

# Описание протокола Modbus

v.1.5

Системы газоаналитические  
многофункциональные

**СГМ ЭРИС-130**  
**СГМ ЭРИС-110 МАП**



**Оглавление**

Оглавление.....	2
1 Параметры канального уровня .....	3
2 О протоколе.....	3
3 Перечень поддерживаемых функций .....	4
3.1 Функция 03 (03h). Чтение группы регистров .....	4
3.2 Функция 06 (06h). Запись одного регистра.....	5
3.3 Функция 08 (08h). Эхо .....	6
3.4 Функция 16 (10h). Запись группы регистров.....	6
3.5 Функция 20 (14h). Чтение комментария устройства .....	7
3.6 Функция 21 (15h). Запись комментария устройства .....	8
3.7 Функция 43 (2bh). Чтение идентификатор устройства .....	8
3.8 Функция 82 (52h). вернуть длину файла .....	11
3.9 Функция 85 (55h). Открыть файл .....	11
3.10 Функция 87 (57h). Закрыть файл.....	12
3.11 Функция 90 (5Ah). Считать порцию файла .....	13
3.12 Функция 91(5Bh). Удалить файл .....	14
3.13 Функция 92 (5Ch). Рестарт удаленного контроллера .....	15
3.14 Функция 110 (6Eh). получить текущее время контроллера.....	15
3.15 Функция 111 (6Fh). Установить текущее время контроллера .....	16
ПРИЛОЖЕНИЕ А Адресное пространство регистров .....	17
ПРИЛОЖЕНИЕ Б Структура файлов архивов .....	26

## 1 Параметры канального уровня

Число стартовых бит (лог.0)	1 бит
Длина информационной посылки	8 бит
Число стоповых бит (лог.1)	1 бит
Контрольный разряд	Четность
Скорость обмена	1200/2400/4800/9600/19200/ 38400/57600/115200 бит/с

## 2 О протоколе

В широковещательном режиме используется адрес 0. Все SL интерпретируют такое сообщение как выполнение определенного действия, но без посылки подтверждения. Так как сеть будет состоять из одного контроллера, то его адрес фиксированный – 2.

Структура цикла запрос/ответ:

Запрос от главного	Размер поля	Ответ подчиненного	Размер поля
Адрес устройства	1 байт	Адрес устройства	1 байт
Код функции	1 байт	Код функции	1 байт
8 - битные байты данных	N байт	8 - битные байты данных	N байт
Контрольная сумма CRC	2 байта	Контрольная сумма CRC	2 байта

Запрос: Код функции в запросе говорит подчиненному устройству, какое действие необходимо провести. Байты данных содержат информацию необходимую для выполнения запрошенной функции. Например, код функции 4 подразумевает запрос на чтение содержимого регистров подчиненного.

Ответ: Если подчиненный дает нормальный ответ, код функции в ответе повторяет код функции в запросе. В байтах данных содержится затребованная информация. Если имеет место ошибка, то код функции модифицируется, и в байтах данных передается причина ошибки.

Запись данных производится от старшего байта к младшему.

Контрольная сумма записывается – сначала младший байт затем старший.

Пример расчёта контрольной суммы:

```
void CalcCRC(unsigned char *DataCRC, unsigned int *len)
{
    unsigned int CRCrtu = 0xFFFF;
    for(unsigned int q=0; q<*len; q++)
    {
        CRCrtu = CRCrtu^DataCRC[q];
        for(char i=0; i<8; i++)
        {
            if((CRCrtu&0x0001)==1) {CRCrtu >>= 1; CRCrtu ^=
0xA001;}
            else {CRCrtu >>= 1;}
        }
    }
    DataCRC[*len] = LOBYTE(CRCrtu);
    DataCRC[*len+1] = HIBYTE(CRCrtu);
    *len+=2;
}
```

### 3 Перечень поддерживаемых функций

Список функций, поддерживаемых МК:

№	Код	HEX	Функция
1	03	03h	Чтение группы регистров
2	06	06h	Запись одного регистра
3	08	08h	Эхо
4	16	10h	Запись группы регистров
5	20	14h	Чтение комментария устройства
6	21	15h	Запись комментария устройства
7	43	2Bh	Чтение идентификатора устройства
8	82	52h	Вернуть длину файла
9	85	55h	Открыть существующий файл
10	87	57h	Закрыть файл
11	90	5Ah	Считать порцию файла
12	91	5Bh	Удалить файл
13	92	5Ch	Рестарт удаленного контроллера
14	110	6Eh	Получить текущее время контроллера
15	111	6Fh	Установить текущее время контроллера

Адресное пространство регистров для команд 0x03, 0x06 и 0x10 приведено в приложении А.

#### 3.1 Функция 03 (03h). Чтение группы регистров

Функция с кодом 03h используется для чтения непрерывного массива (группы регистров) переменных программы из удаленного контроллера. В поле данных запроса задается адрес первой переменной и их количество.

Содержимое переменных в ответном сообщении упаковывается – одна переменная в два байта, первый байт содержит старшие значащие разряды переменной, а второй байт – младшие разряды.

Запрос

Код функции	1 Байт	0x03
Начальный адрес	2 Байта	от 0x0000 до 0xFFFF
Количество переменных	2 Байта	от 1 до N

N – Количество переменных в непрерывном адресном пространстве.

Ответ

Код функции	1 Байт	0x03
Счетчик байт	1 Байт	2*N
Значения входов/выходов	2*N Байт	

N – Количество переменных, указанное в запросе.

Ошибка

Код функции	1 Байт	0x03+0x80
Код ошибки	1 Байт	02, 03, 04, 06

### 3.2 Функция 06 (06h). Запись одного регистра

Пример запроса для чтения двух переменных с адресами 108-109:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	03	Функция	03
Начальный адрес Hi	00	Счетчик байт	04
Начальный адрес Li	6C	Величина переменной Hi (108)	02
Количество переменных Hi	00	Величина переменной Lo (108)	2B
Количество переменных Li	02	Величина переменной Hi (109)	00
		Величина переменной Lo (109)	00

Содержимое переменной 108 в HEX формате 02 2B в десятичном представлении равно 555. Содержимое переменной 109 в десятичном представлении равно 0.

### 3.2 Функция 06 (06h). Запись одного регистра

Функция с этим кодом используется для изменения значения одной переменной (регистра) программы.

В поле данных запроса задается адрес переменной, которую требуется изменить.

Обычный ответ – это эхо запроса, возвращаемое после изменения переменной.

Запрос

Код функции	1 Байт	0x06
Адрес переменной	2 Байта	от 0x0010 до 0xFFFF
Значение переменной	2 Байта	от 0x0010 до 0xFFFF

Ответ

Код функции	1 Байт	0x06
Адрес переменной	2 Байта	от 0x0010 до 0xFFFF
Значение переменной	2 Байта	от 0x0010 до 0xFFFF

Ошибка

Код функции	1 Байт	0x06+0x80
Код ошибки	1 Байт	02, 03, 04, 06

Пример запроса для записи HEX значения 00 03 в регистр 2:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	06	Функция	06
Адрес переменной Hi	00	Адрес переменной Hi	00
Адрес переменной Li	02	Адрес переменной Li	02
Значение переменной Hi	00	Значение переменной Hi	00
Значение переменной Li	03	Значение переменной Li	03

### 3.3 Функция 08 (08h). Эхо

Функция с этим кодом используется для запроса эха от удаленного контроллера. Программа в контроллере не анализирует данные этой функции, кодов ошибок не возвращает. Ответ на эту функцию всегда повторяет запрос.

Запрос

Код функции	1 Байт	0x08
Данные	2 Байта	от 0x0010 до 0xFFFF

Ответ

Код функции	1 Байт	0x08
Данные	2 Байта	от 0x0010 до 0xFFFF

### 3.4 Функция 16 (10h). Запись группы регистров

Функция с этим кодом используется для изменения значений нескольких переменных программы в МК.

Новые значения для управляющих переменных программы задаются в поле данных запроса. Значения упаковываются по два байта для одной переменной.

В поле данных запроса задается начальный адрес изменяемых переменных. Обычный ответ содержит код функции, начальный адрес и количество измененных регистров.

Запрос

Код функции	1 Байт	0x10
Начальный адрес	2 Байта	от 0x0000 до 0xFFFF
Количество переменных	2 Байта	от 0x0000 до 0x0078
Количество байт	1 Байт	2*N
Значения для переменных	2*N Байта	Значения

Ответ

Код функции	1 Байт	0x0F
Начальный адрес	2 Байта	от 0x0000 до 0xFFFF
Количество переменных	2 Байта	2*N

Ошибка

Код функции	1 Байт	0x10+0x80
Код ошибки	1 Байт	02, 03, 04, 06

### 3.5 Функция 20 (14h). Чтение комментария устройства

Пример запроса записи значений 00 0A и 01 02 в две переменные, начиная со второй 2:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	10	Функция	10
Начальный адрес Hi	00	Начальный адрес Hi	00
Начальный адрес Li	02	Начальный адрес Li	02
Количество переменных Hi	00	Количество переменных Hi	00
Количество переменных Li	02	Количество переменных Li	02
Счетчик байт	04		
Значение переменной Hi	00		
Значение переменной Li	0A		
Значение переменной Hi	01		
Значение переменной Li	02		

### 3.5 Функция 20 (14h). Чтение комментария устройства

Функция с этим кодом используется для чтения комментария (расположения).

Информация об устройстве возвращается в виде закодированной строки.

Запрос

Код функции	1 Байт	0x14
Канал	1 Байт	0...255

Ответ

Код функции	1 Байт	0x14
Канал	1 Байт	0...255
Длина строки (N)	1 Байт	от 0 до 16
Строка	N Байт	N

### 3.6 Функция 21 (15h). Запись комментария устройства

Функция с этим кодом используется для сохранения комментария (расположения).

Информация об устройстве сохраняется в виде закодированной строки.

Запрос

Код функции	1 Байт	0x15
Канал	1 Байт	0...255
Длина строки (N)	1 Байт	от 0 до 16
Строка	N Байт	

Ответ

Код функции	1 Байт	0x15
Канал	1 Байт	0...255
Длина строки (N)	1 Байт	от 0 до 16

### 3.7 Функция 43 (2bh). Чтение идентификатор устройства

Функция с этим кодом используется для получения информации об удаленном устройстве и программном обеспечении, там установленном.

Информация об устройстве возвращается в виде строк ASCII символов. В одном ответе может содержаться несколько строк (поточный запрос) или одна строка (индивидуальный запрос). Каждая символьная строка имеет свой идентификационный код (Код Строки). Все строки разбиты на две категории: основная информация и вспомогательная информация. Каждая категория имеет код (Код Категории). Основная информация состоит из трех символьных строк, вспомогательная информация состоит из четырех символьных строк.

Коды символьных строк и их наименования приведены в следующей таблице:

Строка ИД	Название	Значение	Категория информации
0x00	Производитель	ООО Эрис	Основная
0x01	Код продукта	110.00	
0x02	Версия	v.1.00	
0x03	Код производителя	5791	Вспомогательная
0x04	Название продукта	СГМ Эрис	
0x05	Название модели	110	
0x06	Название приложения	Газоанализатор	



### 3.7 Функция 43 (2bh). Чтение идентификатор устройства

Коды категорий и их наименования приведены в таблице:

Категория ИД	Название
0x01	Основная
0x02	Вспомогательная
0x04	Индивидуальная

Запрос:

Код функции	1 Байт	0x2B
Тип интерфейса	1 Байт	0x0E
Код Категории	1 Байт	01/02/04
Код Строки	1 Байт	0x00-0x06

Ответ:

Код функции	1 Байт	0x2B
Тип интерфейса	1 Байт	0x0E
Код Категории	1 Байт	01/02/04
Ответный код категории	1 Байт	01/02/81/82
Признак продолжения	1 Байт	0x00
Код Следующей Строки	1 Байт	0x00
Количество строк	1 Байт	N
Код Строки 1	1 Байт	0x00-0x06
Количество байт в строке 1	1 Байт	N1
Строка 1	N1 Бай	Строка ASCII символов
Код Строки 2	1 Байт	0x00-0x06
Количество байт в строке 2	1 Байт	N2
Строка 2	N2 Бай	Строка ASCII символов

Ошибка:

Код функции	1 Байт	0x2B
Тип интерфейса	1 Байт	0x0E
Код Ошибки	1 Байт	02 или 03

Пример: Чтение строк основной категории:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	2B	Код функции	2B
Тип интерфейса	0E	Тип интерфейса	0E
Код Категории	01	Код Категории	01

### 3.7 Функция 43 (2bh). Чтение идентификатор устройства

Код Строки	00	Ответный код категории	01
		Признак продолжения	00
		Код Следующей Строки	00
		Количество строк	03
		Код Строки 1	00
		Количество байт в строке 1	0D
		Строка 1	“Эрис”
		Код Строки 2	01
		Количество байт в строке 2	0C
		Строка 2	“XXXXXX-XXXXXX”
		Код Строки 3	02
		Количество байт в строке 3	1B
		Строка 3	“v.4.02 Nov 21 2004,09:24:01”

Ответ состоит из трех строк, каждая строка начинается с кода и количества байт в строке.

Пример: Чтение только версии:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	2B	Код функции	2B
Тип интерфейса	0E	Тип интерфейса	0E
Код Категории	04	Код Категории	04
Код Строки	02	Ответный код категории	84
		Признак продолжения	00
		Код Следующей Строки	00
		Количество строк	01
		Код Строки	02
		Количество байт в строке	1B
		Строка	“V4.02 Nov 21 2004,09:24:01”

Ответ состоит только из одной строки символов: версии и даты сборки программы. Параметр «Ответный код категории» равен 0x81 – старший бит, равный единице, говорит об индивидуальном запросе.

### 3.8 Функция 82 (52h). вернуть длину файла

Функция с этим кодом возвращает длину файла. Если запрошенный файл отсутствует, возвращаем 0xFFFFFFFF.

Запрос

Код функции	1 Байт	0x52
Количество байт	1 Байт	от 0x00 до 0xFA
Имя файла	N Байт	ASCII коды
Нулевой байт	1 Байт	0

Ответ

Код функции	1 Байт	0x52
Длина файла	1 Байт	B1
Длина файла	1 Байт	B2
Длина файла	1 Байт	B3
Длина файла	1 Байт	B4

Пример запроса длины файла с именем «1.dat», который состоит из 12800 байт:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	52	Функция	52
Количество байт	05	Длина файла	00
Имя файла байт 1	31	Длина файла	32
Имя файла байт 2	2E	Длина файла	00
Имя файла байт 3	64	Длина файла	00
Имя файла байт 4	61		
Имя файла байт 5	74		
Нулевой байт	00		

### 3.9 Функция 85 (55h). Открыть файл

Функция с этим кодом предназначена для открытия файла архива, например: «ch1.arh». При этом, новые записи архива будут создаваться в файле, например: «ch1.tmp». При закрытии файла, файл «ch1.tmp» автоматически будет переименован в файл «ch1.arh». Если во время чтения файла контроллер будет перезагружен или не будет в течении 5 секунд обращений к файлу архива, то файл автоматически будет закрыт (без удаления), а файл «ch1.tmp» будет добавлен в «ch1.arh».

Одновременно может быть открыт только один файл.

Файлы с расширением «\*.tmp» в файле каталога отсутствуют.

Порядок открытия файла:

1) Если есть открытый файл, то закрываем его.

### 3.10 Функция 87 (57h). Закрывать файл

- 2) Если открывается файл архива, то создаём файл «\*.tmp».
- 3) Если файл присутствует, то возвращаем длину файла.
- 4) Если запрошенный файл отсутствует, возвращаем 0xFFFFFFFF.

Запрос

Код функции	1 Байт	0x55
Количество байт	1 Байт	от 0x00 до 0xFA
Имя файла	N Байт	ASCII коды
Нулевой байт	1 Байт	0

Ответ

Код функции	1 Байт	0x55
Длина файла	1 Байт	B1
Длина файла	1 Байт	B2
Длина файла	1 Байт	B3
Длина файла	1 Байт	B4

Пример запроса на открытие файла с именем «1.dat», который состоит из 12800 байт:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	55	Функция	55
Количество байт	05	Длина файла	00
Имя файла байт 1	31	Длина файла	32
Имя файла байт 2	2E	Длина файла	00
Имя файла байт 3	64	Длина файла	00
Имя файла байт 4	61		
Имя файла байт 5	74		
Нулевой байт	00		

### 3.10 Функция 87 (57h). Закрывать файл

Функция с этим кодом предназначена для закрытия файла. В поле данных запроса с этим кодом функции находится только код функции.

Если был открыт файл архива, например: «ch1.arh», то в этот файл автоматически добавляются записи из файла «ch1.tmp».

Запрос

Код функции	1 Байт	0x57
-------------	--------	------

Ответ

Код функции	1 Байт	0x57
-------------	--------	------

### 3.11 Функция 90 (5Ah). Считать порцию файла

Функция с этим кодом предназначена для получения порции данных из файла на удаленном контроллере. Поле данных запроса содержит код функции и номер порции. Под номер порции выделяется 2 байта.

Ответ на запрос возвращает код функции, номер порции, размер порции в байтах и порцию данных из файла. Размер порции упакован в два байта, начиная с младшего байта. Наибольший размер возвращаемой порции данных – 512 байт.

#### Запрос

Код функции	1 Байт	0x5A
Номер порции	2 Байта	от 0x00 до 0xFF

#### Ответ

Код функции	1 Байт	0x5A
Номер порции	2 Байта	от 0x0000 до 0xFFFF
Размер порции	2 Байта	от 0x0000 до 0x0200
Порция данных	N Байт	

Пример запроса на чтение второй порции из 5 байт данных:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	5A	Функция	5A
Номер порции Lo	02	Номер порции Lo	02
Номер порции Hi	00	Номер порции Hi	00
		Размер порции Lo	05
		Размер порции Hi	00
		Байт 1	01
		Байт 2	02
		Байт 3	03
		Байт 4	04
		Байт 5	05

По данному запросу в файл на удаленном контроллере будет добавлено пять байт данных: 01 02 03 04 05.

### 3.12 Функция 91(5Bh). Удалить файл

Функция с этим кодом предназначена для удаления существующего файла с диска. В поле данных запроса указывается имя файла.

Если в данный момент открыт какой-либо файл, то этот файл закрывается и удаляется.

После успешного удаления в ответе возвращается код функции 91(5Bh). Если файла нет, возвращается -2, если произошла не предвиденная ошибка во время удаления, то возвращается -1. Формат ответа после ошибки имеет следующий вид:

Код функции	1 Байт	0x5B
Код ошибки	1 Байт	-1 или -2

#### Запрос

Код функции	1 Байт	0x5B
Количество байт	1 Байт	от 0x00 до 0xFA
Имя файла	N Байт	ASCII коды
Нулевой байт	1 Байт	0

#### Ответ

Код функции	1 Байт	0x5B
-------------	--------	------

Пример запроса на удаление файла с именем «1.dat»:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	5B	Функция	5B
Количество байт	05		
Имя файла байт 1	31		
Имя файла байт 2	2E		
Имя файла байт 3	64		
Имя файла байт 4	61		
Имя файла байт 5	74		
Нулевой байт	00		

#### Ошибка

Код функции	1 Байт	0x5B
Код ошибки	1 Байт	-1 или -2

### 3.13 Функция 92 (5Ch). Рестарт удаленного контроллера

Функция с этим кодом предназначена для перезапуска удаленного контроллера. В поле данных запроса указывается код функции.

Ответ на запрос с этим кодом функции не возвращается.

По данному запросу программа прекращает сбрасывать сторожевой таймер контроллера и контроллер аппаратно перезапускается. Таким образом, для обработки запроса с функцией 5Ch необходимо активизировать сторожевой таймер контроллера.

Запрос

Код функции	1 Байт	0x5C
-------------	--------	------

### 3.14 Функция 110 (6Eh). получить текущее время контроллера

Функция с этим кодом предназначена для получения показаний часов реального времени с удаленного контроллера. Поле данных запроса содержит код функции.

Ответ содержит код функции, секунды, минуты, часы, день, месяц и год. Год упакован в два байта, начиная с младшего байта. Остальные параметры занимают по одному байту в поле данных ответа.

Запрос

Код функции	1 Байт	0x6E
-------------	--------	------

Ответ

Код функции	1 Байт	0x6E
Резерв	1 Байт	0x00
Резерв	1 Байт	0x00
Секунды	1 Байт	От 0x00 до 0x3B
Минуты	1 Байт	От 0x00 до 0x3B
Часы	1 Байт	От 0x00 до 0x17
День	1 Байт	От 0x01 до 0x1F
Месяц	1 Байт	От 0x01 до 0x0C
Год	2 Байта	От 0x0000 до 0xFFFF

Пример запроса с функцией 6Eh:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	6E	Функция	6E
		Резерв	00
		Резерв	00

### 3.15 Функция 111 (6Fh). Установить текущее время контроллера

	Секунда	14
	Минута	19
	Час	06
	День	0A
	Месяц	05
	Год мл.	D4
	Год ст.	07

Ответ вернул время 6:25:20 и дату 10 мая 2004 года.

### 3.15 Функция 111 (6Fh). Установить текущее время контроллера

Функция с этим кодом предназначена для изменения показаний часов реального времени с удаленного контроллера. Поле данных запроса содержит код функции, секунды, минуты, часы, день, месяц и год. Год упакован в два байта, начиная с младшего байта. Остальные параметры занимают по одному байту в поле данных запроса.

Ответ возвращает код функции.

Запрос

Код функции	1 Байт	0x6F
Резерв	1 Байт	0x00
Резерв	1 Байт	0x00
Секунды	1 Байт	От 0x00 до 0x3B
Минуты	1 Байт	От 0x00 до 0x3B
Часы	1 Байт	От 0x00 до 0x17
День	1 Байт	От 0x01 до 0x1F
Месяц	1 Байт	От 0x01 до 0x0C
Год	2 Байта	От 0x0000 до 0xFFFF

Ответ

Код функции	1 Байт	0x6F
-------------	--------	------

Пример запроса с функцией 6Fh, устанавливающий время 17:30:05 и дату 17 мая 2004 года:

Запрос		Ответ	
Название поля	(Hex)	Название поля	(Hex)
Функция	6F	Функция	6F
Резерв	00		
Резерв	00		
Секунда	05		
Минута	1E		
Час	11		
День	11		
Месяц	05		
Год мл.	D4		
Год ст.	07		



**ПРИЛОЖЕНИЕ А**  
**Адресное пространство регистров**  
**(для команд 0x03/0x06/0x16)**

**Для контроллеров:**

**- СГР Эрис 110 МАП DIN с версией прошивки v.2.00.581;**

**- СГР Эрис 130 с версией прошивки v. 2.00.581.**

Пояснения:

r - регистр доступен только для чтения,

w - регистр доступен для чтения и для записи.

Размер каждого регистра 2 байта, тип WORD.

Адрес	Назначение	Диапазон	Доступ
<b>Общие настройки (0x03/0x06/0x16)</b>			
0x0000	Тип контроллера	0x1100 – для Эрис 110 0x1300 – для Эрис 130	r/-
0x0001	Заводской номер (LO)	0...0x9999	r/-
0x0002	Заводской номер (HI)	0...0x9999	r/-
0x0003	Напряжение на батарее часов, *10В		r/-
0x0004	Неисправности в контроллере	– бит 0 – FLASH – бит 1 - FRAM – бит 3 - LAN – бит 4 - ADC	r/-
0x0005	<b>v.2.01.395</b> Общее состояние. Для квитирования (блокировки звука) / сброса аварии необходимо записать в этот регистр значение 0x0200.	– бит 0 – есть датчик со статусом «Инициализация» – бит 1 – есть датчик со статусом «Отсутствие связи» – бит 2 – есть датчик со статусом «Отсутствие связи с сенсором» – бит 3 – есть датчик со статусом «Обслуживание» – бит 4 – есть датчик со статусом «Порог 1» – бит 5 – есть датчик со статусом «Порог 2» – бит 6 – есть датчик со статусом «Порог 3» – бит 7 – есть датчик со статусом «Превышение сигнала» – бит 8 – есть датчик со статусом «Авария» – бит 9 – блокировка звука / квитирование	r/w
<b>Настройки сети (0x03/0x06/0x16)</b>			
0x0100	RS. Slave. Скорость RS, бит/с	0 – 1200 1 – 2400	r/w

ПРИЛОЖЕНИЕ А

Адрес	Назначение	Диапазон	Доступ
		2 – 4800 3 – 9600 4 – 19200 5 – 38400 6 – 57600 7 – 115200	
0x0101	RS. Slave. Сетевой адрес	1...247	r/w
0x0102	RS. Режим работы портов	0: - COM1 – Slave - COM2 – Master 1: - COM1 - Master - COM2 – Slave	r/w
0x0103	COM2. Режим работы	0 – RS232 1 – RS485	r/w
0x0104	LAN. Сетевой адрес. Старший регистр	0...65535	r/w
0x0105	LAN. Сетевой адрес. Младший регистр	0...65535	r/w
0x0106	LAN. Маска подсети. Старший регистр	0...65535	r/w
0x0107	LAN. Маска подсети. Младший регистр	0...65535	r/w
0x0108	LAN. Шлюз. Старший регистр	0...65535	r/w
0x0109	LAN. Шлюз. Младший регистр	0...65535	r/w
0x010A	LAN. HTTP порт	0...65535	r/w
0x010B	LAN. TCP порт	0...65535	r/w
0x010C	LAN. UDP порт	0...65535	r/w

**Данные каналов (0x03)**

	<i>Регистры состояния каналов (побитно)</i>		
1000...1 007	***Наличие связи		
1008...1 015	***Состояние «Порог 1»		
1016...1 023	***Состояние «Порог 2»		
1024...1 031	***Состояние «Авария»		
1032...1 039	***Состояние «Обслуживание»		
1040...1 047	***Состояние «Превышение»		

ПРИЛОЖЕНИЕ А

Адрес	Назначение	Диапазон	Доступ
<p>*** При представлении последовательность из 16 байт (8 регистров) в виде единого регистра, то какой-либо бит этого регистра будет соответствовать какому либо состоянию соответствующего модуля. Раскладка регистров производится в следующем порядке:</p> <p>R0:R1:R2:R3:R4:R5:R6:R7. Например, для канала 25 получаем:</p> <p style="padding-left: 40px;">регистр = 25 / 16 = 1</p> <p style="padding-left: 40px;">бит в регистре = 25 % 16 = 9</p>			
	<i>Группа 0. Состояние канала</i>		
2000	Канал 1. Заводской номер (LO)	0...0x9999	r/-
2001	Канал 1. Заводской номер (HI) <i>Например, заводской номер «112110123» будет записан как HI: 0x1211, LO: 0x0123</i>	0...0x9999	r/-
2002	Канал 1. Тип модуля	111 – потенциальный (крейт) 112 – токовый (крейт) 113 – потенциальный (DIN) 114 – токовый (DIN) 130 – токовый (СГМ Эрис 130) 10 – XCD 11 – XNX 12 – Optima	r/-
2003	Канал 1. Состояние	– бит 0 – авария (нет связи с датчиком) – бит 1 – порог 1 – бит 2 – порог 2 – бит 3 – кнопка «Сброс» – бит 4 – «Обслуживание» – бит 5 – превышение сигнала – бит 7 – нет связи – бит 8 – ошибка АЦП – бит 9 – в аварийном режиме блокировка звука при первом нажатии, при втором - СБРОС аварии – бит 10 – режим 0 – рабочий, 1 - сервисный (для Optima) – бит 11 – предупреждение "Warning" (для XCD, XNX, Optima) – бит 12 – нет связи с датчиком (для Optima) – бит 13 – авария (какие-либо проблемы с датчиком) (для Optima)	r/-
2004	Канал 1. Текущее значение АЦП	0...65535	r/-
2005	Канал 1. Текущее значение тока * 1000 мА <i>Например, ток 12.456 мА будет</i>	0...65535	r/-

ПРИЛОЖЕНИЕ А

Адрес	Назначение	Диапазон	Доступ
	<i>записан как 12456</i>		
2006	Канал 1. Текущее значение величины *10 <i>Например, текущая концентрация 12,3 будет записана как 123.</i>	0...65535	r/-
2007	Канал 1. Текущее значение ШИМ (ЦАП)	0...1023	r/-
2008	Канал 1. Текущее значение тока (ЦАП), *100 мА	0...2200	r/-
2009...2017	Канал 2 (...)		
	.....		
	<i>Группа 1. Настройки АЦП</i>		
3000	Канал 1. Значение АЦП соответствующие 4 мА / Начальная точка	0...65535	r/-
3001	Канал 1. Значение АЦП соответствующие 20 мА / Конечная точка	0...65535	r/-
3002	Канал 1. Концентрация * 10 ( <b>Только для потенциальных модулей</b> ) <i>Например, концентрация 46,7 будет записана как 467.</i>	0...1000	r/-
3003...3005	Канал 2 (...)		
	.....		
	<i>Группа 2. Настройки ЦАП</i>		
4000	Канал 1. Настройки токового выхода <i>Например, задание тока на выходе 12.45 мА в ручном виде будет выглядеть 0x44DD</i>	- биты 0..11 – значение тока (в мА * 100) - биты 14..15 – тип задания: – 0x00 – автоматический – 0x01 – ручной – 0x02 – точка 4 мА – 0x03 – точка 20 мА	r/-
4001	Канал 1. Код ШИМ равный току 4 мА	0...1023	r/-
4002	Канал 1. Код ШИМ равный току 20 мА	0...1023	r/-
4003...4005	Канал 2 (...)		
	.....		
	<i>Группа 3. Конфигурация канала</i>		
5000	Канал 1. Начальное значение величины *10	0...9999 (40)	r/-
5001	Канал 1. Конечное значение величины соответствующие 20 мА (200 мВ) *10	0...9999 (200)	r/-

ПРИЛОЖЕНИЕ А

Адрес	Назначение	Диапазон	Доступ
5002	Канал 1. «Мёртвая» зона, * 10	0...99	
5003	Канал 1. Ток питания датчика в мА <b>(Только для потенциальных модулей)</b> <i>Например, ток 75 мА будет записан как 75 (0x004B).</i>	50...200	r/-
5004	Канал 1. Порог 1 *10. Если равен 0, то порог не обрабатывается. Если ст.бит (15) равен 0, то порог обрабатывается на превышение, а если равен 1, то на снижение. <i>Например, порог 12,5 будет записан как 125 (0x007D).</i>	0...9999 (0)	r/-
5005	Канал 1. Порог 2 *10. Если равен 0, то порог не обрабатывается. Если ст.бит (15) равен 0, то порог обрабатывается на превышение, а если равен 1, то на снижение. <i>Например, порог 12,5 будет записан как 125 (0x007D).</i>	0...9999 (0)	r/-
5006	Канал 1. Порог 3 *10. Если равен 0, то порог не обрабатывается. Если ст.бит (15) равен 0, то порог обрабатывается на превышение, а если равен 1, то на снижение. <i>Например, порог 12,5 будет записан как 125 (0x007D).</i>	0...9999 (0)	r/-
5007	Канал 1. Гистерезисы *10 <i>Например, гистерезисы по 2,5 будут записаны как 6425 (0x1919).</i>	мл.байт: гистерезис 1 ст.байт: гистерезис 2	r/-
5008	Канал 1. Гистерезисы *10 <i>Например, гистерезисы по 2,5 будут записаны как 6425 (0x1919).</i>	мл.байт: гистерезис 3 ст.байт: резерв	r/-
5009	Канал 1. Задержки срабатывания порогов, в секундах <i>Например, задержки по 5 секунд будут записаны как 1285 (0x0505).</i>	мл.байт: для порога 1 ст.байт: для порога 2	r/-
5010	Канал 1. Задержки срабатывания порогов, в секундах <i>Например, задержки по 5 секунд будут записаны как 1285 (0x0505).</i>	мл.байт: для порога 3 ст.байт: резерв	r/-
5011	Канал 1. Время автоматического сброса аварии, в секундах	0..200	r/-
5012	Канал 1. Время автоматического сброса порога 1, в секундах	0..200	r/-
5013	Канал 1. Время автоматического сброса порога 2, в секундах	0..200	r/-
5014	Канал 1. Время автоматического	0..200	r/-

ПРИЛОЖЕНИЕ А

Адрес	Назначение	Диапазон	Доступ
	сброса порога З, в секундах		
5015	Канал 1. Настройки модуля	<p>Биты 0..3 – тип газа:</p> <p>0 – канал отключен</p> <p>1 – CH</p> <p>2 – O<sub>2</sub></p> <p>3 – H<sub>2</sub>S</p> <p>4 – SO<sub>2</sub></p> <p>5 – NO</p> <p>6 – NO<sub>2</sub></p> <p>7 – Cl<sub>2</sub></p> <p>8 – NH<sub>3</sub></p> <p>9 – CO</p> <p>10 – CO<sub>2</sub></p> <p>Биты 4..7 – единица измерения:</p> <p>0 – мг/м<sup>3</sup></p> <p>1 – % об.д.</p> <p>2 – ppm</p> <p>3 – ppb</p> <p>4 – млн.<sup>-1</sup></p> <p>5 – %НКПР</p> <p>6 – % НПВ</p> <p>7 – % LEL</p> <p>8 – mA</p> <p>9 – LEL*M</p> <p>10 – %Vol</p> <p>11 – г/м<sup>3</sup></p> <p>12 – UEG</p> <p>13 – Ratio</p> <p>14 – ppm*m</p> <p>15 – EG*m</p> <p>Биты 8..9 – тип сброса аварии:</p> <p>0 – автоматический</p> <p>1 – ручной</p> <p>Биты 10..14 – резерв</p> <p>Бит 15– резерв дискретность для СГМ 112(4)</p>	r/-
50016... 5031	Канал 2 (...)		
	.....		
	<i>Группа 4. Конфигурация архивации канала</i>		
6000	Канал 1. Тип архивации	<p>0 – отключено</p> <p>1 – интервальный</p> <p>2 – дельта</p>	r/-
6001	Канал 1. Интервал архивации, в	0..18000	r/-

ПРИЛОЖЕНИЕ А

Адрес	Назначение	Диапазон	Доступ
	секундах		
6002	Канал 1. Контрольная точка	мл.байт: минуты ст.байт (биты 0-6): часы ст.байт (бит 7): использовать контрольную точку	r/-
6003	Канал 1. Дельта <i>Например, дельта 12,5 будет записана как 125 (0x007D).</i>	0,5...50,0%	r/-
6004	Канал 1. Интервал контроля дельта	0,5...50,0%	r/-
6005	Канал 1. Резерв		r/-
6006	Канал 1. Резерв		r/-
6007...6 013	Канал 2 (...)		
	•••••		
	<i>Группа 5. Конфигурация связи</i>		
7000	Сетевой адрес	1...247	r/-
7001	Скорость	0 – 1200 1 – 2400 2 – 4800 3 – 9600 4 – 19200 5 – 38400 6 – 57600 7 – 115200	r/-

ПРИЛОЖЕНИЕ А

Адресное пространство регистров каналов (начальные адреса групп)

Канал	Группа 0	Группа 1	Группа 2	Группа 3	Группа 4	Группа 5
Канал 1	2000	3000	4000	5000	6000	7000
Канал 2	2009	3003	4003	5016	6007	7002
Канал 3	2018	3006	4006	5032	6014	7004
Канал 4	2027	3009	4009	5048	6021	7006
Канал 5	2036	3012	4012	5064	6028	7008
Канал 6	2045	3015	4015	5080	6035	7010
Канал 7	2054	3018	4018	5096	6042	7012
Канал 8	2063	3021	4021	5112	6049	7014
Канал 9	2072	3024	4024	5128	6056	7016
Канал 10	2081	3027	4027	5144	6063	7018
Канал 11	2090	3030	4030	5160	6070	7020
Канал 12	2099	3033	4033	5176	6077	7022
Канал 13	2108	3036	4036	5192	6084	7024
Канал 14	2117	3039	4039	5208	6091	7026
Канал 15	2126	3042	4042	5224	6098	7028
Канал 16	2135	3045	4045	5240	6105	7030
Канал 17	2144	3048	4048	5256	6112	7032
Канал 18	2153	3051	4051	5272	6119	7034
Канал 19	2162	3054	4054	5288	6126	7036
Канал 20	2171	3057	4057	5304	6133	7038
Канал 21	2180	3060	4060	5320	6140	7040
Канал 22	2189	3063	4063	5336	6147	7042
Канал 23	2198	3066	4066	5352	6154	7044
Канал 24	2207	3069	4069	5368	6161	7046
Канал 25	2216	3072	4072	5384	6168	7048
Канал 26	2225	3075	4075	5400	6175	7050
Канал 27	2234	3078	4078	5416	6182	7052
Канал 28	2243	3081	4081	5432	6189	7054
Канал 29	2252	3084	4084	5448	6196	7056
Канал 30	2261	3087	4087	5464	6203	7058
Канал 31	2270	3090	4090	5480	6210	7060
Канал 32	2279	3093	4093	5496	6217	7062
Канал 33*	2288	3096	4096	5512	6224	7064
Канал 34*	2297	3099	4099	5528	6231	7066



ПРИЛОЖЕНИЕ А

Канал 35*	2306	3102	4102	5544	6238	7068
Канал 36*	2315	3105	4105	5560	6245	7070
Канал 37*	2324	3108	4108	5576	6252	7072
Канал 38*	2333	3111	4111	5592	6259	7074
Канал 39*	2342	3114	4114	5608	6266	7076
Канал 40*	2351	3117	4117	5624	6273	7078

Размер групп (в регистрах (WORD))

Группа 0	Группа 1	Группа 2	Группа 3	Группа 4	Группа 5
9	3	3	16	7	2

Расчёт адреса регистра

$$\text{Adr\_reg} = \text{BASE} + \text{CH} * \text{SIZE} + \text{OFFSET}$$

, где:

- «BASE» - базовый адрес группы;
- «CH» - номер канала;
- «SIZE» - размер группы;
- «OFFSET» - смещение регистра внутри группы.

**ПРИЛОЖЕНИЕ Б**  
**Структура файлов архивов**

Имя	Описание
NS.arh	Архив нештатных ситуаций
ch1.arh	Архив данных канала 1
ch2.arh	Архив данных канала 2
...	...
ch40.arh	Архив данных канала 40

Каждый файл архива состоит из набора записей: заголовков и архивные данные.

Структура заголовка:

```
typedef struct
{
    unsigned char  Priznak; // Признак записи
    unsigned char  arh[3];  // Строка ARH
    unsigned char  Number;  // Номер канала
    unsigned char  Config;  // Тип газа, единица
    unsigned int   Limit1;  // Порог 1
    unsigned int   Limit2;  // Порог 2
} TARhHead;
```

Структура записи данных:

```
typedef struct
{
    unsigned char  Priznak; // Признак записи   FCh
    unsigned long  Time;    // Дата и время события
    unsigned char  Channel; // Канал
    unsigned int   Value;   // Значение *10
    unsigned char  Event;   // Событие
} TARhNote;
```

Типы возможных событий (поле Event) для файла «NS.arh»:

- 0 - нет
- 1 - Отказ датчика (сенсора)
- 2 - Сработал порог 1
- 3 - Сработал порог 2
- 4 - Превышение сигнала
- 5 - Режим «обслуживание»
- 6 - Ошибка связи
- 7 - Выход из нештатной ситуации
- 8 - Включение прибора
- 9 - Выключение прибора
- 10 - Сброс времени
- 11 - Отказ АЦП
- 12 - Блокировка звука
- 13 - "Warning" (для XCD, XNX, Optima)

- 14 - "Alarm" (для Optima)
- 15 - "Low Signal" (для Optima)
- 16 - "Beam-Block" (для Optima)
- 17 - "Off-Line" (для Optima)
- 18 - Нет связи с датчиком (для Optima)

Типы возможных событий (поле Event) для файла «chX.arh»:

- 0 - нет
- 1 - обрыв датчика
- 2 - порог 1
- 3 - порог 2
- 4 - превышение сигнала