



# **Регулятор температуры РТМ-03 "СТРУМЕНЬ"**

## **Инструкция оператора по работе с последовательным каналом связи**

**Исполнение регулятора для управления  
системами отопления**

Действительно с версии 1.0

Минск 2020

# СОДЕРЖАНИЕ

<b>1</b>	<b>ВВЕДЕНИЕ</b>	<b>3</b>
1.1	КАНАЛЫ СВЯЗИ	3
1.2	ПРОТОКОЛ	3
<b>2</b>	<b>ПАРАМЕТРЫ КОНФИГУРАЦИИ РЕГУЛЯТОРА</b>	<b>4</b>
2.1	ПАРАМЕТРЫ ПОСЛЕДОВАТЕЛЬНЫХ ПОРТОВ	4
2.2	ТИП ПРИВОДА КОНТУРА РЕГУЛИРОВАНИЯ	5
<b>3</b>	<b>ОБЩИЕ ПАРАМЕТРЫ РЕГУЛЯТОРА</b>	<b>6</b>
3.1	УПРАВЛЕНИЕ РЕЖИМОМ «ПРОГРАММИРОВАНИЕ»	6
3.2	РЕЖИМ РАБОТЫ КОНТУРОВ	6
3.3	ИМЯ И НОМЕР СЕРИЙНЫЙ	7
3.4	ВРЕМЯ И ДАТА	8
<b>4</b>	<b>РАБОТА С ТЕМПЕРАТУРАМИ</b>	<b>9</b>
4.1	ЧТЕНИЕ ТЕМПЕРАТУР ПО НОМЕРУ	9
<b>5</b>	<b>КАНАЛЫ ВВОДА-ВЫВОДА</b>	<b>9</b>
5.1	ЧТЕНИЕ ДИСКРЕТНЫХ ВХОДОВ	9
5.2	ЧТЕНИЕ ДИСКРЕТНЫХ ВЫХОДОВ	10
5.3	ЧТЕНИЕ АНАЛОГОВЫХ ВХОДОВ	10
5.4	ЧТЕНИЕ АНАЛОГОВЫХ ВЫХОДОВ	11
5.5	РАБОТА С ЦИФРОВЫМИ МАНОМЕТРАМИ	11
<b>6</b>	<b>ПАРАМЕТРЫ И КОЭФФИЦИЕНТЫ РЕГУЛЯТОРА</b>	<b>12</b>
6.1	ТИП КОНТУРА РЕГУЛИРОВАНИЯ	12
6.2	УСТАНОВКА ФУНКЦИЙ	12
6.3	КОЭФФИЦИЕНТЫ ТЕМПЕРАТУРНОГО ГРАФИКА	15
6.4	КОЭФФИЦИЕНТЫ ПИД-РЕГУЛЯТОРА	16
6.5	КОЭФФИЦИЕНТЫ ФУНКЦИИ «ОГРАНИЧЕНИЕ ОБРАТКИ»	17
6.6	КОЭФФИЦИЕНТЫ ФУНКЦИИ «ОГРАНИЧЕНИЕ ПО СЕТИ»	18
6.7	КОЭФФИЦИЕНТЫ "ФУНКЦИИ КОМПЕНСАЦИИ"	19
6.8	КОЭФФИЦИЕНТЫ ФУНКЦИИ «СНИЖЕНИЕ И ПРОГРЕВ»	20
6.9	КОЭФФИЦИЕНТЫ ФУНКЦИИ «ФУНКЦИЯ ЗАЩИТЫ»	21
6.10	КОЭФФИЦИЕНТЫ ФУНКЦИИ «ПРЕДЕЛ ОТОПЛЕНИЯ»	21
6.11	КОЭФФИЦИЕНТЫ ФУНКЦИИ «ФУНКЦИЯ НАСОСОВ ПОДПИТКИ»	22
6.12	КОЭФФИЦИЕНТЫ ФУНКЦИИ «КОЭФФИЦИЕНТЫ НАСОСОВ КОНТУРОВ»	23
6.13	КОЭФФИЦИЕНТЫ ФУНКЦИИ " РАБОТА НАСОСОВ ПО ТЕМПЕРАТУРЕ"	24
<b>7</b>	<b>ЧТЕНИЕ СЕРВИСНЫХ ПАРАМЕТРОВ РЕГУЛЯТОРА</b>	<b>25</b>
7.1	ЧТЕНИЕ ОШИБОК И ПРЕДУПРЕЖДЕНИЙ	25
<b>8</b>	<b>АРХИВЫ</b>	<b>26</b>
8.1	ЧТЕНИЕ АРХИВА СОБЫТИЙ	26
8.2	ЧТЕНИЕ АРХИВА ПОСЕЩЕНИЙ	26
8.3	ЧТЕНИЕ АРХИВА ОШИБОК	27
8.4	ПАРАМЕТРЫ АРХИВОВ ТЕМПЕРАТУР И ДАВЛЕНИЙ "	28
8.5	ЧТЕНИЕ АРХИВОВ ТЕМПЕРАТУР И ДАВЛЕНИЙ	29
	<b>ПРИЛОЖЕНИЕ А. СХЕМЫ ПОДКЛЮЧЕНИЯ РТ</b>	<b>30</b>
	<b>ПРИЛОЖЕНИЕ Б ПОДСЧЕТ КОНТРОЛЬНОЙ СУММЫ</b>	<b>30</b>

# 1 ВВЕДЕНИЕ

Настоящее руководство оператора по работе с каналом связи предназначено для подключения регулятора температуры РТМ-03 (в дальнейшем РТ) к ПЭВМ и содержит описание протокол обмена.

## 1.1 Каналы связи

Для связи с внешними устройствами регулятор имеет два независимых последовательных канала связи.

Параметры обмена для интерфейсов RS232 или RS485 (в зависимости от исполнения РТ) могут быть установлены в следующих пределах:

- скорость обмена -от 600 до 19200 бод;
- тип паритета -нет, четность, нечетность;
- число информационных бит -8;
- число стоповых бит -1 или 2.

РТ позволяет проводить сеансы обмена по обоим каналам связи одновременно. В некоторых исполнениях РТ канал связи COM0 используется для организации обмена с дополнительными устройствами системы и работает в режиме «Главный» со своим протоколом обмена. Схемы подключения РТ по интерфейсам RS232 и RS485 представлены в приложении А.

## 1.2 Протокол

Диалог между компьютером и РТ основан на принципе “главный-подчинённый”. Роль “главного” всегда играет компьютер, а регулятор (регуляторы) “подчиненный”.

Основные положения протокола приведены ниже:

- протокол предусматривает одно главное и до 254 подчинённых устройств;
- сообщения, которыми обмениваются между собой главное и подчинённое устройства помещаются в пакеты. Структура пакетов представлена на Рисунок 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.

**Рисунок 1-** Структура пакета ответа без информации.

Байт1	Байт2	Байт3- Байт n-1	Байт n и n+1
адрес	код команды	сообщение	CRC

Поля ‘адрес’, ‘код команды’ и ‘CRC’ составляют обрамление пакета, а поле ‘сообщение’ –содержимое пакета.

Поле ‘адрес’ имеет размерность 1 байт и содержит сетевой адрес РТ, поле ‘код команды’ тоже однобайтное и определяет функцию, выполняемую пакетом. Допустимые значения этого поля будут подробно определены ниже.

Поле ‘CRC’ это 2-х байтное число представляющее собой результат шифрования циклическим кодом полей ‘адрес’, ‘функция’ и ‘сообщение’. Более подробно о контроле циклическим кодом и пример его расчета приведено в приложении Б.

Наполнение поля ‘сообщение’ зависит от функции выполняемой пакетом. Инициатива обмена всегда принадлежит ПЭВМ, он формирует пакет команды, передает

его в линию и ожидает ответ от РТ. В свою очередь все РТ 'прослушивают' линию и если пакет команды принят без искажений, проверяют поле 'адрес' и если оно равно 0 или совпадает с собственным адресом, то обнаруживший совпадение РТ приступает к более детальному анализу команды. После анализа не позже чем через 0,2 секунды ответ будет отправлен в линию.

Если запрос выполнен и не требуется передача информации, РТ отправляет пакет «Запрос выполнен». Структура пакета на Рисунок 2.

Если запрос выполнен и требуется передача информации, РТ отправляет пакет с сообщением, формат которого определен для каждого запроса индивидуально.

Если запрос не выполнен РТ возвращает пакет «Запрос ошибочен». Структура пакета «Запрос ошибочен» представлена на Рисунок 3. В поле «код ошибки» указывается причина отказа в выполнении запроса.

Если запрос содержит не верную контрольную сумму или поле «адрес» запроса не совпадает с адресом РТ – РТ не отвечает на запрос.

При запросе параметров записанных в память EEPROM, РТ передает параметры в составе структуры с контрольной суммой «uchar sum». Это 1- байтное число циклической суммы структуры параметров из EEPROM. Более подробно о контроле циклическим кодом и пример его расчета приведен в приложении Б. Если контрольная сумма структуры параметров не верна, то регулятор передает структуру параметров по умолчанию. В некоторых запросах факт передачи структуры параметров по умолчанию указывается в сообщении.

**Рисунок 2** Структура пакета “Запрос выполнен”.

Байт1	Байт2	Байт 3 и 4
адрес	код команды	CRC

Параметр *код команды* – 0xE2

**Рисунок 3** Структура пакета «Запрос ошибочен».

Байт1	Байт2	Байт3 и 4	Байт 4 и 5
адрес	код команды	код ошибки	CRC

Параметр *код команды* – 0xE1

Параметр *код ошибки* – представлен в Таблица 1.

**Таблица 1**

Код ошибки	Описание
0x01	ошибочный параметр
0x02	не существующий ' <i>код команды</i> '
0x03	ошибка чтения параметра
0x04	ошибка записи параметра
0x05	программирование запрещено
0x06	запрет обмена по последовательному порту

## 2 Параметры конфигурации регулятора

### 2.1 Параметры последовательных портов

1) Формат запроса «Чтение параметров последовательных портов».

0	1	2	3	4	5
адр	код	№com	zero	CRC	

Поле «код»- 0x0E; // READ\_COMPRM

Поле «№com» - номер COM порта 0 – COM0, 1-COM1.

**Ответ** – структура TComPrm.

```
typedef struct // Структура ответа на запрос "Чтение параметров порта"
{
uchar net_adr;
uchar command;
uchar com; // Номер порта: 0 – COM0, 1-COM1
uchar dummy ;
ComPrm comPrm;
ushort summ;
} TComPrm;
```

2) Формат структуры параметров последовательных портов в Таблица 2

**Таблица 2**

Сообщение			Общий размер 8 байт
формат	№ байта	данные	размер
unsigned short	4	скорость передачи	2 байта
	5		
unsigned char	6	тип интерфейса	1 байт
unsigned char	7	длина посылки	1 байт
unsigned char	8	тип паритета	1 байт
unsigned char	9	число стоповых бит	1 байт
unsigned char	10	адрес PT	1 байт
unsigned char	11	не используется	1 байт

```
typedef struct // Описание параметров последовательного порта
{
ushort baudrate; // скорость обмена
uchar type;
uchar char_len; // количество бит
uchar parity; // паритет
uchar num_stop; // число стоповых бит
uchar net_adr; // сетевой адрес
uchar sum;
}ComPrm;
```

## 2.2 Тип привода контура регулирования

1) Запрос " Чтение типа привода "

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x18. // READ\_TIP\_PRIVOD

```
typedef struct      // Структура запроса " Чтение типа привода "
{
uchar   net_adr;
uchar   command;
ushort  summ;
} TComRTipPrivod;
```

```
typedef struct      // Структура ответа на запрос "Чтение типа привода",
{
uchar   net_adr;
uchar   command;
uchar   tipPrivod[3];
uchar   dymmy;
ushort  summ;
}TComRWTipPrivod;
```

“Запрос выполнен” или «Запрос ошибочен».

### 3 Общие параметры регулятора

#### 3.1 Управление режимом «Программирование»

```
typedef struct      // Структура команды "Установка режима программирования"
{
uchar   net_adr;
uchar   command;
uchar   kodDostup[10];
ushort  summ;
}TComSetDostupFull;
```

поле « command »= 0x7F. // SET\_MODE\_PRG

```
typedef struct      // Структура команды "Выключение режима «Программирование"
{
uchar   net_adr;
uchar   command;
ushort  summ;
}TComClrProgI;
```

Поле «код»= 0x80. // CLR\_MODE\_PRG

**Ответ** – пакет “Запрос выполнен” или «Запрос ошибочен». Прочитать состояние режима «Программирование» - см. 3.3.

#### 3.2 Режим работы контуров

1) Запрос «Чтение режима работы контура»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x05. // READ\_REJK

Поле «№ Контура» - номер запрашиваемого контура от 0 до 2.

```
typedef struct                // Структура запроса "Чтение режима работы контура"
{
uchar   net_adr;
uchar   command;
uchar   kontur;
uchar   dymmy;
ushort  summ;
}TComRejK;
```

```
typedef struct                // Структура ответа на запрос "Чтение режима работы контура"
{
uchar   net_adr;
uchar   command;
uchar   kontur;
uchar   rejK;                // Текущий режим работы контура
uchar[16] dymy;
ushort  summ;
}TComReadRejK;
```

### 3.3 Имя и номер серийный

1) Запрос имени и номера PT

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

поле «код»= 0x10. // READ\_NAME

```
typedef struct                // Структура команды " Запрос имени, номера"
{
uchar   net_adr;
uchar   command;
ushort  summ;
}TReadStatus;
```

```
typedef struct                // Структура ответа на команду " Запрос имени, номера "
{
uchar   net_adr;
uchar   command;
uchar   numer[8];           // номер PT, формат ASCII
uchar   name[8];           // имя, формат ASCII
uchar   net_adr1;          // сетевой адрес порта последовательного обмена COM1
uchar   status;
uchar[nn];
ushort  summ;
}TStatus;
```

**Ответ** – пакет “Запрос выполнен” или «Запрос ошибочен».

### 3.4 Время и дата

1) Формат запроса «Чтение времени»

0	1	2	3
адр	код	CRC	

Поле «код»= 0x07. – // READ\_TIME

**Ответ** – структура TComTime.

```
typedef struct                // Структура запроса "Установка времени"
{                             // Ответ на запрос "Чтение времени"
    uchar net_adr;
    uchar command;
    TTime time;
    ushort summ;
}TComTime;
```

Параметры времени и даты- см. Таблица 3.

2) Формат данных времени и даты в структуре TTime.

**Таблица 3** Параметры времени и даты

Сообщение			Общий размер 8 байт
формат	№ байта	данные	размер
unsigned char	2	секунды	1 байт
unsigned char	3	минуты	1 байт
unsigned char	4	часы	1 байт
unsigned char	5	день недели	1 байт
unsigned char	6	месяц	1 байт
unsigned char	7	год	1 байт
unsigned char	8	не используется	1 байт
unsigned char	9	не используется	1 байт

```
typedef struct
{
    uchar sec;
    uchar min;
    uchar hour;
    uchar day;
    uchar month;
    uchar year;
    uchar dummy;
    uchar sum;
}TTime;
```

## 4 Работа с температурами

### 4.1 Чтение температур по номеру

Запрос считывания показаний датчика температуры по номеру датчика.

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Датчика	zero	CRC	

Поле «код»= 0x01. //READ\_TEMPN

Поле «№ Датчика» - номер датчика температуры от 1 до 8.

```
typedef struct      // Структура команды "Чтение температуры"
{
uchar   net_adr;
uchar   command;
uchar   numer_t;
uchar   dummy;
ushort  summ;
}TComReadTempN;
```

```
typedef struct      // Структура ответа на команду "Чтение температуры"
{
uchar   net_adr;
uchar   command;
uchar   numer_t;
uchar   dummy;
float   temp;
ushort  term_KZ;      // Ошибки датчиков – короткое замыкание
ushort  term_XX;     // Ошибки датчиков - обрыв
ushort  summ;
}TComTempN;
```

## 5 Каналы ввода-вывода

### 5.1 Чтение дискретных входов

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x03. // READ\_DIN

```
typedef struct      // Структура команды "Чтение дискретных входов "
{
uchar   net_adr;
uchar   command;
ushort  summ;
}TComReadTempN;
```

```
typedef struct // Структура ответа на команду "Чтение дискретных входов "
{
uchar net_adr;
uchar command;
uchar chanal_DIN[4];
ushort summ;
}TComReadDIN;
```

## 5.2 Чтение дискретных выходов

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x04. // READ\_DOUT

```
typedef struct // Структура команды "Чтение дискретных выходов "
{
uchar net_adr;
uchar command;
ushort summ;
}TComReadTempN;
```

```
typedef struct // Структура ответа на команду "Чтение дискретных выходов "
{
uchar net_adr;
uchar command;
uchar chanal_DOUT[2];
ushort summ;
}TComReadDOUT;;
```

## 5.3 Чтение аналоговых входов

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x14. // READ\_AIN

```
typedef struct // Структура запроса " Чтение аналоговых входов "
{
uchar net_adr;
uchar command;
ushort summ;
} TComAIN;
```

```
typedef struct // Структура ответа на запрос "Чтение аналоговых входов"
{
uchar net_adr;
uchar command;
uchar dummy;
uchar dummy1;
uchar typ[4]; // Тип каналов AIN
```

```
float  chanal_AIN[4];    // Аналоговые каналы ввода
uchar  dummy2;
uchar  dummy3;
ushort summ;
}TComReadAIN;
```

## 5.4 Чтение аналоговых выходов

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x15. // READ\_AOUT

```
typedef struct          // Структура запроса " Чтение аналоговых выходов "
{
uchar  net_adr;
uchar  command;
ushort summ;
} TComAOUT;
```

```
typedef struct          // Структура ответа на запрос "Чтение аналоговых выходов"
{
uchar  net_adr;
uchar  command;
ushort  chanal_AOUT[4];
ushort summ;
}TComReadAOUT;
```

## 5.5 Работа с цифровыми манометрами

1) Запрос «Чтение параметров настройки и данных цифровых манометров»

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x51. // READ\_MP

```
typedef struct          // Структура запроса " Чтение параметров настройки и данных
{                          // цифровых манометров»
uchar  net_adr;
uchar  command;
ushort summ;
} TComMP;
```

```
typedef struct          // Структура ответа на запрос «Чтение настроек и данных цифровых
{                          манометров MP»
uchar  net_adr;
uchar  command;
ushort controlMP;        // Опрос манометров – (номер бита - номер манометра)
uchar  data_MP[16];      // Прочитанные данные (давления) с манометров
ushort no_MP;           // Ошибки – нет манометра (номер бита - номер манометра)
ushort error_MP;        // Ошибки в работе манометров (номер бита - номер манометра)
```

```
ushort    summ;
}TComReadMP;
```

## 6 Параметры и коэффициенты регулятора

### 6.1 Тип контура регулирования

1) Запрос «Чтение типа контура регулирования»

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x08. // READ\_TIPK

```
typedef struct    // Структура запроса "Чтение типа контура регулирования "
{
uchar   net_adr;
uchar   command;
ushort  summ;
} TComTipK;
```

```
typedef struct    // Структура ответа на запрос "Чтение типа контура регулирования "
{
uchar   net_adr;
uchar   command;
uchar   tipK[3];           // Установленный тип контура регулирования
uchar   tipK_rab[3];       // Рабочий тип контура регулирования
ushort  summ;
}TComReadTipK;
```

### 6.2 Установка функций

1) Запрос «Чтение установок функций»

Считываются одновременно все установки функций – функции РТ, функции контуров регулирования, функции насосов всех контуров, функции контура подпитки.

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x0B. // READ\_READ\_TIPF

```
typedef struct    // Структура команды " Чтение установок функций "
{
uchar   net_adr;
uchar   command;
ushort  summ;
} TComF;
```

```
typedef struct    // Структура ответа на команду "Чтение установок функций»
{
uchar   net_adr;
```

```

uchar  command;
ushort tipNasosPodp;
ushort tipNasos[3];
ushort rtmFunk;
ushort tipFunk[3];
ushort summ;
}TComReadF;

```

## 2) Формат описателя функций контура регулирования

Функции контура регулирования 1 (2, 3) - размер два байта, каждый бит отвечает за состояние функции (1-включена, 0-выключена). Соответствие битов описателя и функций приведено в Таблица 4

**Таблица 4**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ф-ция	f16	f15	f14	f13	f12	f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1

- f1** – функция «контроль обратного сетевого теплоносителя»;
- f2** – функция «быстрое ограничение обратного сетевого теплоносителя»;
- f3** – функция «предел отопления»;
- f4** – функция «ограничение по температуре сетевого теплоносителя»;
- f5** – функция «контроль и регулирование обратного сетевого теплоносителя»;
- f6** – функция «включение контура регулирования по внешнему сигналу»;
- f7** – функция «работа снижения и прогрева по усредненной наружной температуре»;
- f8** – функция «усреднение наружной температуры»;
- f9** – функция «расчет снижения и прогрева»;
- f10** – функция «годовая программа»;
- f11** – функция «модифицирование характеристики регулирующего клапана»;
- f12** – функция «защита при аварии насоса»;
- f13** – функция «защита от замораживания»;
- f14** – функция нет;
- f15** – функция «предел отопления “открытие”»;
- f16** – функция «упреждающее дифференцирование».

## 3) Формат описателя функций насосов контура регулирования

Функции насосов контура регулирования 1 (2, 3) - размер два байта, каждый бит отвечает за состояние функции (1-включена, 0-выключена). Соответствие битов описателя и функций приведено в Таблица 5.

**Таблица 5**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ф-ция	f16	f15	f14	f13	f12	f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1

- f1** – функция «работа насосов по температуре»;
- f2** – функция «работа насосов по пределу отопления»;
- f3** – функция «защита насоса от сухого хода»;
- f4** – функция «прокрутка насосов в режиме СТОП»;
- f5** – функция «резервирование насосов»;
- f6** – функция нет
- f7** – функция «инверсия входа датчика охраны»;
- f8** – функция «сброс ошибок насосов»;
- f9** – функция «работа насосов по времени»;
- f10** – функция нет;
- f11** – функция нет;
- f12** – функция нет;

f13 – функция нет;  
 f14 – функция нет;  
 f15 – функция нет;  
 f16 – функция нет;

4) Формат описателя функций РТМ

Функции насосов контура регулирования 1 (2, 3) - размер два байта, каждый бит отвечает за состояние функции (1-включена, 0-выключена). Соответствие битов описателя и функций приведено в Таблица 6.

**Таблица 6**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ф-ция	f16	f15	f14	f13	f12	f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1

f1 – функция «пароль»;  
 f2 – функция «дистанционное управление»;  
 f3 – функция «работа выхода авария»;  
 f4 – функция «контроль работы обмена по I<sup>2</sup>C»;  
 f5 – Функция "SMS-сообщение";  
 f6 – Функция "Перевод времени по сезонам";  
 f7 – Функция "Выход "Авария" импульсный";  
 f8 – функция нет;  
 f9 – функция нет;  
 f10 – функция нет;  
 f11 – функция нет;  
 f12 – функция нет;  
 f13 – функция нет;  
 f14 – функция нет;  
 f15 – функция нет;  
 f16 – функция нет;

5) Формат описателя функций насосов подпитки

Функции насосов контура подпитки - размер два байта, каждый бит отвечает за состояние функции (1-включена, 0-выключена). Соответствие битов описателя и функций приведено в Таблица 7.

**Таблица 7**

№ бита	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ф-ция	f16	f15	f14	f13	f12	f11	f10	f9	f8	f7	f6	f5	f4	f3	f2	f1

f1 – функция «работа насоса подпитки»;  
 f2 – функция «работа подпитки»;  
 f3 – функция «защита насоса от сухого хода»;  
 f4 – функция нет;  
 f5 – функция «резервирование насосов»;  
 f6 – функция нет;  
 f7 – функция «инверсия входа датчика охраны»;  
 f8 – функция «сброс ошибок насосов подпитки»;  
 f9 – функция «работа насосов по времени»;  
 f10 – функция «инверсные уровни датчиков подпитки»;  
 f11 – функция «работа датчиков подпитки типа “поплавок”»;  
 f12 – функция нет;  
 f13 – функция нет;  
 f14 – функция нет;  
 f15 – функция нет;  
 f16 – функция нет;

### 6.3 Коэффициенты температурного графика

1) Запрос « Чтение коэффициентов температурного графика»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x20. // READ\_KGR

Поле «№ Контура» - номер контура РТ от 0 до 2.

```
typedef struct // Структура запроса " Чтение коэф-тов температурного графика»
{
uchar net_adr;
uchar command;
uchar kontur;
uchar dummy;
ushort summ;
} TComKGr;
```

```
typedef struct // Ответ на запрос "Чтение коэффициентов температурного графика»
{
uchar net_adr;
uchar command;
uchar kontur;
uchar status;
TEBlockGr koeff;
ushort summ;
}TComReadKGr;
```

```
typedef struct
{
float temp_osn;
float kru_1;
float kru_2;
float temp_p;
float temp_max;
float temp_min;
float temp_niz1;
float temp_niz2;
float temp_vozd;
float kp;
float kmax;
uchar dummy;
uchar summ;
}TEBlockGr;
```

**Таблица 8**–Коэффициенты температурного графика

Сообщение общий размер 46 байт			
формат	Коэф-т	данные	размер
float	temp_osn	температура основания графика	4 байта
float	kru_1	крутизна 1-го участка	4 байта
float	kru_2	крутизна 2-го участка	4 байта
float	temp_p	температура точки перегиба	4 байта

float	temp_max	температура максимальная.	4 байта
float	temp_min	температура минимальная	4 байта
float	temp_niz1	температура снижения 1	4 байта
float	temp_niz2	температура снижения 2	4 байта
float	temp_vozd	температура воздуха в помещении	4 байта
float	kp	не используется	4 байта
float	kmax	не используется	4 байта
unsigned char	dummy	не используется	1 байт
unsigned char	summ	сгс8	1 байт

## 6.4 Коэффициенты ПИД-регулятора

1) Запрос «Чтение коэффициентов ПИД-регулятора»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x21. // READ\_KREG

Поле «№ Контура» - номер контура РТ от 0 до 2.

```
typedef struct          // Структура запроса " Чтение коэф-тов ПИД-регулятора"
{
uchar   net_adr;
uchar   command;
uchar   kontur;
uchar   dymmy;
ushort  summ;
} TComKReg;
```

```
typedef struct          // Структура ответа на запрос "Чтение коэф-тов ПИД-регулятора"
{
uchar   net_adr;
uchar   command;
uchar   kontur;
uchar   status;
TEBlockReg koeff;
ushort  summ;
} TComReadKReg;
```

```
typedef struct
{
float    t_int;          // Время интегрирования
float    t_dif;          // Время дифференцирования
float    k_prop;         // Коэффициент пропорциональный
float    k_dif;          // Коэффициент дифференциальный
float    u_max;          // Максимальное значение
float    u_min;          // Минимальное значение
float    dummy1;
float    k_dif_pred;     // Коэффициент предварительного дифференцирования
float    term_dif_pred;  // Номер датчика для пред дифф.
uchar    dummy;          // подравняли до чет
```

```
uchar    summ;
} TEBlockReg;
```

2) Описания коэффициентов ПИД-регулятора в Таблица 9.

**Таблица 9**–Коэффициенты ПИД – регулятора в структуре TEBlockReg

сообщение			общий размер 38 байт
формат	№ байта	данные	размер
float	t_int	время интегрирования	4 байта
float	t_dif	время дифференцирования	4 байта
float	k_prop	диапазон пропорциональности	4 байта
float	k_dif	коэф. усил. диффер-ой части	4 байта
float	u_max	U max – Ограничение макс. значения	4 байта
float	u_min	U min - Ограничение мин. значения	4 байта
float	dummy1	не используется	4 байта
float	k_dif_pred	кэфф-нт предварительного диффер-ния	4 байта
float	term_dif_pred	номер датчика для пред-го диффер-ния	4 байта
unsigned char	dummy	не используется	1 байт
unsigned char	summ	crc8	1 байт

### 6.5 Коэффициенты функции «Ограничение обратной»

1) Запрос «Чтение коэффициентов функции «Ограничение обратной»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x22. // READ\_KOBR

Поле «№ Контура» - номер контура РТ от 0 до 2.

```
typedef struct          // Структура запроса «Чтение коэф-тов функции «Ограничение {
uchar  net_adr;        // обратной»
uchar  command;
uchar  kontur;
uchar  dymmy;
ushort summ;
} TComKObr;
```

```
typedef struct          //
{                       //
uchar  net_adr;        // Ответ на запрос "«Чтение коэф-тов функции «Ограничение
uchar  command;        // обратной»
uchar  kontur;
uchar  status;
TEBlockObr koeff;
ushort summ;
}TComSetKObr;
```

```
typedef struct
{
float    t_obr_0;
```

```

float    k_obr;
float    t_obr_min;
float    time_obr;
float    delta_obr;
float    mo_obr_ur;
float    mo_obr_time;
float    k_vlp;
uchar    dummy;    // подравняли до чет
uchar    summ;
} TEBlockObr;

```

2) Описание коэффициентов функции «Ограничение обратки» в Таблица 10.

**Таблица 10**–Коэффициенты функции «Ограничение обратки» в структуре TEBlockObr

сообщение			общий размер 30 байт
формат	Коэфф-ты	Описание	размер
float	t_obr_0	Температура графика обратки при tn=0	4 байта
float	k_obr	Крутизна графика обратки	4 байта
float	t_obr_min	Минимальное значение графика обратки	4 байта
float	time_obr	Постоянная времени ограничения	4 байта
float	delta_obr	Гистерезис функции ограничения обратки	4 байта
float	mo_obr_ur	Уровень мгновенного ограничения обратки	4 байта
float	k_vlp	Коэффициент влияния обратки на подачу	4 байта
unsigned char	dummy	не используется	1 байт
unsigned char	summ	сгс8	1 байт

## 6.6 Коэффициенты функции «Ограничение по сети»

1) Запрос «Чтения коэф-тов функции «Ограничение по сети»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x23. // READ\_KOGRС

Поле «№ Контура» - номер контура РТ от 0 до 2.

```

typedef struct    // Структура запроса «Чтения коэф-тов функции «Ограничение {
uchar    net_adr;    // по сети»
uchar    command;
uchar    kontur;
uchar    dymmy;
ushort    summ;
} TComOgrC;

```

```

typedef struct    //
{    // Ответ на запрос «Чтения коэф-тов функции «Ограничение по сети»
uchar    net_adr;
uchar    command;

```

```

uchar   kontur;
uchar   status;
TEBlockOgrC koef;
ushort  summ;
}TComSetKOgrC;

```

```

typedef struct
{
float    ogrC_term;
float    ogrC_delta;
uchar    dummy;           // подравняли до чет
uchar    summ;
}TEBlockOgrC;

```

2) Описания коэф-нтов функции «Ограничение по сети» в Таблица 11.Таблица 1.

**Таблица 11** Коэффициенты функции «Ограничение по сети»

сообщение			общий размер 10 байт
формат	Коэфф-ты	Описание	размер
float	ogrC_term	Номер латчика температуры функции	4 байта
float	ogrC_delta	Минимальная разность температур	4 байта
unsigned char	dummy	не используется	1 байт
unsigned char	summ	crc8	1 байт

### 6.7 Коэффициенты "Функции компенсации "

1) Запрос «Чтение коэффициентов "Функции компенсации»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x24. // READ\_KKOMPENS

Поле «№ Контура» - номер контура РТ от 0 до 2.

```

typedef struct           // Структура запроса «Чтение коэф-тов "Функции компенсации "
{
uchar   net_adr;
uchar   command;
uchar   kontur;
uchar   dymmy;
ushort  summ;
}TComKKompens;

```

```

typedef struct          //
{                       // Ответ на запрос «Чтение коэффициентов "Функции компенсации"
uchar   net_adr; //
uchar   command;
uchar   kontur;
uchar   status;
TEBlockKompens koef;
ushort  summ;
}TComSetKKompens;

```

```

typedef struct
{
float      kompens_tn; // Постоянная компенсации по наружному воздуху
float      kompens_tv; // Постоянная компенсации по воздуху помещения
float      kompens_K; // Коэффициент влияния температуры воздуха на график
uchar      dummy;     // подравняли до чет
uchar      summ;
} TEBlockKompens;

```

## 6.8 Коэффициенты функции «Снижение и прогрев»

1) Запрос «Чтение коэффициентов функции «Снижение и прогрев»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x25. // READ\_KBLOCKREDUCE

Поле «№ Контура» - номер контура РТ от 0 до 2.

```

typedef struct // Структура запроса «Чтение коэф-тов функции«Снижение и прогрев»
{
uchar net_adr;
uchar command;
uchar kontur;
uchar dymmy;
ushort summ;
} TComBlockReduce;

```

```

typedef struct // Структура ответа «Чтение коэф-тов функции«Снижение и прогрев»
{
uchar net_adr; //
uchar command;
uchar kontur;
uchar status;
TEBlockNightReduce koeff;
ushort summ;
}TComSetBlockReduce;

```

```

typedef struct
{
float nred_tempN_0; // Наружная температура при которой снижение =0%
float nred_tempN_max; // Наружная температура при которой снижение =100%
float nred_timeN_0; // Время сокращения дневного цикла при tn=0°C
float nred_time_kru; // Крутизна для расчета времени сокращения дневного цикла
float warUp_timeN_0; // Время утреннего прогрева при tn=0°C
float warUp_kru; // Наклон характеристики изменения утреннего прогрева
float warUp_tempN_max; //tn при которой утренний прогрев максимальный
float warUp_tempN_0; //Наружная температура при которой утренний прогрев =0
float warUp_temp_max; //Максимальная температура утреннего прогрева
uchar dummy; // подравняли до чет
uchar summ;
} TEBlockNightReduce;

```

## 6.9 Коэффициенты функции «Функция защиты»

1) Запрос «Чтение коэф-тов «Функция защиты»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x26. // READ\_KBLOCKPROTECT

Поле «№ Контура» - номер контура РТ от 0 до 2.

```
typedef struct // Структура запроса «Чтение коэф-тов «Функция защиты»
{
uchar net_adr;
uchar command;
uchar kontur;
uchar dummy;
ushort summ;
} TComBlockProtect;
```

```
typedef struct //
{ // Структура ответа на запрос «Чтение коэф-тов"Функция защиты "
uchar net_adr;
uchar command;
uchar kontur;
uchar status;
TEBlockProtect koeff;
ushort summ;
}TComSetBlockProtect;
```

```
typedef struct
{
float errorTemp;
float levelKompens; // Макс. уровень компенсации (обратка, в помещении, ветер)
float termFrost; // Номер датчика для функции "Защита от замораживания"
float tempFrostNasos; // Температура срабатывания насоса в СТОП
float tempFrostKOT; // Температура срабатывания КОТ в СТОП
uchar dummy; // подравняли до чет
uchar summ;
} TEBlockProtect;
```

## 6.10 Коэффициенты функции «Предел отопления»

1) Запрос «Чтение коэф-тов функции «Предел отопления»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x27. // READ\_KBLOCKEKO\_LIMIT

Поле «№ Контура» - номер контура РТ от 0 до 2.

```
typedef struct // Структура запроса «Чтение коэф-тов функции «Предел отопления»
{
uchar net_adr;
```

```

uchar  command;
uchar  kontur;
uchar  dymmy;
ushort summ;
} TComBlockEKO_limit;

```

```

typedef struct      //
{      //Структура ответа на запроса «Чтение коэф-тов функции «Предел отопления»
uchar  net_adr; //
uchar  command;
uchar  kontur;
uchar  status;
TEBlockFunk koeff;
ushort summ;
}TComSetBlockEKO_limit;

```

```

typedef struct
{
float level;
float gisterezis;
uchar term;
uchar time;
uchar status;      // подравняли до чет
uchar summ;
} TEBlockFunk;

```

2) Описание коэффициентов функции «Предел отопления» в Таблица 12.

**Таблица 12** Коэффициенты функции «Предел отопления»

сообщение			общий размер 12 байт
формат	Коэфф-т	Описание	размер
float	level	Уровень температуры	4 байта
float	gisterezis	гистерезис	4 байта
unsigned char	term	Установки см. примечание *	1 байт
unsigned char	time	время см. примечание **	1 байт
unsigned char	status	не используется	1 байт
unsigned char	summ	crc8	1 байт

**Примечание:**

\* "установки" – биты 0-3 № датчика температуры.

Бит 7: если равен 0 – функция срабатывает, если значение температуры с датчика больше значения "уровень температуры", если равен 1 – функция срабатывает, если значение температуры с датчика меньше значения "уровень температуры".

\*\* "время" – значение в часах через какое время должна сработать функция при выполненных условиях.

### 6.11 Коэффициенты функции «Функция насосов подпитки»

1) Запрос «Чтение коэф-тов «Функция насосов подпитки»

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x29. // READ\_FunkPODPITKA

```
typedef struct // Структура запроса «Чтение коэф-тов «Функция насосов подпитки»
{
uchar net_adr;
uchar command;
ushort summ;
} TComFPodpitka;
```

```
typedef struct // Структура ответа «Чтение коэффициентов "Функция насосов подпитки»
{
uchar net_adr;
uchar command;
uchar dummy;
uchar status;
TNasos koeff;
ushort summ;
}TComSetFPodpitka;
```

```
typedef struct
{
float time_rez; // Время работы до смены насоса
float time_on; // Время задержки включения насоса
float time_off; // Время задержки выключения насоса
float time_prokr; // Время включения прокрутки насосов
uchar dummy; // подравняли до чет
uchar summ;
} TNasos;
```

### **6.12 Коэффициенты функции «Коэффициенты насосов контуров»**

1) Запрос «Чтение коэффициентов функции «Коэффициенты насосов контуров»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№ Контура	zero	CRC	

Поле «код»= 0x2B. // READ\_NASOS

Поле «№ Контура» - номер контура РТ от 0 до 2.

```
typedef struct // Структура запроса «Чтение коэф-тов функции «Коэффициенты насосов
{ // контуров»
uchar net_adr;
uchar command;
uchar kontur;
uchar dymmy;
ushort summ;
} TComNasos;
```

```
typedef struct //
{ //
uchar net_adr; //Ответ на запрос «Чтение коэф-тов функции «Коэф-ты насосов
uchar command; // контуров»
uchar kontur;
uchar status;
```

```
TNasos koeff;
ushort summ;
}TComSetNasos;
```

```
typedef struct
{
float time_rez;           // Время работы до смены насоса
float time_on;           // Время задержки включения насоса
float time_off;          // Время задержки выключения насоса
float time_prokr;        // Время работы прокрутки насоса
uchar dummy;            // подравняли до чет
uchar summ;
} TNasos;
```

### 6.13 Коэффициенты функции " Работа насосов по температуре"

1) Запрос «Чтение коэффициентов функции «Работа насосов по температуре»  
 Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№К	zero	CRC	

Поле «код»- 0x2D; // READ\_NTEMP  
 №К - номер контура от 0 до 2;

```
typedef struct // Структура запроса «Чтение коэффициентов функции «Работа
{ // насосов по температуре»
uchar net_adr;
uchar command;
uchar kontur;
uchar dymmy;
ushort summ;
} TComNTemp;
```

```
typedef struct //
{ //
uchar net_adr; // Ответ на запроса "Чтение коэффициентов функции «Работа
uchar command; // насосов по температуре»
uchar kontur;
uchar status;
TEBlockFunk koeff;
ushort summ;
}TComSetNTemp;
```

```
typedef struct
{
float level;
float gisterezis;
uchar term;
uchar time;
uchar dymmy;
uchar summ;
} TEBlockFunk;
```

- 2) Описание коэффициентов функции «Работа насосов по температуре» в Таблица 13.

**Таблица 13 Коэффициенты функции «Работа насосов по температуре»**

сообщение			общий размер 12 байт
формат	Коэфф-т	Описание	размер
float	level	Уровень температуры	4 байта
float	gisterezis	Гистерезис	4 байта
unsigned char	term	Установки см. Примечание *	1 байт
unsigned char	time	Задержка срабатывания функции	1 байт
unsigned char	dymmy	не используется	1 байт
unsigned char	summ	crc8	1 байт

**Примечание:**

\* " level " – биты 0-3 № датчика температуры.

Бит 7: если равен 0 – функция срабатывает, если значение температуры с датчика больше значения "level", если равен 1 – функция срабатывает, если значение температуры с датчика меньше значения " level ".

## 7 Чтение сервисных параметров регулятора

### 7.1 Чтение ошибок и предупреждений

Формат запроса 4 байта.

0	1	2	3
адр	код	CRC	

Поле «код»= 0x06. – чтение ошибок и предупреждений// READ\_ERROR

```
typedef struct // Структура запроса "Чтение ошибок и предупреждений"
{
uchar net_adr;
uchar command;
ushort summ;
}TComError;
```

```
typedef struct // Структура ответа на запрос "Чтение ошибок и предупреждений"
{
uchar net_adr;
uchar command;
ushort error;
ushort warning[4]; // Предупреждения контуров 1-3 и общие предупреждения
PT
ushort summ;
}TComReadError;
```

## 8 Архивы

### 8.1 Чтение архива событий

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№записи	кол-во записей	CRC	

Поле «код»- 0x32; // READ\_ARH\_SETTING

Поле «№записи» - номер записи архива от 0 до 31.

Поле «кол-во записей» - количество считываемых записей

Формат ответа 14 байт (при считывании одной записи):

0	1	2	3	4-11	12	13
адр	код	№записи	кол-во записей	сообщение	CRC	

Поле «сообщение» см. Таблица 14;

**Таблица 14** Чтение архива событий

Сообщение			общий размер 8 байт
формат	№ байта	данные	размер
unsigned char	4	минуты	1 байт
unsigned char	5	часы	1 байт
unsigned char	6	день	1 байт
unsigned char	7	месяц	1 байт
unsigned int	8-9	evens *	2 байт
unsigned int	10-11	№ последней записи	2 байт

биты параметра evens															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
не используются					power	prog	события контура 3			события контура 2			события контура 1		

Поле «**события**» контура 1, 2, 3 могут принимать значения:

0 – переключение РТ в ручной режим работы;

1 – переключение РТ в режим работы “Стоп”;

2 – переключение РТ в режим работы “Нормальный”;

3 – переключение РТ в режим работы “Снижение 1”;

4 – переключение РТ в режим работы “Снижение 2”;

5 – переключение РТ в режим работы “Программный”.

Поле «**prog**» может принимать значения:

0 –РТ в режиме работы «Автоматический»;

1 – РТ в режиме работы «Программирование»;

Поле «**power**» может принимать значения:

0 –питания РТ включено;

1 –питания РТ выключено;

### 8.2 Чтение архива посещений

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№записи	кол-во записей	CRC	

Поле «код»- 0x33; // READ\_ARH\_FIX  
 Поле «№записи» - номер записи архива от 0 до 55.  
 Поле «кол-во записей» - количество считываемых записей

Формат ответа 16 байт (при считывании одной записи):

0	1	2	3	4-13	14	15
адр	код	№записи	кол-во записей	сообщение	CRC	

Поле «сообщение» см Таблица 15.

**Таблица 15** Чтение архива посещений

сообщение			общий размер 10 байт
формат	№ байта	данные	размер
unsigned char	4	секунды	1 байт
unsigned char	5	минуты	1 байт
unsigned char	6	часы	1 байт
unsigned char	7	день	1 байт
unsigned char	8	месяц	1 байт
unsigned char	9	год	1 байт
unsigned char	10	не используется	1 байт
unsigned char	11	не используется	1 байт
unsigned char	12	посещение	1 байт
unsigned char	13	crc8	1 байт

Поле «посещение» принимает значения:  
 0 – отметка о окончании обслуживания;  
 1 – отметка о начале обслуживания.

### 8.3 Чтение архива ошибок

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	№записи	кол-во записей	CRC	

Поле «код»- 0x34; // READ\_ARH\_ERR  
 Поле «№записи» - номер записи архива от 0 до 31.  
 Поле «кол-во записей» - количество считываемых записей

Формат ответа 26 байт (при считывании одной записи):

0	1	2	3	4-23	24	25
адр	код	№записи	кол-во записей	сообщение	CRC	

Поле «сообщение» см. Таблица 16.

**Таблица 16** Чтение архива ошибок и предупреждений

сообщение			общий размер 10 байт
формат	№ байта	данные	размер
unsigned char	4	секунды	1 байт
unsigned char	5	минуты	1 байт
unsigned char	6	часы	1 байт
unsigned char	7	день	1 байт
unsigned char	8	месяц	1 байт
unsigned char	9	год	1 байт
unsigned char	10	не используется	1 байт
unsigned char	11	не используется	1 байт

unsigned short	12-13	ошибки	2 байта
unsigned short	14-15	предупреждения контура 1	2 байта
unsigned short	16-17	предупреждения контура 2	2 байта
unsigned short	18-19	предупреждения контура 3	2 байта
unsigned short	20-21	предупреждения контура подпитки	2 байта
unsigned char	22	№ последней записи	1 байт
unsigned char	23	crc8	1 байт

Поле «**ошибки**» - установка битов в 1 является признаком наличия ошибки.

Перечень ошибок:

- 0x01 – Температура CPU выше 50°C";
- 0x02 – Не исправен датчик t°;
- 0x04 – Не исправен датчик давления;
- 0x10 – Авария контура 1;
- 0x20 – Авария контура 2;
- 0x40 – Авария контура 3;
- 0x80 – Перезапуск часов;
- 0x100 – Сбой часов;
- 0x200 – Тип контура по умолчанию;
- 0x400 – Режим работы по умолчанию;
- 0x1000 – Авария датчика контрольной температуры;
- 0x2000 – Авария датчика температуры доп. выхода;
- 0x4000 – Ошибка CRC COM 0;
- 0x8000 – Ошибка CRC COM 1.

Поле «**предупреждения контура 1(2,3)**» - установка битов в 1 является признаком наличия предупреждения. Перечень предупреждений:

- 0x01 – Ошибка чтения годовой программы;
- 0x02 – Ошибка чтения недельной программы;
- 0x04 – Ошибка работы функции «Ограничение по сети»;
- 0x08 – Ошибка работы функции «Контроль обратки»;
- 0x010 – Ошибка работы функции «Мгновенное ограничение по обратке»;
- 0x020 – Ошибка работы насоса;
- 0x040 – Ошибка работы основного насоса;
- 0x080 – Ошибка работы резервного насоса;
- 0x0100 – Ошибка работы функции «Упреждающее дифференцирование»;
- 0x0200 – Ошибка работы функции «Предел отопления»;
- 0x0400 – Ошибка работы функции «Защита от замораживания»;
- 0x0800 – Ошибка работы функции «Работа насоса по T».

Поле «**предупреждения контура подпитки**»- установка битов в 1 является признаком наличия предупреждения. Перечень предупреждений:

- 0x020 – Ошибка работы насоса;
- 0x040 – Ошибка работы основного насоса;
- 0x080 – Ошибка работы резервного насоса.

## 8.4 Параметры архивов температур и давлений ”

- 1) Запрос «Чтение параметров архивов T и P»

Формат запроса 6 байта

0	1	2	3	4	5
адр	код	архив	zero	CRC	

Поле «код»= 0x37. // READ\_ARH\_CONTROL

Поле «архив» – тип архива: 0 - архив Т, 1 - архив Р.

```
typedef struct // Структура запроса «Чтение параметров архивов Т и Р»
{
uchar net_adr;
uchar command;
uchar arhiv;
uchar dymmy;
ushort summ;
} TComNTemp;

typedef struct // Структура ответа на запроса "Чтение параметров архивов Т и Р"
{uchar net_adr;
uchar command;
uchar tipArj; // 0 - архив Т, 1 - архив Р
uchar dummy;
unsigned int tec_point; // Указатель текущего адреса точки записи архива
TSettingArjTermo control;
ushort summ;
}TComReadArhControl;

typedef struct
{
uchar thermometers; // Выбор датчиков Т или Р для архивации
uchar interval; // Интервал записи архива
uchar setting; // Бит 0 – пуск архива, бит 1 – разрешения цикла
uchar CrcSum;
}TSettingArjTermo;
```

2) Описание структуры параметров архива TSettingArjTermo

**Поле «thermometers»** - Выбор датчиков Т или Р для архивации: 0 – не архивировать, 1 – архивировать.

**Поле «interval»** - интервал записи архива: от 1 до 59 (минут).

**Поле «setting»** - Управление работой архива: бит 0 – разрешение работы архива, бит 1 – разрешения циклической записи архива, бит 3 – признак того, что прошел один цикл записи архива.

## 8.5 Чтение архивов температур и давлений

1) Формат точки записи архива

Формат точки записи архивов температур и давлений зависит от количества архивируемых параметров (температур или давлений). Постоянная и всегда присутствующая часть – время записи точки архива и минимум один параметр. Формат точки архива с одним параметром показан на Рисунок 4.

**Рисунок 4**

Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7	Байт 8
Минуты	Часы	Дни	Месяцы	Записываемый параметр (float)			

Формат точки архива с четырьмя параметрами показан на Рисунок 5.

**Рисунок 5**

Байт 1	Байт 2	Байт 3	Байт 4	Байты 5-8	Байты 9-12	Байт 13-16	Байт 17-20
Минуты	Часы	Дни	Месяцы	Параметр 1	Параметр 2	Параметр 3	Параметр 4

Для считывания точек архивов температур и давлений используется запрос считывания памяти EEPROM.

Архив давлений находится по адресам EEPROM: 0x2000 – 0x27FF.

Архив температур находится по адресам EEPROM: 0x2800 – 0xFFFF.

При считывании архивов следует учитывать признак наличия циклической записи архива и состояние указателя текущего адреса точки записи архива.

## ПРИЛОЖЕНИЕ А. Схемы подключения РТ

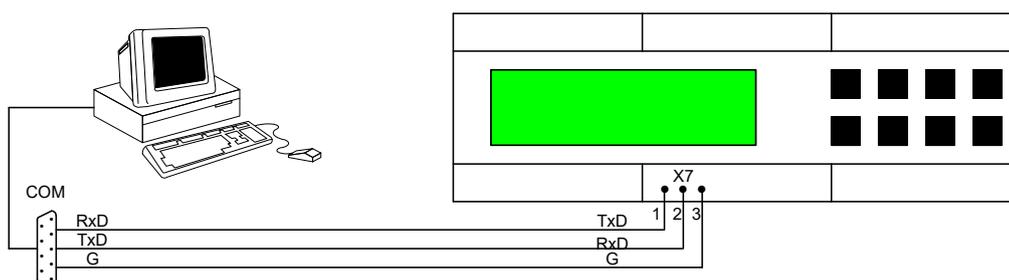


Рис1. Схема подключения РТМ-03 к ПЭВМ по каналу связи типа RS232.

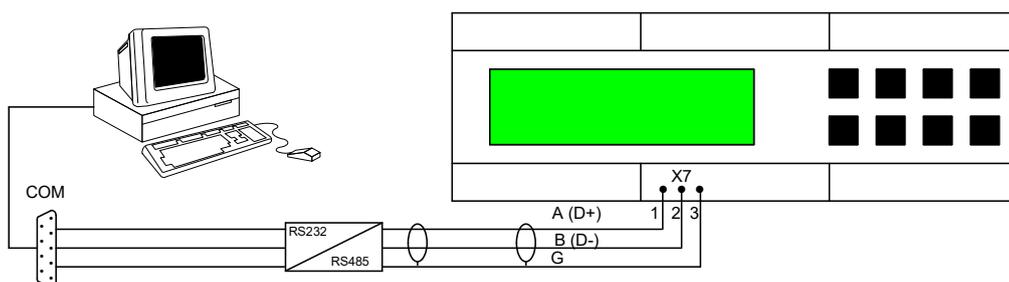


Рис2. Схема подключения РТМ-03 к ПЭВМ по каналу связи типа RS485.

## ПРИЛОЖЕНИЕ Б Подсчет контрольной суммы

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 С. 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, 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, 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 ^ tblCRClo[idx];
    }
}

```

```
        CRClo=tblCRClo[idx];  
    }  
    return ((CRClo << 8) | CRChi);  
}
```

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.

# ГРАН

СИСТЕМА-С

Республика Беларусь

220141, г. Минск, ул. Ф.Скорины, 54а,

Тел./ факс: (017) 373-85-82

E-mail: [info@strumen.com](mailto:info@strumen.com)

<http://www.strumen.com>

Отдел сбыта:

(017) 351-41-87

(017) 374-81-89

(029) 158-93-37

Отдел технического обслуживания:

(017) 355-58-09

(029) 365-82-09

Консультации по вопросам применения и вопросам выполнения пуско-наладочных работ:

(017) 373-85-82

(029) 683-20-99

(029) 728-86-41