество байт, значения которых нужно прочитать. В режиме чтение архива осуществляется по 64 байта. Адрес начала архива равен 0ч4000.

4. Циклический избыточный код (CRC).

Контроль циклическим кодом применяется для повышения надежности передачи данных. Смысл контроля заключается в следующем. Запрос подвергается шифровке циклическим кодом. Полученный результат добавляется в конец запроса, и весь пакет отправляется подчиненному устройству. Подчиненное устройство выполняет те же действия над байтами запроса и сравнивает полученный результат с CRC принятого пакета, и в случае положительного результата выполняет требуемое действие. Затем оно формирует ответное сообщение, подвергает его той же процедуре шифровки, “прицепляет” полученный код в конец пакета и посылает его обратно главному устройству. Главное устройство выполняет ту же процедуру дешифровки, проверяя правильность принятого пакета. Вероятность обнаружения ошибки в одном разряде байта пакета равна 99,998%. В качестве примера рассмотрим вычисление CRC в виде функции написанной на языке C. Все возможные значения CRC помещены в два массива. Один массив содержит все возможные значения для старшего байта CRC, а второй – для младшего.

```
const unsigned char tblCRChi[]=
{
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, 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
};

```

```

const unsigned char tblCRClo[]=
{
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
};

```

```

unsigned short __fastcall CRC16(unsigned char *msg, unsigned short len)
{
    unsigned short idx;
    unsigned char CRChi=0xFF;
    unsigned char CRClo=0xFF;

    while(len--)
    {
        idx=(CRChi ^ *msg++) & 0xFF;
        CRChi=CRClo ^ tblCRChi[idx];
        CRClo=tblCRClo[idx];
    }
    return ((CRChi << 8) | CRClo);
}

```

Функция в качестве параметров принимает указатель на сообщение используемое для формирования CRC (msg) и размер сообщения в байтах (len), а возвращает 16-ти битное значение CRC.

ПРИЛОЖЕНИЕ 1

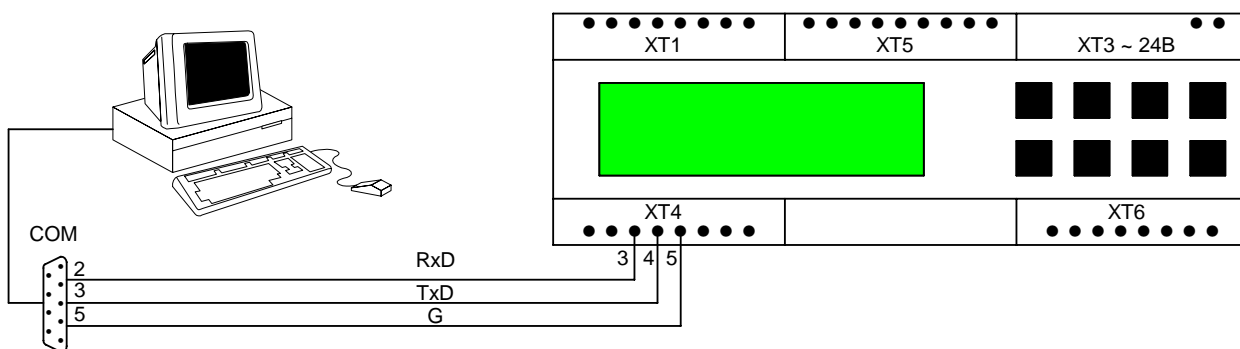


Рис1. Схема подключения РТМ-03 к ПЭВМ по каналу связи типа RS232.

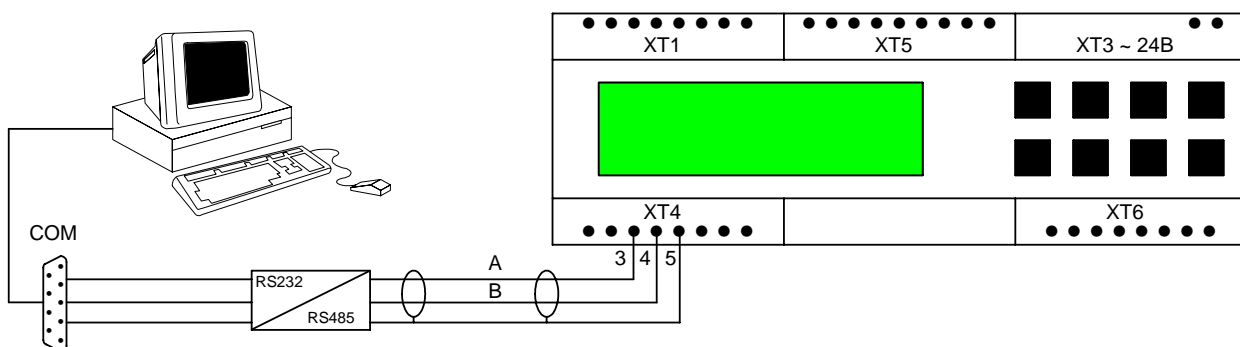


Рис2. Схема подключения РТМ-03 к ПЭВМ по каналу связи типа RS485.

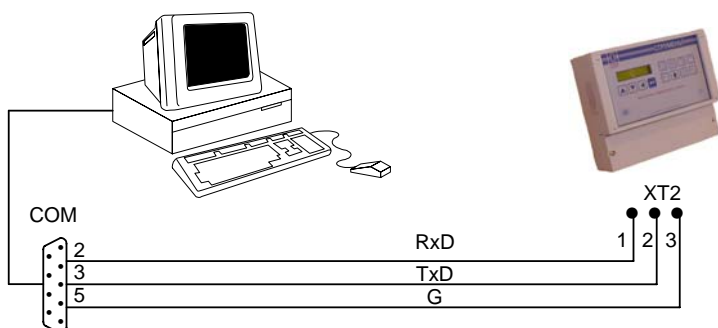


Рис3. Схема подключения РТМ-02 к ПЭВМ по каналу связи типа RS232.

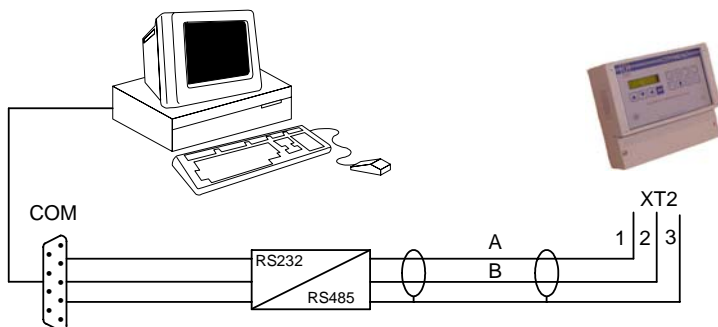


Рис4. Схема подключения РТМ-02 к ПЭВМ по каналу связи типа RS485.