

**ПРИБОР ПОКАЗЫВАЮЩИЙ  
КП1М**

**Протокол обмена**

**2.556.103 Д**

## СОДЕРЖАНИЕ

1 Введение	3
2 Формат сообщения MODBUS-ASCII	3
3 Формат сообщения MODBUS-RTU	4
4 Описание команд	6
5 Описание регистров	9

## 1 Введение

Протокол **MODBUS** предназначен для связи между приборами, объединенными в сеть с организацией обмена типа «**MASTER-SLAVE**». При этом только **MASTER** может инициировать сообщения, называемые ЗАПРОС, на который **SLAVE** формирует сообщение, называемое ОТВЕТ.

Обмен сообщениями осуществляется в режиме последовательной передачи. Параметры последовательного обмена должны быть одинаковы для всех объектов сети **MODBUS**: 1 старт-бит, 8 бит данных, 1 стоп-бит.

Протокол **MODBUS** определяет структуру сообщений ЗАПРОС и ОТВЕТ.

## 2 Формат сообщения MODBUS-ASCII

2.1 Каждый байт в сообщении посылается как два знака ASCII. Главным преимуществом данного режима является то, что он позволяет иметь различные интервалы времени между посылками, без появления ошибки.

Формат сообщения представлен на рисунке 1.

Любой байт адреса, команды или данных представляется в виде двух ASCII знаков: в числе 62h знаки «6» и «2» представляются и передаются, как 36h и 32h.

Старт	Адрес	Команда	Данные	КС	Стоп
1 знак	2 знака	2 знака	N знаков	2 знака	2 знака

**Рисунок 1** – Формат сообщения MODBUS-ASCII

2.2 Сообщения начинаются маркером начала сообщения – знаком двоеточия (:) (3Ah), заканчиваются маркером конца сообщения – двумя байтами (0Dh и 0Ah).

2.3 Поле адреса содержит два знака. Адреса **SLAVE** находятся в десятичном диапазоне 0-247. Адрес 0 присваивается **SLAVE**, которые должны отвечать на ЗАПРОС с любым адресом.

При формировании запроса **MASTER** устанавливает в поле адреса адрес запрашиваемого **SLAVE**, в ответе в поле адреса возвращается адрес **SLAVE**.

2.4 Поле команд содержит два знака - код команды. В ЗАПРОСЕ код команды указывает **SLAVE**, какое действие выполнить.

В ОТВЕТЕ поле команды служит для подтверждения приема ЗАПРОСА.

В случае приема без ошибок поле команды ОТВЕТА повторяется код команды.

При ошибке поле команды содержит признак ошибки, сформированный, как код команды, в старшем бите которого значение 1. Дополнительно в поле данных ОТВЕТА помещается уникальный код ошибки.

Например, в поле команды запроса содержится код команды 03h (0000 0011 b) – команда «Считать информацию регистров настроек».

Если **SLAVE** без ошибок принял ЗАПРОС, то в поле команды ОТВЕТ повторяет исходный код команды 03h (0000 0011 b), в случае ошибки ОТВЕТ содержит признак ошибки и код ошибки (см. рисунок 2).

ОТВЕТ	
Поле адреса SLAVE	05h (0000 0101 b)
Поле команды	83h (1000 0011 b)
Поле данных	07h (0000 0111 b)
КС	71h

Рисунок 2 – Структура сообщения ОТВЕТ в случае ошибки

2.5 Поле данных содержит:

– в ЗАПРОСЕ – дополнительную информацию, которую использует SLAVE для выполнения команды;

– в ОТВЕТЕ – запрашиваемые данные при отсутствии ошибок,, а в случае ошибки – код ошибки (смотри рисунок 2).

2.6 Поле контрольной суммы (КС) содержит два знака.

Значение КС вычисляется передающим устройством и добавляется к сообщению. Принимающее устройство во время приема сообщения вычисляет КС и сравнивает вычисленное и принятое значения. Несовпадение этих двух значений является ошибкой.

Пример расчета КС:

Сложить все байты в сообщении, за исключением маркеров начала и конца сообщения в однобайтном поле, исключая перенос.

Выполнить байтовую операцию вычитания из 00h значения получившееся в п.1

Пример расчета контрольной суммы

Адрес	02h	
Поле команды	01h	
Поле данных 1	00h	
Поле данных 2	00h	
Поле данных 3	00h	
Поле данных 4	08h	
Результат п.1	0Bh	
КС(Результат п.2)	F5h	

Рисунок 3 - Расчет контрольной суммы

### 3 Формат сообщения MODBUS-RTU

3.1 Формат сообщения представлен на рисунке 4

Старт	Адрес	Команда	Данные	КС	Стоп
Пауза 3.5 байт	1байт	1 байт	N байт	2 байта	Пауза 3.5 байт

Рисунок 4- Формат сообщения **MODBUS-RTU**

3.2 Сообщения начинаются и заканчиваются паузой длительностью не менее длительности передачи 3.5 байт.

3.3 Поле адреса содержит 1 байт. Адреса **SLAVE** находятся в десятичном диапазоне 0-247. Адрес 0 присваивается **SLAVE**, которые должны отвечать на ЗАПРОС с любым адресом.

При формировании запроса **MASTER** в поле адреса сообщения устанавливается адрес запрашиваемого **SLAVE**, в поле адреса ОТВЕТ возвращает адрес **SLAVE**.

3.4 Описание поля команды смотри в п. 2.4.

3.5 Описание поля данных смотри в п.2.5

3.6 Поле контрольной суммы содержит два байта.

Значение КС вычисляется передающим устройством и добавляется к сообщению. Принимающее устройство во время приема сообщения вычисляет КС и сравнивает вычисленное и принятое значения. Несовпадение этих двух значений является ошибкой.

Контрольная сумма вычисляется по стандарту CRC-16

Контрольная сумма передается в следующем порядке сначала идет младший байт, затем передается старший байт.

Например, если значение контрольной суммы равно 1241h, то сначала передается младший байт 41h затем следует старший байт 12h.

### Пример расчета контрольной суммы

```
unsigned int CRC16(unsigned char *msg, unsigned char len)
    // msg - указатель массив содержащий сообщение
    // len - количество байт в сообщении
```

```
{
register unsigned char CRCHi = 0xFF;
register unsigned char CRCLo = 0xFF;
register unsigned char index;
while (len--)
{
index = CRCHi ^ *msg++;
CRCHi = CRCLo ^ aCRCHi[index];
CRCLo = aCRCLo[index];
}
return (CRCHi * 256 + CRCLo);
}
```

```
unsigned char flash aCRCHi[] = {
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, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40 };
```

```
unsigned char flash aCRCLo[] = {
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 };
```

## 4 Описание команд

4.1 Команда 04h. Считать информацию регистров данных  
Команда 04h служит для чтения регистров данных.

### ЗАПРОС

Определяет адрес **SLAVE**, начальный адрес и число регистров данных, значения которых необходимо считать.

На рисунке 5 приведен пример запроса на считывание регистров данных 2–4 **SLAVE** с адресом 17 (11h):

ЗАПРОС		
Адрес <b>SLAVE</b>		11h
Код команды		04h
Начальный адрес HI		00h
	LO	01h
Число регистров HI		00h
	LO	03h
КС		--

**Рисунок 5** - Структура сообщения ЗАПРОС команды 04h

### ОТВЕТ

Содержит адрес **SLAVE**, код команды, поле регистров данных и число байт в поле регистров данных.

Содержимое регистра является шестнадцатиразрядным числом (два байта).

На рисунке 6 приведен пример ОТВЕТА на ЗАПРОС.

ОТВЕТ		
Адрес <b>SLAVE</b>		11h
Код команды		04h
Число байт		06h
Регистр данных 02h	HI	00h
	LO	0Ah
Регистр данных 03h	HI	00h
	LO	0Bh
Регистр данных 04h	HI	00h
	LO	0Ch
КС		--

**Рисунок 6** – Структура сообщения ОТВЕТ команды 04h

4.2 Команда 03h. **Считать значения регистров настроек**  
**Команда 03h** служит для чтения регистров настроек.

### ЗАПРОС

Определяет адрес **SLAVE**, начальный адрес и число регистров настроек, значение которых необходимо считать.

На рисунке 7 приведен пример ЗАПРОСА на чтение регистров настроек 2–4 **SLAVE** с адресом 17 (11h):

ЗАПРОС		
Адрес <b>SLAVE</b>		11h
Код команды		03h
Начальный адрес	HI	00h
	LO	01h
Число регистров	HI	00h
	LO	03h
КС		--

**Рисунок 7** – Структура сообщения ЗАПРОС команды 03h

#### ОТВЕТ

Содержит адрес **SLAVE**, код команды, поле регистров настроек и число байт в поле регистров настроек.

Содержимое регистра является шестнадцатиразрядным числом (два байта).

На рисунке 8 приведен пример ОТВЕТА на ЗАПРОС.

ОТВЕТ		
<i>Название поля</i>		
Адрес SLAVE		11h
Код команды		03h
Число байт		06h
Регистр настроек 02h	HI	00h
	LO	0Ah
Регистр настроек 03h	HI	00h
	LO	0Bh
Регистр настроек 04h	HI	00h
	LO	0Ch
КС		--

**Рисунок 8** – Структура сообщения ОТВЕТ команды 03h

### 4.3 Команда 10h. Установить значение регистров настроек

**Команда 10h** служит для установки значений регистров настроек.

Регистры настройки **SLAVE** могут иметь статус «только чтение», при попытке установить в них новое значение остаются без изменений.

#### ЗАПРОС

Определяет адрес **SLAVE**, начальный адрес, число регистров настроек, поле регистров настроек и число байт в поле регистров настроек.

На рисунке 9 приведен пример ЗАПРОСА на установку значений регистров настроек 2-4 **SLAVE** с адресом 17 (11h):



ЗАПРОС			
Адрес SLAVE			11h
Код команды			10h
Начальный адрес	HI		00h
	LO		01h
Число регистров	HO		00h
	LO		03h
Число байт			06h
Регистр настройки 02h	HI		00h
	LO		0Ah
Регистр настройки 02h	HI		00h
	LO		0Bh
Регистр настройки 02h	HI		00h
	LO		0Ch
КС			--

**Рисунок 9** – Структура сообщения ЗАПРОС команды 10h

#### ОТВЕТ

Содержит адрес **SLAVE**, код команды, начальный адрес и число регистров в поле регистров настроек.

На рисунке 10 приведен пример ОТВЕТА на ЗАПРОС.

ОТВЕТ			
Адрес slave			11h
Код команды			10h
Начальный адрес	Hi		00h
	Lo		01h
Число регистров	Hi		00h
	Lo		03h
КС			--

**Рисунок 10** – Структура ОТВЕТА команды 10h

## 5 Описание регистров

5.1 Описание регистров настроек приведено в таблице 1

**Таблица 1**

Регистр	Обозначение, содержание регистра	Тип	R/W
0000	<b>HI</b> HI Адрес прибора в сети	uChar	R/W
	<b>LO</b> LO Скорость обмена в сети	uChar	R/W

**Продолжение таблицы 1**

Регистр	Обозначение, содержание регистра	Тип	R/W
0001	<b>HI</b> Тип протокола =0 MODBUS ASCII =1 MODBUS RTU	uChar	R/W
	<b>LO</b> HCX (вид входного сигнала) =0 50П =1 100П =2 50М ( $W_{100} = 1,4260$ ) =3 100М ( $W_{100} = 1,4260$ ) =4 Pt50 =5 Pt100 =6 50М =7 100М =8 градуировка 21 =9 градуировка 23 =10 S =11 K =12 L =13 B =14 A1 =15 J =16 N =17 0-5 мА =18 0-20 мА =19 4-20 мА =20 20-4 мА =21 0-20 мВ =22 0-100 мВ =23 0-1 В =24 $\pm 1$ В =25 0-1000 Ом	uChar	R/W
0002	<b>HI</b> Суточная коррекция времени, с	uChar	R/W
	<b>LO</b> Положение запятой =0 0000 =1 000.0 =2 00.00 =3 0.000	uChar	R/W
0003	<b>HI</b> Период фильтрации 0-10 с	uChar	R/W
	<b>LO</b> Период архивации 0-99 с	uChar	R/W
0004	Нижний предел диапазона измерений $Y_0$	Int	R/W
0005	Резерв		
0006	Верхний предел диапазона измерений $Y_K$	Int	R/W
0007	Резерв		
0008	Нижний предел барграфа (преобразования), $Z_0$	Int	R/W

**Продолжение таблицы 1**

Регистр	Обозначение, содержание регистра	Тип	R/W
0009	Резерв		
0010	Нижний предел барграфа (преобразования), $Z_k$	Int	R/W
0011	Резерв		
0012	<b>HI</b> Флаги режимов работы прибора Бит 0 Корнеизвлечение Бит 1 Компенсация холодного спая Бит 2 Масштабирование	uChar	R/W
	<b>LO</b> Резерв	Char	R/W
0013 - 0014	Значение уставки 1	Float	R/W
0015 - 0016	Значение зоны возврата уставки 1	Float	R/W
0017	<b>HI</b> Флаги уставки 1 Бит 0 тип уставки (=0 меньше, =1 больше) Бит 1 Вид уставки (=0 меньше, =1 больше) Бит 2 Исходное состояние реле (=0 разомкнуто, =1 замкнуто) Бит 3 Состояние реле при обрыве (=0 разомкнуто, =1 замкнуто) Бит 4 Включение уставки (=0 отключена, =1 включена)	uChar	R/W
	<b>LO</b> Резерв	Char	R/W
0018 - 0019	Значение уставки 2	Float	R/W
0020 - 0021	Значение зоны возврата уставки 2	Float	R/W
0022	<b>HI</b> Флаги уставки 2	uChar	R/W
	<b>LO</b> Резерв	uChar	R/W
0023 - 0024	Значение уставки 3	Float	R/W
0025 - 0026	Значение зоны возврата уставки 3	Float	R/W
0027	<b>HI</b> Флаги уставки 3	uChar	R/W
0028 - 0029	Значение уставки 4	Float	R/W
0030 - 0031	Значение зоны возврата уставки 4	Float	R/W
0032	<b>HI</b> Флаги уставки 4	uChar	R/W

5.2 Описание регистров данных приведено в таблице 2 (для ASCII, RTU).

**Таблица 2**

Регистр	Обозначение, содержание регистра	Тип	R/W
0000	<b>HI</b> Секунды	uChar	R/W
	<b>LO</b> Минуты	uChar	R/W
0001	<b>HI</b> Час	uChar	R/W
	<b>LO</b> День недели	uChar	R/W
0002	<b>HI</b> День	uChar	R/W
	<b>LO</b> Месяц	uChar	R/W
0003	<b>HI</b> Год	uChar	R/W
	<b>LO</b> Резерв	uChar	R/W
0004	Флаги ошибки	Unsigned int	R/W
0005	Флаги предупреждений	Unsigned int	R/W
0006	<b>HI</b> Состояния уставок	uChar	R/W
	<b>LO</b> Состояния реле	uChar	R/W
0007 - 0008	Температура датчика холодного спая	Float	R/W
0009 – 0010	Измеренная физическая величина	Float	R/W
0011 – 0012	Измеренное значение	Float	R/W



**Контактная информация:**

**Адрес:** 454047, Россия, Челябинск,  
ул. Павелецкая 2-я, д. 36, стр. 3, оф. 203

**Телефон:** +7 351 725-75-64

**Факс:** +7 351 725-89-59

**E-mail:** sales@tpchel.ru

**Сайт:** www.tpchel.ru

**Сервисная служба:** +7 (351) 725-74-72, 725-75-10

**Продукция произведена ООО «ТЕПЛОПРИБОР-СЕНСОР»**

**2023**