



## Протокол обмена по интерфейсу RS-485 MODBUS для цифровых регуляторов массового расхода газа (РРГ-12 и др.)

### Руководство по программированию

#### Содержание

1. Общие положения .....	1
2. Форматы сообщений .....	1
2.1. Формат байта .....	2
2.2. Формат кадра .....	2
3. Форматы данных .....	2
3.1. Формат INT .....	2
3.2. Формат WORD .....	3
3.3. Формат BOOL .....	3
4. Генерация и проверка контрольной суммы .....	3
4.1. Формирование контрольной суммы алгоритмическим способом .....	4
4.2. Формирование контрольной суммы табличным способом .....	4
5. Описание системы команд .....	6
5.1. Функция 03h – чтение группы регистров .....	6
5.2. Функция 06h – установка регистра .....	6
5.3. Функция 0F – установка битов .....	7
5.4. Обработка ошибок .....	7
6. Адресное пространство .....	8
6.1. Регистры .....	8
6.2. Флаги состояния 1 .....	9
6.3. Флаги состояния 2 .....	10
7. Полезные ссылки .....	10

#### 1. Общие положения

Данный протокол служит для организации обмена данными между цифровыми регуляторами расхода РРГ12, связанными в сети по интерфейсу EIA/TIA-485 с ведущим устройством, работающим по протоколу MODBUS, например, персональным компьютером (PC), программируемым логическим контроллером (PLK), интеллектуальным сенсорным экраном (HMI).

В основу протокола обмена положен протокол MODBUS-RTU. Отличие заключается в поддержке ограниченного набора команд.

При построении сети используется принцип организации ведущий-ведомый (master-slave). В сети может присутствовать только один ведущий узел и несколько ведомых узлов. В качестве ведущего узла выступает PC либо PLK, в качестве ведомых узлов – регуляторы РРГ12. При данной организации инициатором циклов обмена может выступать исключительно ведущий узел.

Запросы ведущего узла - индивидуальные (адресуемые к конкретному прибору). Ведомые узлы осуществляют передачу, отвечая на индивидуальные запросы ведущего узла. При обнаружении ошибок в получении запросов, либо невозможности выполнения полученной команды, ведомый узел, в качестве ответа, генерирует сообщение об ошибке.

#### 2. Форматы сообщений

Протокол обмена имеет четко определенные форматы сообщений. Описывается формат байт и формат кадров. Соблюдение форматов обеспечивает правильность и устойчивость функционирования сети.

##### 2.1. Формат байта

Приборы настраиваются на работу с одним форматом байт данных: без контроля паритета. Количество стоп битов – один. Передача восьми бит данных производится младшими битами вперед.

Передача байт осуществляется на скоростях, кратных 1200 бит/с (9600, 19200, 38400). По умолчанию, при изготовлении, приборы настраиваются на работу со скоростью 19200 бит/с.

##### 2.2. Формат кадра

Длина кадра не может превышать 256 байт. Контроль начала и окончания кадра осуществляется при помощи интервалов молчания, длиной не менее времени передачи 25мс. Формат кадра приведен на рис. 2.3. Поле адреса занимает один байт.

интервал молчания > 3.5 байта	Адрес	Код функции	Данные	Контрольная сумма	интервал молчания > 3.5 байта
	1 байт	1 байт	до 252 байт	2 байта	

Рис. 2.3. – Формат кадра.

Кадр должен передаваться как непрерывный поток байт. Правильность принятия кадра дополнительно контролируется проверкой контрольной суммы.

На рис. 2.4 приведен пример последовательной передачи 3 кадров. Перед началом передачи и после передачи каждого кадра интервал молчания превышает 25мс.

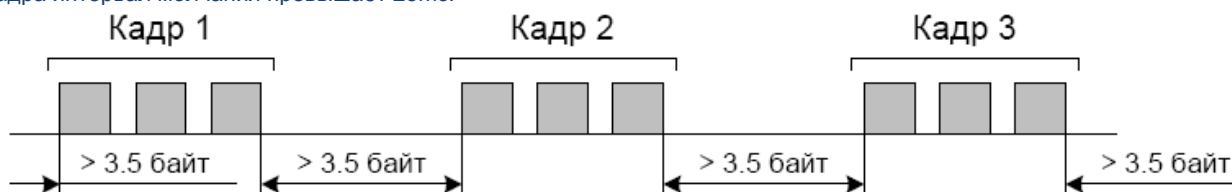


Рис. 2.4. – Пример передачи последовательности кадров

Интервал молчания задается в мс. Интервал молчания РРГ12 имеет значение 20 мс.

### 3. Форматы данных

Регуляторы имеют три формата программно-доступных регистров (таблица 3.1). Протокол обмена не имеет средств указания типа передаваемых данных. Тип конкретного регистра определяется исключительно описанием адресного пространства. Вся ответственность по толкованию типа ложится на программное обеспечение верхнего уровня.

Таблица 3.1 - Форматы программно-доступных регистров

Тип	Размерность	Диапазон	Примечание
INT	2 байта	-32,768.....+32,767	Знаковое целое
WORD	2 байта	0.....FFFFh	Беззнаковое целое
BOOL	1,2 байта	0.....FFFFh	Битовый массив – 16 бит

#### 3.1. Формат INT

Формат целое знаковое представлен на рис. 3.3. Данные передаются старшим байтом вперед.

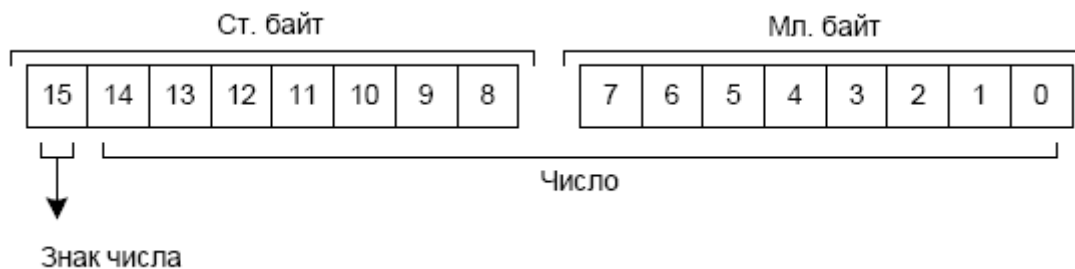


Рис. 3.3. - Формат INT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0
-	FC18h = -1000														

Рис. 3.4. - Пример кодирования числа -1000 в формате INT

Данный формат используется при чтении значения расхода.

#### 3.2. Формат WORD

Формат целое беззнаковое представлен на рис. 3.5. Данные передаются старшим байтом вперед. На рис. 3.6. приведен пример кодирования числа 1000 в формате WORD.

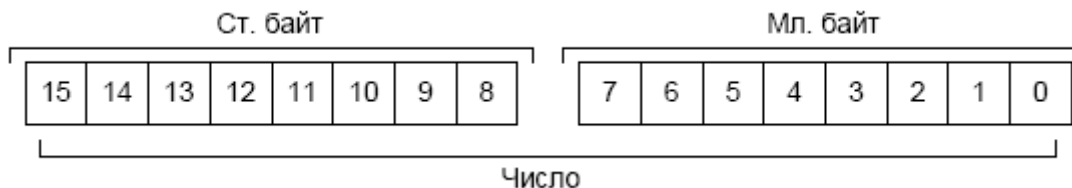


Рис. 3.5. - Формат WORD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0
03E8h = 1000															

**Рис. 3.6. - Пример кодирования числа 1000 в формате WORD**

Данный формат используется при задании расхода.

#### 3.4. Формат BOOL

Формат BOOL предназначен для кодирования дискретных величин. Один регистр типа BOOL содержит состояние шестнадцати логических переменных. Адресация переменных начинается с младших разрядов. Значение 1 соответствует TRUE (истина), 0 – FALSE (ложь).

#### 4. Генерация и проверка контрольной суммы

Контрольная сумма (CRC16) представляет собой циклический проверочный код на основе неприводимого полинома A001h. Передающее устройство формирует контрольную сумму для всех байт передаваемого сообщения. Принимающее устройство аналогичным образом формирует контрольную сумму для всех байт принятого сообщения и сравнивает ее с контрольной суммой, принятой от передающего устройства. При несовпадении сформированной и принятой контрольных сумм генерируется сообщение об ошибке.

Поле контрольной суммы занимает два байта. Контрольная сумма в сообщении передается младшим байтом вперед. Можно выделить два способа формирования контрольной суммы: алгоритмический и табличный.

##### 4.1. Формирование контрольной суммы алгоритмическим способом

Алгоритмический способ формирования контрольной суммы прост в понимании, но имеет более низкое быстродействие по сравнению с табличным. Контрольная сумма формируется по следующему алгоритму:

1. загрузка CRC регистра (16 бит) единицами (FFFFh);
2. исключающее ИЛИ с первыми 8 битами байта сообщения и содержимым CRC регистра;
3. сдвиг результата на один бит вправо;
4. если сдвигаемый бит = 1, исключающее ИЛИ содержимого регистра со значением A001h;
5. если сдвигаемый бит = 0, повторить шаг 3;
6. повторять шаги 3, 4, 5 пока не будут выполнены 8 сдвигов;
7. исключающее ИЛИ со следующими 8 битами байта сообщения и содержимым CRC регистра;
8. повторять шаги 3 – 7 пока все байты сообщения не будут обработаны;
9. конечное содержимое регистра будет содержать контрольную сумму.

Подпрограмма алгоритмического формирования контрольной суммы на языке Паскаль:

```

type
TsendRecvBuf : array [0..255] of byte;
const
Polynom = $A001;
function GenerateCRC(Buf:TsendRecvBuf;Count:word):word;
var
i : word;
CRC : word;
bitCounter : byte;
begin
CRC:=$FFFF;
for i:=0 to Count - 3 do
begin
CRC:=CRC xor Buf[i];
for bitCounter:=0 to 7 do
begin
if (CRC and $0001) = 0 then
CRC:=CRC shr 1
else
begin
CRC:=CRC shr 1;
CRC:=CRC xor Polynom;
end;
end;
end;
result:=CRC;
end;

```

##### 4.2. Формирование контрольной суммы табличным способом.

Подпрограмма табличного формирования контрольной суммы на языке ассемблера микроконтроллера 8051:

```

Table1: ; 00/08 01/09 02/0A 03/0B 04/0C 05/0D 06/0E 07/0F
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;00
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;01

```

```

db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;02
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;03
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;04
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;05
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;06
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;07
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;08
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;09
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;0A
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;0B
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;0C
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;0D
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h ;0E
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h
db 000h, 0C1h, 081h, 040h, 001h, 0C0h, 080h, 041h ;0F
db 001h, 0C0h, 080h, 041h, 000h, 0C1h, 081h, 040h
Table2: ; 00/08 01/09 02/0A 03/0B 04/0C 05/0D 06/0E 07/0F
db 000h, 0C0h, 0C1h, 001h, 0C3h, 003h, 002h, 0C2h ;00
db 0C6h, 006h, 007h, 0C7h, 005h, 0C5h, 0C4h, 004h
db 0CCh, 00Ch, 00Dh, 0CDh, 00Fh, 0CFh, 0CEh, 00Eh ;01
db 00Ah, 0CAh, 0CBh, 00Bh, 0C9h, 009h, 008h, 0C8h
db 0D8h, 018h, 019h, 0D9h, 01Bh, 0DBh, 0DAh, 01Ah ;02
db 01Eh, 0DEh, 0DFh, 01Fh, 0DDh, 01Dh, 01Ch, 0DCh
db 014h, 0D4h, 0D5h, 015h, 0D7h, 017h, 016h, 0D6h ;03
db 0D2h, 012h, 013h, 0D3h, 011h, 0D1h, 0D0h, 010h
db 0F0h, 030h, 031h, 0F1h, 033h, 0F3h, 0F2h, 032h ;04
db 036h, 0F6h, 0F7h, 037h, 0F5h, 035h, 034h, 0F4h
db 03Ch, 0FCh, 0FDh, 03Dh, 0FFh, 03Fh, 03Eh, 0FEh ;05
db 0FAh, 03Ah, 03Bh, 0FBh, 039h, 0F9h, 0F8h, 038h
db 028h, 0E8h, 0E9h, 029h, 0EBh, 02Bh, 02Ah, 0EAh ;06
db 0EEh, 02Eh, 02Fh, 0EFh, 02Dh, 0EDh, 0ECh, 02Ch
db 0E4h, 024h, 025h, 0E5h, 027h, 0E7h, 0E6h, 026h ;07
db 022h, 0E2h, 0E3h, 023h, 0E1h, 021h, 020h, 0E0h
db 0A0h, 060h, 061h, 0A1h, 063h, 0A3h, 0A2h, 062h ;08
db 066h, 0A6h, 0A7h, 067h, 0A5h, 065h, 064h, 0A4h
db 06Ch, 0ACh, 0ADh, 06Dh, 0AFh, 06Fh, 06Eh, 0AEh ;09
db 0AAh, 06Ah, 06Bh, 0ABh, 069h, 0A9h, 0A8h, 068h
db 078h, 0B8h, 0B9h, 079h, 0BBh, 07Bh, 07Ah, 0BAh ;0A
db 0BEh, 07Eh, 07Fh, 0BFh, 07Dh, 0BDh, 0BCh, 07Ch
db 0B4h, 074h, 075h, 0B5h, 077h, 0B7h, 0B6h, 076h ;0B
db 072h, 0B2h, 0B3h, 073h, 0B1h, 071h, 070h, 0B0h
db 050h, 090h, 091h, 051h, 093h, 053h, 052h, 092h ;0C
db 096h, 056h, 057h, 097h, 055h, 095h, 094h, 054h
db 09Ch, 05Ch, 05Dh, 09Dh, 05Fh, 09Fh, 09Eh, 05Eh ;0D
db 05Ah, 09Ah, 09Bh, 05Bh, 099h, 059h, 058h, 098h
db 088h, 048h, 049h, 089h, 04Bh, 08Bh, 08Ah, 04Ah ;0E
db 04Eh, 08Eh, 08Fh, 04Fh, 08Dh, 04Dh, 04Ch, 08Ch
db 044h, 084h, 085h, 045h, 087h, 047h, 046h, 086h ;0F
db 082h, 042h, 043h, 083h, 041h, 081h, 080h, 040h
;При входе R2 – количество байт в сообщении,
; Addr_buff – начальный адрес буфера сообщения
;При выходе CRC16 – младший байт контрольной суммы (LB),
; CRC16+1 - старший байт контрольной суммы (HB).
GenerateCRC:
mov R0, #Addr_buff ;
mov CRC16, #0FFh ;
mov CRC16+1, #0FFh ;
$11: mov A, @R0 ;
xrl A, CRC16 ;
mov B, A ;
mov DPTR, #Table1 ;
movc A, @A+DPTR ;

```

```

mov CRC16,CRC16+1 ;
xrl CRC16,A ;
mov A,B ;
mov DPTR,#Table2 ;
movc A,@A+DPTR ;
mov CRC16+1,A ;
inc R0 ;
djnz R2,$11 ;
ret

```

## 5. Описание системы команд

### 5.1. Функция 03h – чтение группы регистров

Функция 03h обеспечивает чтение содержимого регистров ведомого устройства.

Широковещательная передача поддерживается. В запросе ведущего содержится адрес начального регистра, а также количество слов для чтения. Адресация регистров начинается с нуля.

Различия между запросом чтения регистров различных типов не существует. К примеру, запрос на чтение регистра типа INT совпадает с запросом на чтение одного регистра типа WORD.

Ответ ведомого содержит количество возвращаемых байт и запрошенные данные.

Пример запроса и ответа приведён на рис. 5.1.

Запрос							
Адрес	Функция	Нач.адрес НВ	Нач.адрес ЛВ	Кол.слов НВ	Кол.слов ЛВ	CRC LB	CRC НВ
01h	03h	00h	02h	00h	02h	65h	CBh

Ответ – значение регистра 00h =1, значение регистра 01h =1;

Адрес	Функция	Кол.байт	Данные НВ НВ	Данные НВ ЛВ	Данные ЛВ НВ	Данные ЛВ ЛВ	CRC LB	CRC НВ
01h	03h	04h	00h	91h	00h	51h	6Ah	22h

**Рис. 5.1. - Пример запроса и ответа функции 03h – чтение группы регистров**

### 5.2. Функция 06h – установка регистра

Функция 06h обеспечивает запись в регистр ведомого устройства. Широковещательная передача не поддерживается.

В запросе ведущего содержится адрес регистра и данные для записи. Адресация регистров начинается с нуля. Ответ ведомого совпадает с запросом ведущего и содержит адрес регистра и установленные данные. Пример запроса и ответа приведён на рис. 5.2.

Запрос:							
Адрес	Функция	Адрес регистра НВ	Адрес регистра ЛВ	Данные НВ	Данные ЛВ	CRC LB	CRC НВ
01h	06h	00h	04h	00h	02h	49h	CAh

Ответ:

Адрес	Функция	Адрес регистра НВ	Адрес регистра ЛВ	Данные НВ	Данные ЛВ	CRC LB	CRC НВ
01h	06h	00h	04h	00h	02h	49h	CAh

**Рис. 5.2. - Пример запроса и ответа функции 06h – установка регистра**

Функция записи имеет аппаратные и программные ограничения, описанные в разделе 6 “Адресное пространство”.

### 5.3. Функция 0Fh – установка группы флагов

Команда с номером 15 используется для установки/снятия одного или нескольких флагов. В байте 0 передается сетевой адрес. В байте 3 передается начальный адрес группы флагов (0-6), в байте 5 передается число флагов, в байте 7 передаются задания флагов. Сумма адреса и числа флагов не должна быть больше максимального адреса флага (<8). Широковещательная передача не поддерживается.

Биты числа задания должны соответствовать значениям флагов. Так значение первого бита соответствует состоянию флага с первым адресом, заданным в байте 3. Значение второго бита соответствует состоянию флага со вторым адресом, равным значению бита 3 +1, и т.д. Первый бит – наименее значимый бит. Пример запроса и ответа приведён на рис. 5.3.

Запрос									
Адрес	Функция	Нач.адрес НВ	Нач.адрес LB	Кол-во флагов НВ	Кол-во флагов LB	Кол-во байт	Данные	CRC LB	CRC НВ
01h	0Fh	00	00	00	02	1	01	1Fh	57h

Ответ:

Адрес	Функция	Нач.адрес НВ	Нач.адрес LB	Кол-во флагов НВ	Кол-во флагов LB	CRC LB	CRC НВ
01h	0Fh	00	00	00	02	D4h	0Ah

**Рис. 5.3. - Пример запроса и ответа функции 0Fh – установка группы флагов**

В результате выполнения этой команды установили бит по адресу 0 (установили режим регулирования) и сбросили бит по адресу 1 (выбрали аналоговый вход).

Команда 15 дублирует команду 3 для регистра флагов состояния 1 и позволяет управлять только определенными флагами.

#### 5.4 Обработка ошибок

В случае возникновения ошибочной ситуации при принятии кадра (ошибка сетевого адреса, ошибка контрольной суммы) ведомое устройство ответ не возвращает.

В случае возникновения ошибки в формате или значении передаваемых данных (неподдерживаемый код функции и т. д.) ведомое устройство должно принять кадр запроса и сформировать ответ с признаком и кодом ошибки. Признаком ошибки является установленный в единицу старший бит в поле функции. Под код ошибки отводится отдельное поле в ответе.

Пример ответа приведен на рис. 5.7. Коды ошибок приведены в таблице 5.1.

Запрос – функция 30h не поддерживается

Адрес	Функция	Данные	CRC LB	CRC НВ
01h	30h		XXh	XXh

Ответ:

Адрес	Функция	Код ошибки	CRC LB	CRC НВ
01h	B0h	01h	94h	00h

**Рис. 5.7. - Пример ответа после возникновения ошибки**

**Таблица 5.1. - Коды ошибок**

Код ошибки	Название	Описание
01h	ILLEGAL FUNCTION	Принятый код функции не может быть обработан
02h	ILLEGAL DATA ADDRESS	Адрес регистра(флага), указанный в запросе не доступен
03h	ILLEGAL DATA VALUE	Величина содержащаяся в поле данных запроса является не допустимой величиной
04h	ILLEGAL NUMBER OF VALUE	Ошибка в количестве регистров(флагов)

## 6. Адресное пространство

### 6.1 Регистры

Все регистры адресного пространства устройства доступны для чтения. Для некоторых регистров запрещена операция записи.

Команда	Адрес регистра	Содержимое
3, 6	0	Сетевой адрес
3	1	№ РРГ
3, 6	2	Флаги состояния 1
3	3	Флаги состояния 2
3,6	4	Задание расхода газа
3	5	Выход. Расход газа
3,6	6	Скорость работы СОМ-порта

**Примечания**

- 1) В слове расхода газа, прочитанном из РРГ старший бит – передает знак результата, 0-плюс, 1-минус. Другими словами формат этого слова – int. Формат слова задания расхода – WORD(unsigned int).
- 2) Определить величину расхода газа(величину давления) в %(P%) можно по формуле  $P\% = P/100.0 + (-)\text{Допуск}\%$ , где 'P' – десятичное число, полученное в слове расхода, 'Допуск%' - неточность определения расхода в %, заданная в паспорте. Диапазон изменений показаний величины расхода(давления): -0,5% ... + 130%.
- 3) Определить величину задания расхода газа (давления) в %(Задание%) можно по формуле:  $\text{Задание}\% = \text{Задание}/100.0$ , где 'Задание' – десятичное число, передаваемое/полученное в слове задания. Примечание. Измерение и задание величины давления и задания давления производится аналогично.
- 4) Изменение скорости обмена последовательного порта. В слово записывают значения -0,1,2 - соответствующие скоростям: 0- 9600 bod, 1- 38400 bod, 2-FFh -19200 bod. РРГ предустановлен на скорость обмена 19200 bod.
- 5) Сетевой адрес устанавливается/читается только из младшего байта. Диапазон адресов 1-255.

**6.2 Флаги состояния 1**

Значения битов регистра флагов состояния 1 следующие.

Команда	Бит	Состояние		Значение
3,6,15	0	0		Режим измерения
		1		Режим регулирования
3,6,15	1	0		Аналоговый вход
		1		Цифровой вход
3,6,15	2, 3	Бит 3	Бит 2	
		0	1	Затвор открыт
		1	0	Затвор закрыт
		0	0	Затвор в режиме регулирования
3,6,15	4	0		Восстановл. аналогов. вход, регулир, скорость 19200, РРГ
		1		Восстановл. вход ц/а, регул./изм., скорость UART, РДГ/РРГ
3,6,15	5	0		Режим РРГ
		1		Режим РДГ
3,6,15	6	1		Установка нуля
	7	x		-

**Примечания**

- 1) Флаги состояния 1 могут считываться командой 3 и устанавливаться командами 6 и 15.
- 2) Бит 1 также устанавливается в 1 автоматически при вводе задания с компьютера
- 3) Команда на установку нуля (бит 6 = 1) подается через 5-10 мин после прекращения подачи газа на регулятор при закрытом затворе и не ранее, чем через 20-40 мин после подачи питания на РРГ. Бит 6 сбрасывается в нуль автоматически после завершения в микропроцессоре процедуры установки нуля.
- 4) Установка способа возврата в рабочее состояние при включении и сбоях в питании – бит 4. Значения бита определяют способ перехода в рабочий режим при включении или сбоях питания:  
0 – первый способ,  
1-FFh - второй способ.  
В первом способе при каждом включении питания РРГ работает в режиме регулирования с аналоговым вводом задания расхода. Скорость UART -19200 bod. При сбое питания, РРГ входит в этот же режим.  
Во втором способе, РРГ сохраняет установленный ранее режим работы (задание цифровое/аналоговое, скорость СОМ-порта, регулятор расхода/давления, регулирование/измерение, затвор (вкл./выкл./регул.) ). Если сохраняется цифровой ввод, то восстанавливается и установленное ранее значение расхода.  
РРГ поставляется предустановленным в первый способ работы.
- 5) Внешнее управление затвором и режимами РРГ/РДГ имеет приоритет над программным.  
Внимание. После перевода внешнего управления затвором в нейтральное положение ( 0V на контакте 8 разъема DRB-9M), начинают работать программные установки затвора -открыт/закрыт/регулирование - введенные в любое время.
- 6) При установленном бите 3 или 4 флагов состояния 2 изменение состояния бита 5 флагов состояния 1 не влияет на работу регулятора.

**6.3. Флаги состояния 2.**

Значения битов регистра флагов состояния 2 следующие.

Команда	Бит	Состояние		Значение
3	0	0		Норма
		1		Недостаточно газа более 20 сек.

**СИСТЕМЫ ТЕХНОЛОГИЧЕСКОГО ГАЗОСНАБЖЕНИЯ: ПРОЕКТИРОВАНИЕ, ПРОИЗВОДСТВО, МОНТАЖ**



3	1	0	Затвор в режиме регулирования	
		1	Затвор в положениях вкл. или выкл.	
3	2	0	Выбор РРГ	
		1	Выбор РДГ	
3	3	0	-	
		1	Доступен только РДГ.	
3	4	0	-	
		1	Доступен только РРГ.	
3	5	Бит 5	Бит 6	Состояние внешнего управления затвором
		1	0	Затвор открыт
		0	1	Затвор закрыт
		0	0	Режим регулирования
	7	x		-

**Примечание.** Флаги состояния 2 могут только считываться командой 3.

#### 7. Полезные ссылки

Дополнительную информацию можно найти по следующим адресам:

1. <http://www.eltochpribor.ru> - сайт ООО «ЭЛТОЧПРИБОР»
2. <http://www.modbus.org/> - сайт, посвященный стандарту MODBUS;
3. <http://www.rs485.com/> - сайт, посвященный последовательным интерфейсам EIA/TIA-232, EIA/TIA-422, EIA/TIA-485;