



**СИСТЕМА СБОРА ДАННЫХ,  
ДИСПЕТЧЕРИЗАЦИИ И УПРАВЛЕНИЯ  
«ССДУ-03»**

**ИНСТРУКЦИЯ ОПЕРАТОРА  
ПО РАБОТЕ С ПОСЛЕДОВАТЕЛЬНЫМ КАНАЛОМ СВЯЗИ**

**СИФП 102.00.000 И1  
версия 1.0**



МИНСК 2013

## СОДЕРЖАНИЕ

Введение.....	4
1 Краткие технические характеристики ССДУ .....	4
2 Особенности протокола обмена .....	5
2.1 Типы данных .....	6
2.1.1 Одноэлементное множество .....	7
2.1.2 Упорядоченные множества по возрастанию .....	7
2.1.3 Упорядоченные множества по убыванию .....	8
2.1.4 Неупорядоченное множество .....	8
2.1.5 16-ти разрядная битовая маска .....	8
2.2 Два формата заголовков .....	9
3 Описание команд протокола.....	10
3.1 Транзитный обмен .....	11
3.2 Изменение паролей .....	11
3.3 Управление режимами работы ССДУ .....	11
3.3.1 Чтение текущего режима .....	12
3.3.2 Установка режима.....	12
3.4 Параметры последовательных портов .....	13
3.4.1 Чтение параметров последовательных портов .....	14
3.4.2 Запись параметров последовательных портов .....	15
3.5 Конфигурация узлов .....	15
3.5.1 Чтение конфигурации узлов .....	17
3.5.2 Запись конфигурации узлов .....	17
3.6 Дата и время ССДУ .....	17
3.6.1 Чтение текущей даты и времени .....	17
3.6.2 Запись текущей даты и времени.....	18
3.7 Календарь выходных дней .....	18
3.7.1 Чтение календаря выходных дней.....	19
3.7.2 Запись календаря выходных дней .....	19
3.7.3 Обнуление календаря выходных дней.....	20
3.8 Дата и время переключения сезонов.....	20
3.8.1 Чтение дат и времени переключения сезонов.....	20
3.8.2 Запись дат и времени переключения сезонов .....	20
3.8.3 Обнуление даты и времени начала сезонов .....	21
3.9 Архивы ССДУ .....	21
3.9.1 Архив состояний обмена.....	21
3.9.2 Архив корректировок .....	22
3.9.3 Чтение архивов.....	22
3.10 Архивные данные по узлам за период времени.....	23
3.11 Конфигурация расписания .....	24
3.11.1 Чтение конфигурации расписания .....	26
3.11.2 Запись конфигурации расписания.....	26
3.11.3 Очистка конфигурации расписания .....	26
3.12 Текущие данные по множеству узлов.....	27
3.13 Конфигурация встроенного GSM-модуля .....	27
3.13.1 Параметры GSM-модуля .....	27
3.14 Другие команды .....	28

**СИСТЕМА СБОРА ДАННЫХ, ДИСПЕТЧЕРИЗАЦИИ И УПРАВЛЕНИЯ «ГРАН-ЭЛЕКТРО ССДУ»**  
**ИНСТРУКЦИЯ ОПЕРАТОРА ПО РАБОТЕ С ПОСЛЕДОВАТЕЛЬНЫМ КАНАЛОМ СВЯЗИ**  
СИФП 102.00.000 И1

---

<b>Приложение А</b> .....	29
<b>Приложение Б</b> .....	31
<b>Приложение В</b> .....	33
<b>Приложение Г</b> .....	34
<b>Приложение Д</b> .....	35
Для заметок .....	40

## **Введение**

Настоящая инструкция оператора по работе с последовательным каналом связи предназначена для правильного подключения системы сбора данных, диспетчеризации и управления «ССДУ-03» (далее – ССДУ) к внешним устройствам и содержит технические характеристики и протокол обмена.

## **1 Краткие технические характеристики ССДУ**

ССДУ предназначена для построения распределенных систем АСКУЭ на базе устройств, поддерживающих протокол М-Bus.

ССДУ имеет в своем составе 4 последовательных порта (СОМ1...СОМ4).

Порт СОМ1 подключен к интерфейсу опто-порта и платы расширения и предназначен для подключения к более высокому уровню иерархии. Выбор текущего интерфейса осуществляется с помощью соответствующих клавиш (верхняя клавиша включает модуль расширения, нижняя клавиша включает опто-порт). Если СОМ1 находится в режиме опто-порта и в течение пяти минут не было входящих запросов, то СОМ1 автоматически переключится в режим платы расширения. Параметры связи СОМ1 в режиме опто-порт – скорость 9600, 8 бит данных, без паритета, один стоповый бит.

Порт СОМ2 имеет интерфейс RS-232 и предназначен для подключения к более высокому уровню иерархии.

Порт СОМ3 имеет интерфейс RS-232 и предназначен для подключения приборов учета к ССДУ.

Порт СОМ4 имеет интерфейсы RS-232 и RS-485 (не допускается одновременное использование) и предназначен для подключения приборов учета к ССДУ.

Максимальная скорость обмена по каждому порту 115200 бод.

Программное обеспечение ССДУ позволяет сконфигурировать каждый из портов, а также задать ему функцию, которую он будет выполнять в составе АСКУЭ.

Внешний порт предназначен для подключения к более высокому уровню иерархии АСКУЭ и постоянно находится в состоянии приема. К локальному порту могут быть подключены приборы учета.

Каждое ССДУ может обслуживать до 399 подчиненных устройств, в дальнейшем называемых узлами. Узлы могут быть распределены по разным портам, причем их заводские номера должны быть уникальны в пределах одного порта.

Каждый узел должен быть зарегистрирован в базе данных конфигурации ССДУ (в дальнейшем по тексту – БДК).

Каждая запись БДК содержит следующие параметры:

- тип узла;
- серийный номер;
- номер порта, к которому он подключен;

и т.д.

Индекс записи в БДК является логическим адресом узла. Нулевой индекс предназначен для обращения к самому ССДУ.

ССДУ в автоматическом режиме выполняет задания, определенные в расписании системы:

- Синхронизация времени ССДУ со временем заданного NTP-сервера.
- Опрос узлов М-Bus – позволяет считывать показания с точек учета и сохранять их в архиве ССДУ. Включает в себя четыре задания, в каждом из которых

можно индивидуально настроить расписание запуска и группу опрашиваемых узлов.

- E-Mail, текущие значения – позволяет считывать и передавать показания точек учета в виде E-Mail сообщений. Включает в себя два задания, в каждом из которых можно индивидуально настроить расписание запуска и группу опрашиваемых узлов.
- E-Mail, архивы – позволяет передавать показания, хранящиеся в архиве ССДУ в виде E-Mail сообщений. Включает в себя два задания, в каждом из которых можно индивидуально настроить расписание запуска и группу узлов.
- FTP, архивы – позволяет передавать показания, хранящиеся в архиве ССДУ на заданный FTP-сервер. Включает в себя два задания, в каждом из которых можно индивидуально настроить расписание запуска и группу узлов.

ССДУ ведет архив корректировок. Размер архива составляет 100 записей. В архиве корректировок отмечаются все внешние воздействия, включая изменения конфигурации узлов, включение и отключение сетевого напряжения и т.д.

Для передачи данных на верхний уровень используются пассивные и активные способы передачи данных.

К пассивным способам передачи данных относятся:

1. Проводной интерфейс.
2. Оптический интерфейс.
3. CSD соединение (GSM-модем).
4. TCP соединение (соединение через интернет).

К активным способам передачи данных относятся:

1. Передача данных с помощью E-Mail сообщений.
2. Передача данных на FTP сервер.

## **2 Особенности протокола обмена**

Протокол обмена с ССДУ полностью совместим с протоколом обмена «Гран-Система-С». В протокол для ССДУ добавлены две новые функции N1 и N2, расширен перечень возможных ошибок и добавлен расширенный вариант заголовка.

Протокол предусматривает наличие на одной шине одного главного и до 399 подчиненных устройств. Подключение подчиненных устройств допускается только на локальный порт (СОМ3, СОМ4). Протокол имеет следующие особенности:

- Для обеспечения высокой достоверности передачи используется избыточный циклический код (CRC). Более подробно циклический код описан в приложении А.
- Максимальный размер пакета не должен превышать 1024 байта.
- Если в пакет помещаются типы данных, состоящие из нескольких байт, то самый младший байт типа должен следовать первым.
- Протокол адаптирован для работы в сетях с негарантированным временем доставки пакетов, в частности для сетей GSM (см. раздел «Два формата заголовков»).

## **2.1 Типы данных**

В зависимости от типа запроса и ответа в протоколе могут использоваться различные типы данных: целые 2-х и 4-х байтные числа, числа с плавающей запятой в соответствии с IEEE 754, множества, а также различные структуры. В дальнейшем по тексту при описании типов данных будет использоваться нотация языка «C». Используются следующие обозначения типов данных:

- *unsigned char* либо *uchar* – однобайтное число без знака;
- *signed char* либо *char* – однобайтное число со знаком;
- *unsigned short* либо *ushort* – двухбайтное число без знака;
- *signed short* либо *short* – двухбайтное число со знаком;
- *unsigned long* либо *ulong* – четырехбайтное число без знака;
- *bool* – однобайтное число (*true*(1) / *false*(0));
- *float* – четырехбайтное число с плавающей запятой в соответствии с IEEE 754.

При передаче по каналу связи младший байт многобайтного числа всегда передается первым.

Для представления даты и времени в протоколе используется следующая 6-ти байтная структура:

```
struct TDateAndTime
{
    unsigned char    sec;
    unsigned char    min;
    unsigned char    hour;
    unsigned char    day;
    unsigned char    month;
    unsigned char    year;
};
```

где: *sec* – секунды (допустимые значения 0...59);  
*min* – минуты (допустимые значения 0...59);  
*hour* – часы (допустимые значения 0...23);  
*day* – день (допустимые значения от 1 до максимального дня в месяце);  
*month* – месяц (допустимые значения 1...12);  
*year* – год (календарный год минус 2000).

Другие типы структур будут рассмотрены более подробно при описании запросов и ответов в разделе «Описание команд протокола». В этом разделе подробно будет расписан только тип «множество», так как он используется практически в каждом запросе.

Для задания упорядоченной или неупорядоченной последовательности однобайтных чисел используется тип данных «множество». Каждое число, входящее в последовательность в зависимости от контекста может интерпретироваться как беззнаковое (0...255 – для однобайтных множеств, 0...65535 – для двухбайтных множеств) или как число со знаком (-128...+127 – для однобайтных множеств, -32768...+32767 – для двухбайтных множеств). Тип данных «множество» имеет следующий формат:

тип	аргументы
-----	-----------

**Рисунок 1**

Поле «тип» определяет тип множества и имеет размерность 1 байт. Поле «аргументы» зависит от поля «тип» и может иметь размерность от 1 до 256 байт.

В настоящее время протокол поддерживает 12 типов множеств. Перечисление типов множеств приведено в приложении В.

### 2.1.1 Одноэлементное множество

В протоколе присутствуют два типа одноэлементных множеств:

- однобайтное одноэлементное множество (тип 0);
- двухбайтное одноэлементное множество (тип 7).

Этот тип множества предназначен для определения множества, состоящего из одного элемента. Поле «аргументы» имеет размерность 1 либо 2 байта в зависимости от типа. Пример представления числа 23 показан на рисунке 2.

тип	аргументы
0	23
7	23   0

Рисунок 2

### 2.1.2 Упорядоченные множества по возрастанию

В протоколе определены четыре варианта упорядоченных множеств по возрастанию:

- однобайтное, с указанием первого и последнего значения (тип 1);
- однобайтное, с указанием первого значения и количества (тип 2);
- двухбайтное, с указанием первого и последнего значения (тип 8);
- двухбайтное, с указанием первого значения и количества (тип 9).

Все вышеперечисленные множества задаются при помощи двух аргументов. Первым аргументом является начальный элемент множества. Вторым аргументом для множеств типа 1 и 8 является последний элемент множества, а для множеств типа 2 и 9 – количество элементов. Для однобайтных множеств размерность аргументов – один байт, для двухбайтных – два байта.

На рисунке 3 показан пример задания множества чисел [5...15] с использованием упорядоченных множеств по возрастанию.

тип	аргументы
1	5   15
8	5   0   15   0
2	5   11
9	5   0   11   0

Рисунок 3

Если значение первого элемента множества будет больше последнего (для типов 1 и 8), то результирующее множество будет представлять собой последовательность чисел с переходом через 0. Например, если первый элемент равен 7, а последний равен 5, то такое множество будет представлять следующую последовательность:

7,8,9...254,255,0,1,2...5 (для типа 1).

Аналогичная ситуация будет и для множеств с типами 2 и 9, если сумма значений первого элемента и количества элементов превысит 255 (65535).

### 2.1.3 Упорядоченные множества по убыванию

Так же как и для множеств по возрастанию, так и для множеств по убыванию в протоколе определено 4 варианта:

- однобайтное, с указанием первого и последнего значения (тип 3);
- однобайтное, с указанием первого значения и количества (тип 4);
- двухбайтное, с указанием первого и последнего значения (тип 10);
- двухбайтное, с указанием первого значения и количества (тип 11).

Все выкладки, приведенные в предыдущем абзаце, относятся и к этим типам множеств, за исключением того, что они определяют последовательность чисел по убыванию. Например, если требуется сформировать последовательность: 0,-1,-2...-12, то форматы множеств для типов 3(10) и 4(11) должны быть представлены как на рисунке 4 и рисунке 5.

тип	аргументы			
3	0	244 (-12)		
10	0	0	244(-12)	255

Рисунок 4

тип	аргументы			
4	0	13		
11	0	0	13	0

Рисунок 5

### 2.1.4 Неупорядоченное множество

Неупорядоченное множество (тип 5) позволяет задать произвольный набор чисел. Первый байт поля «*аргументы*» должен содержать количество элементов множества, а остальные байты должны представлять значения последовательности. Размерность аргументов – один байт. Пример формата неупорядоченного множества для последовательности 19,7,200,93,77,0,5,3 приведен на рисунке 6.

тип	аргументы								
5	8	19	7	200	93	77	0	5	3

Рисунок 6

### 2.1.5 16-ти разрядная битовая маска

Это множество (тип 6) позволяет сформировать произвольную последовательность однобайтных чисел по возрастанию в диапазоне 0...15. Поле «*аргументы*» двухбайтное и содержит 16-ти разрядную маску (младший байт первый). Последовательность чисел формируется исходя из порядковых номеров битов в маске, установленных в '1'. На рисунке 7 показан пример задания последовательности чисел 1,2,4,5,8,10,14,15 (байты маски представлены в шестнадцатеричном формате).

тип	аргументы	
6	36h	C5h

Рисунок 7



## **2.2 Два формата заголовков**

В протоколе определено два варианта заголовков: простой и расширенный. Формат простого заголовка для пакета-запроса имеет размерность 4 байта и представлен на рисунке 8.

адрес	1	параметр	операция
-------	---	----------	----------

**Рисунок 8**

Поле «адрес» должно содержать 0, адрес устройства или FFh при широковещательной адресации. Значение второго поля для простого заголовка всегда должно содержать число 1. Поле «параметр» определяет тип запроса. Кодировка параметров приведена в Приложении В. Поле «операция» уточняет тип операции, которая должна быть выполнена для параметра, заданного в поле «параметр». Кодировка операций приведена в Приложении В. Формат заголовка для ответного пакета представлен на рисунке 9.

адрес	1(81h)	параметр	результат
-------	--------	----------	-----------

**Рисунок 9**

Как видно из рисунка, заголовки запроса и ответа различаются только последним полем. Поле «результат» содержит результат выполнения запроса. Число 0 свидетельствует об успешном выполнении запроса, а не 0 – об ошибке. В случае ошибки, во втором поле дополнительно будет установлен в '1' старший бит. Кодировка результата выполнения запроса приведена в Приложении В.

Формат расширенного заголовка имеет размер 8 байт и представлен на рисунке 10.

адрес	2	размер пакета (2 байта)	N пакета	флаги	параметр	операция
-------	---	-------------------------	----------	-------	----------	----------

**Рисунок 10**

Как видно из рисунка, расширенный заголовок отличается от простого наличием 3-х дополнительных полей. Поле «размер пакета» в запросе и ответе должно содержать количество байт пакета, включая заголовок, сообщение и CRC. Размер этого поля составляет 2 байта. Поле «N пакета» предназначено для идентификации пакета. Поле «флаги» уточняет тип пакета. В настоящее время задействован только самый младший бит этого поля. Значение этого бита должно быть равно '0' для запроса и '1' для ответа. Оставшиеся 7 старших бит обязательно должны быть установлены в '0', так как они зарезервированы для дальнейшего использования. Формат расширенного заголовка для ответного пакета приведен на рисунке 11.

адрес	2(82h)	размер пакета (2 байта)	N пакета	флаги	параметр	результат
-------	--------	-------------------------	----------	-------	----------	-----------

**Рисунок 11**

Так же как и для простого заголовка, поле «операция» заменяется полем «результат». Кодировка результата выполнения запроса приведена в Приложении В.

Как уже упоминалось выше, поле «N пакета» позволяет присвоить каждому пакету уникальный номер, что позволяет однозначно отслеживать соответствие полученного ответа посланному запросу. Такая возможность особенно актуальна в сетях с

негарантированным временем доставки пакетов. Проблема заключается в том, что такие сети (например, GSM) могут задерживать доставку пакета на время превышающее тайм-аут ожидания ответа. В этом случае, главное устройство может интерпретировать задержку ответа как его отсутствие и послать следующий запрос, а далее получить ответ на предыдущий запрос. Чтобы исключить такую ситуацию, устройство инициатор запроса может нумеровать каждый запрос уникальным номером, а затем при приеме игнорировать все ответы, в которых поле «*N пакета*» не совпадает с аналогичным полем запроса. Частично такая возможность реализована и в простом заголовке. Для нумерации пакетов могут использоваться 2 старших бита поля «*параметр*», так как для нумерации параметров используются только 6 младших бит.

В дополнение к тому, что простой и расширенный заголовки различаются размерами, они используют различные алгоритмы для определения завершения приема пакета. Признаком завершения приема пакета для простого заголовка является тайм-аут между байтами превышающий время передачи 7 байт (но не менее 16 мс). Для расширенного заголовка используется следующий алгоритм:

- если принято меньше 4-х первых байт заголовка, то используется алгоритм определения завершения пакета такой же, как и для короткого заголовка;
- если принято 4 и более байт, то признаком завершения считаются либо прием количества байт, соответствующего значению поля ‘Размер пакета’, либо тайм-аут между байтами, превышающий время передачи 350 байт.

### **3 Описание команд протокола**

При описании команд протокола используются следующие соглашения:

- после названия команды всегда приводится номер параметра и допустимые операции, которые можно для него выполнить;
- структуры и типы данных используют нотацию языка «C»;
- для обозначения заголовка (см. раздел «Два формата заголовков») используется сокращение ‘HDR’;
- для обозначения типа данных «множество» (см. раздел «Типы данных») используется следующее обозначение  $M\{name\}(range)$ , где «*name*» название множества, а «*range*» диапазон допустимых значений;
- для обозначения массивов используется следующее обозначение:

$A[name_1][name_2][name_3]$

где «*name\_x*» – имя множества, по которому индексируется массив.

Для многих параметров имеется возможность рассчитать размер ответа в байтах при выполнении операции чтения. В этом случае необходимо сформировать запрос как для операции чтения, но в поле «*операция*» заголовок необходимо установить значение 3. Ответный пакет будет иметь следующий формат:

HDR	размер	CRC
-----	--------	-----

**Рисунок 12**

Поле «*размер*» это двухбайтное число, которое содержит размер ответа в байтах, включая заголовок и CRC. В дальнейшем при описании команд не акцентируется внимание на этой операции. Чтобы определить параметры, для которых она допустима необходимо руководствоваться приложением Б «Кодировка типов параметров» или обращать внимание на строку «Допустимые операции» после названия команды.

При описании команд часто используются выражение «узлы». Понятие «узлы» включает в себя все подключенные к ССДУ устройства.

### **3.1 Транзитный обмен**

Номер параметра: 1

На данный момент данная функция зарезервирована для последующей реализации в протоколе ССДУ-03. Функция будет предназначена для прямого обращения к устройствам с нестандартными протоколами связи.

### **3.2 Изменение паролей**

Номер параметра: 2

Допустимые операции: запись

Для защиты ССДУ от несанкционированного вмешательства в его работу используются два вида паролей: пароль администратора и коммерческий пароль. Пароль администратора (код 0) защищает данные, отвечающие за настройки самого ССДУ, последовательных портов, конфигурации узлов и т.д. Коммерческий пароль (код 1) защищает данные, влияющие на коммерческий учет. К таким данным относятся: тарифные расписания, календари выходных и праздничных дней, установка времени и т.д.

Предприятием-изготовителем ССДУ поставляются с паролем администратора и коммерческим паролем равным «00000000», где каждый символ это ASCII код символа 0 (десятичный код 48, шестнадцатеричный код 30h). В протоколе предусмотрена операция записи пароля и запрещена операция чтения. Формат пакета для изменения пароля приведен на рисунке 13.

HDR	M{тип пароля} (0,1)	A[тип пароля] <старый пароль, новый пароль>	CRC
-----	---------------------	---	-----

**Рисунок 13**

Пример пакета с простым заголовком заменяющего коммерческий пароль «00000000» на пароль «12345678» приведен на рисунке 14.

заголовок				M		старый пароль								новый пароль								CRC		
A	1	2	1	0	1	48	48	48	48	48	48	48	48	48	48	49	50	51	52	53	54	55	56	CRC

**Рисунок 14**

Аналогичным образом, используя другой тип множества, можно заменить оба пароля.

### **3.3 Управление режимами работы ССДУ**

Номер параметра: 3

Допустимые операции: чтение, запись

ССДУ может функционировать в следующих режимах:

- режим инициализации (код 0);
- режим опроса (код 1);
- режим конфигурирования (код 2);
- режим автоконфигурирования (код 3).

Каждый из режимов предназначен для выполнения определенных функций.

В режим инициализации ССДУ переходит при подаче питания. В этом режиме происходит внутреннее тестирование ССДУ, настройка и конфигурирование внутренних данных.

Опрос – это основной режим работы ССДУ. В этот режим устройство автоматически входит после завершения инициализации и выполняет все операции по синхронизации и опросу.

Режим конфигурирования, как следует из названия, предназначен для настройки и конфигурирования ССДУ. Переход в этот режим возможен только из режима опроса и при задании правильного пароля администратора. Большинство команд записи могут быть выполнены только в этом режиме.

В режиме автоконфигурирования ССДУ автоматически находит и регистрирует устройства, подключенные к локальным портам. Возможны два варианта автоконфигурирования: замена и добавление. При замене – все уже зарегистрированные устройства удаляются и заменяются новыми. При добавлении – к уже зарегистрированным устройствам добавляются новые. Поиск подключенных устройств выполняется только по тем портам, у которых параметр *status* имеет значение «локальный» (см. раздел «Параметры последовательных портов»), поэтому очень важно перед включением режима автоконфигурирования правильно настроить параметры последовательных портов. Переход в режим автоконфигурирования возможен только из режима конфигурирования, а возврат осуществляется автоматически после завершения поиска всех подключенных устройств (возврат происходит в режим конфигурирования).

### 3.3.1 Чтение текущего режима

Запрос для чтения текущего режима приведен на рисунке 15.

HDR	CRC
-----	-----

Рисунок 15

В ответном сообщении ССДУ возвращает пакет, представленный на рисунке 16.

HDR	режим (1 б.)	узел (2 б.)	% (1 б.)	маска (4 б.)	CRC
-----	--------------	-------------	----------	--------------	-----

Рисунок 16

Как видно из рисунка, поле «сообщение» состоит из 4 дополнительных полей. Поле «режим» содержит код текущего режима работы ССДУ. Поле «узел» содержит индекс текущего обрабатываемого узла. Поле «%» отображает ход выполнения текущего процесса и имеет смысл только для режима автоконфигурирования. Поле «маска» отображает маску заводского номера искомого узла в формате VCD и имеет смысл только в режиме автоконфигурирования.

### 3.3.2 Установка режима

Форматы запросов для установки различных режимов отличаются один от другого, поэтому ниже будут подробно расписаны форматы для каждого режима. Протокол не предусматривает возможности установки режима инициализации.

Формат пакета для установки режима опроса приведен на рисунке 17.

HDR	1	CRC
-----	---	-----

**Рисунок 17**

Формат пакета для установки режима конфигурирования приведен на рисунке 18.

HDR	2	пароль администратора (8 байт)	CRC
-----	---	--------------------------------	-----

**Рисунок 18**

19. Формат пакета для установки режима автоконфигурирования приведен на рисунке 19.

HDR	3	параметр	CRC
-----	---	----------	-----

**Рисунок 19**

Поле «параметр» позволяет задать два варианта автоконфигурирования:

- 0 – заменяет текущую конфигурацию узлов новыми;
- 1 – оставляет текущую конфигурацию узлов и добавляет новые узлы.

### **3.4 Параметры последовательных портов**

Номер параметра: 5

Допустимые операции: чтение, подсчет размера ответа, запись

ССДУ имеет в своем составе 4 последовательных порта (COM1...COM4). Конфигурация каждого из портов описывается следующей структурой:

```
struct TComPrm
```

```
{  
    unsigned char    status;        //[0]  
    unsigned char    type_interface // [1]  
    unsigned char    protocol;      //[2]  
    unsigned char    com_mode;      //[3]  
    unsigned long    baudrate;      //[4]  
    unsigned char    parity;        //[5]  
    unsigned char    num_stop;      //[6]  
    unsigned short   time_interval; //[7]  
};
```

где: **status** – статус порта (0 – отключен, 1 – внешний, 2 – локальный);  
**type\_interface** – тип интерфейса (0 – RS232, 1 – RS485, 2 – RS232 с эхом);  
**protocol** – тип протокола (0 – протокол «Гран-Система-С», 1 – протокол «M-BUS»);  
**com\_mode** – режим работы (0 – стандартный, 1 – прозрачный);  
**baudrate** – скорость обмена (1200...115200);  
**parity** – бит паритета (0 – отсутствует, 1 – нечетность, 2 – четность);  
**num\_stop** – количество стоп-бит (1 или 2);  
**time\_interval** – тайм-аут между байтами в мс (0 – расчет по умолчанию).

Параметр **status** определяет роль последовательного порта в системе. Если порт не используется, то параметр **status** должен равняться 0. Если порт предназначен для связи с

устройствами верхнего уровня, то его статус должен быть установлен в 1. Если же порт предназначен для подключения подчиненных узлов, то статус порта должен быть локальным. Порт COM1 всегда имеет статус «внешний», порт COM2 может иметь статус «внешний» или «отключен», остальным портам может быть присвоен статус «локальный» или «отключен», однако надо иметь в виду, что при включении режима автоконфигурирования (см. раздел «Управление режимами работы ССДУ») поиск узлов будет выполняться только для портов имеющих статус «локальный».

Параметр *type\_interface* позволяет выбрать требуемый тип интерфейса. Тип «RS232 с эхом» может быть установлен в том случае, если к порту подключен преобразователь интерфейса, генерирующий эхо-сигнал (некоторые типы преобразователей RS232 / MBUS).

Параметр *protocol* определяет тип протокола, используемый для данного порта. Внешние порты должны иметь тип протокола «Гран-Система-С». Для локальных портов тип протокола должен совпадать с типом протокола подключенных устройств.

Значение параметра *com\_mode* имеет смысл только для тех портов, у которых параметр *status* установлен в 1 (внешний). Значение этого параметра изменяет способ адресации ССДУ. Если значение *com\_mode* равно 0, то ССДУ реагирует только на 0, свой собственный адрес и 255 (см. раздел «Особенности протокола»).

Если значение *com\_mode* равно 1, то в ССДУ изменяется способ адресации. В этом режиме ССДУ «забывает» свой собственный адрес и реагирует на адрес 0 как на свой собственный, отсутствует широковещательный обмен с адресами 0 и 255, и допустим обмен только точка-точка, т.е. на одной шине может присутствовать только одно ССДУ. Если в этом режиме ССДУ получает пакет с адресом отличным от 0, то адрес интерпретируется как индекс в массиве описаний конфигурации узлов (см. раздел «Конфигурация узлов») и по этому индексу извлекается конфигурация узла. Если тип узла не «пустой», то поле «адрес» принятого пакета заменяется сетевым адресом узла, пересчитывается CRC и пакет отправляется в порт, к которому подключен узел. При приеме ответного пакета от узла восстанавливается адрес пакета, пересчитывается CRC и пакет возвращается отправителю. Основным достоинством этого режима является то, что возможен абсолютно «прозрачный» обмен с узлами. Прозрачная адресация возможна только для узлов с протоколом «Гран-Система-С».

Параметры *baudrate*, *parity* и *num\_stop* позволяют установить требуемые параметры обмена по последовательному порту. Диапазон скоростей для всех 4-х портов составляет 1200...115200 бод.

Параметр *time\_interval* позволяет задать значение таймаута ожидания ответа от устройства (только для узлов с протоколом «Гран-Система-С»). Если значение этого параметра равно 0, то ССДУ рассчитывает значение таймаута автоматически, исходя из скорости обмена, вида паритета и количества стоповых бит.

### 3.4.1 Чтение параметров последовательных портов

За одну транзакцию можно прочитать любой набор параметров для одного или нескольких портов. Формат запроса и ответа приведен на рисунке 20 и рисунке 21 соответственно.

HDR	M{порты} (0...3)	M{параметры} (0...7)	CRC
-----	------------------	----------------------	-----

Рисунок 20

HDR	A[порты][параметры]	CRC
-----	---------------------	-----

Рисунок 21

Например, если для ССДУ с сетевым адресом 1 необходимо узнать тип интерфейса и скорость обмена для порта COM2, то необходимо сформировать запрос как на рисунке 22.

заголовок				M{порты}		M{параметры}			CRC	
1	1	5	0	0	1	6	18	0	241	126

Рисунок 22

Если у порта COM2 тип интерфейса 1 (RS485) и скорость обмена 19200, то ССДУ возвратит ответ как на рисунке 23.

заголовок				тип	скорость обмена			CRC		
1	1	5	0	1	0	75	0	0	122	222

Рисунок 23

### 3.4.2 Запись параметров последовательных портов

Изменить параметры последовательных портов можно только в режиме конфигурирования. Аналогично, как и при чтении, необходимо указать перечень узлов и набор параметров и сформировать соответствующий двумерный массив значений параметров как на рисунке 24.

HDR	M{порты }(0...3)	M{параметры} (0...7)	A[порты][параметры]	CRC
-----	------------------	----------------------	---------------------	-----

Рисунок 24

## 3.5 Конфигурация узлов

Номер параметра: 6

Допустимые операции: чтение, подсчет размера ответа, запись

Каждое устройство (узел), которое обслуживается ССДУ должно быть зарегистрировано в базе данных конфигурации (БДК). Размер БДК позволяет хранить до 400 записей. Индекс записи в БДК одновременно является и логическим адресом устройства, причем запись с индексом 0 всегда хранит конфигурацию самого ССДУ.

Конфигурация узлов описывается следующей структурой:

```

struct TNodeConfig
{
    unsigned char    node_type;           //[0]
    unsigned char    node_status         //[1]
    unsigned char    n_port;             //[2]
    unsigned char    tariff_idx;         //[3]
    unsigned short   net_adr;            //[4]
    unsigned short   group_mask;         //[5]
    unsigned char    id[8];              //[6]
    unsigned char    type_device[16];    //[7]
    unsigned char    serial_num[10];     //[8]
    TDateAndTime     produce_date;       //[9]
    TDateAndTime     check_date;         //[10]
    unsigned short   soft_ver;           //[11]
    unsigned char    parol[8];           //[12]
};
    
```

где: **node\_type** – тип узла (0 – пустой, 1 – ССДУ, 2 – М-Bus устройство);  
**node\_status** – статус узла (0 – подключен, 1 – отключен);  
**n\_port** – порт, к которому подключен узел;  
**tariff\_idx** – индекс тарифного расписания (для устройств М-Bus всегда 0);  
**net\_adr** – сетевой адрес узла (для устройств М-Bus всегда 0);  
**group\_mask** – маска группы;  
**id** – идентификатор пользователя (производитель устройства для М-Bus устройств);  
**type\_device** – строковое обозначение типа устройства;  
**serial\_num** – строковое обозначение заводского номера устройства;  
**produce\_date** – дата изготовления (для устройств М-Bus заполняется нулями);  
**check\_date** – дата очередной поверки (для устройств М-Bus заполняется нулями);  
**soft\_ver** – версия программного обеспечения;  
**parol** – пароль.

Параметр **node\_type** определяет тип устройства.

Параметр **node\_status** позволяет исключить из опроса соответствующее ему устройство.

Параметр **n\_port** содержит номер порта, к которому подключен узел. Нумерация портов начинается с 0, т.е. СОМ1 имеет индекс 0, СОМ4 – 3.

Параметр **tariff\_idx** хранит индекс тарифного расписания, допустимые значения 0...3.

Параметр **net\_adr** содержит сетевой адрес устройства. Данный параметр имеет смысл только для устройств с протоколом «Гран-Система-С» Необходимо иметь в виду, что размерность этого параметра составляет два байта. Сетевые адреса устройств должны быть уникальны только в пределах одного порта.

Параметр **group\_mask** позволяет объединять отдельные устройства в логические группы. Такое объединение позволяет выбирать различные группы устройств при параметризации расписания. ССДУ позволяет создать 16 групп, которые кодируются одним битом 16-ти разрядного параметра **group\_mask**. Принадлежность устройства к той или другой группе задается установкой соответствующего бита в параметре **group\_mask**, причем одно и то же устройство может быть включено одновременно в несколько групп. Если значение **group\_mask** равно 0 либо 0xFFFF, то это означает, что узел принадлежит всем группам.

Параметр **id** имеет размерность 8 байт. Для устройств с протоколом «Гран-Система-С» это поле содержит произвольный набор данных, который может задать пользователь. Для устройств с протоколом «М-Bus» в данном поле передается символьное обозначение производителя устройства.

Параметры **type\_device** и **serial\_num** – строковое представление типа и заводского номера устройства, размерность каждого из них составляет соответственно 16 и 10 байт. При автоконфигурировании эти параметры заполняются значениями, считанными с соответствующего устройства.

Параметры **produce\_date** и **check\_date** содержат дату и время изготовления устройства, а также дату и время очередной поверки устройства соответственно. Данные параметры имеют смысл только для устройств с протоколом «Гран-Система-С»; для устройств с протоколом «М-Bus» данные параметры заполняются нулями. Параметр **produce\_date** заполняется при автоконфигурировании, а параметр **check\_date** можно



заполнить вручную датой очередной поверки электросчетчика. Формат этих параметров представлен в разделе «Типы данных».

Параметр *soft\_ver* содержит версию программного обеспечения (ПО) узла в двоичном двухбайтном формате. Для устройств с протоколом «Гран-Система-С» старший байт содержит основную версию (major), а младший байт дополнительную (minor). Например, если версия ПО имеет номер 1.18 то параметр *soft\_ver* в шестнадцатеричном виде будет иметь вид 112h. Для устройств с протоколом «M-Bus» старший байт всегда равен нулю, а младший содержит версию устройства. При автоконфигурировании этот параметр заполняется автоматически.

Параметр *parol* содержит пароль соответствующего узла для узлов с протоколом «Гран-Система-С». При автоконфигурировании параметр *parol* заполняется восемью ASCII-символами «0» (см. раздел «Изменение паролей»). Для обеспечения конфиденциальности при чтении значение этого параметра всегда равно восьми символам «\*».

### 3.5.1 Чтение конфигурации узлов

Формат запроса и ответа для чтения конфигурации узлов приведены на рисунке 25 и рисунке 26 соответственно.

HDR	M{ узлы } (0...254)	M{параметры}(0...12)	CRC
-----	---------------------	----------------------	-----

Рисунок 25

HDR	A[узлы][параметры]	CRC
-----	--------------------	-----

Рисунок 26

### 3.5.2 Запись конфигурации узлов

Изменение конфигурации узлов возможно только в режиме конфигурирования. Формат запроса представлен на рисунке 27.

HDR	M{ узлы } (0...254)	M{параметры}(0...12)	A[узлы][параметры]	CRC
-----	---------------------	----------------------	--------------------	-----

Рисунок 27

## 3.6 Дата и время ССДУ

Номер параметра: 8

Допустимые операции: чтение, запись

ССДУ в своем составе имеет энергонезависимые часы реального времени, однако погрешность их хода более 2-х секунд в сутки. Для обеспечения требований, предъявляемых к современным системам АСКУЭ, требуется периодическая синхронизация часов верхним уровнем.

В дополнение к синхронизации часов извне по каналу связи можно заставить ССДУ синхронизировать свои часы по серверу точного времени в Интернете.

### 3.6.1 Чтение текущей даты и времени

При чтении возвращается текущая дата и время в формате *TDateAndTime* (см. раздел «Типы данных»). Формат запроса приведен на рисунке 28, а ответа на рисунке 29.

HDR	CRC
-----	-----

Рисунок 28

HDR	дата и время (6 байт)	CRC
-----	-----------------------	-----

Рисунок 29

### 3.6.2 Запись текущей даты и времени

В протоколе предусмотрено два варианта команды записи текущей даты и времени. Первый вариант предназначен для синхронизации времени, а второй для установки времени. Синхронизация подразумевает изменение времени не более чем на 30 минут и при этом проверяется, не исчерпан ли лимит корректировок, составляющий 30 минут в месяц. Для этого при каждой синхронизации вычисляется разность между новым и текущим временем, и абсолютное значение разницы накапливается в регистре корректировок. Если при очередной синхронизации лимит будет исчерпан, то команда не будет выполнена и будет возвращен код ошибки 4 («Несанкционированный доступ»). Формат запроса для синхронизации времени приведен на рисунке 30.

HDR	дата и время (6 байт)	CRC
-----	-----------------------	-----

Рисунок 30

Установка времени позволяет установить любое корректное значение даты и времени, и при этом будет обнулен регистр корректировок. Для выполнения установки требуется коммерческий пароль. Формат запроса для установки времени приведен на рисунке 31.

HDR	коммерческий пароль (8 байт)	дата и время (6 байт)	CRC
-----	------------------------------	-----------------------	-----

Рисунок 31

## 3.7 Календарь выходных дней

Номер параметра: 10

Допустимые операции: чтение, запись, обнуление, расчет размера ответа

ССДУ автоматически рассчитывает календарь выходных дней с учетом стандартных белорусских праздников на любой год любого месяца. Однако имеется возможность составить свой календарь на любой месяц любого года. Для этих целей ССДУ содержит массив из 12 объектов типа *TWeekend*.

```
struct TWeekend
{
    unsigned long    days;
    unsigned char   year;
    bool            valid;
};
```

где: *days* – дни месяца;  
*year* – год;  
*valid* – признак достоверности структуры.

Биты 32-битного параметра *days* представляют собой дни месяца. Бит 0 соответствует 1 дню, бит 1 – 2 дню и т.д. Выходные дни кодируются логической ‘1’, а рабочие дни – логическим ‘0’.

Параметр *year* уточняет, для какого года составлен календарь (параметр *days*). Отсчет ведется от 2000 года.

Параметр *valid* подтверждает достоверность структуры. Если его значение равно *true*, то данные, содержащиеся в структуре *TWeekend*, достоверны, а если *false* – то нет.

Каждая из структур предназначена для одного месяца. Когда возникает необходимость получить календарь выходных дней для конкретного месяца и года, то ССДУ по номеру месяца находит соответствующую структуру и если ее параметр *valid* равен *true* и параметр *year* соответствует требуемому году, то возвращается параметр *days*, в противном случае выполняется автоматический расчет календаря.

ССДУ автоматически обновляет календарь выходных дней для текущего и следующего месяца всех подключенных к нему устройств с протоколом «Гран-Система-С», поэтому в случае возникновения необходимости изменить календарь для всех устройств, достаточно провести эту операцию только для ССДУ самого верхнего уровня иерархии.

### 3.7.1 Чтение календаря выходных дней

Протокол позволяет за одну транзакцию прочитать календари выходных дней для одного или нескольких месяцев. Формат запроса приведен на рисунке 32.

HDR	M{год}(0...254)	M{месяц}(1...12)	CRC
-----	-----------------	------------------	-----

Рисунок 32

В ответном пакете ССДУ возвратит массив четырехбайтных значений *days*. Размер массива будет равняться произведению количества элементов множеств «год» и «месяц».

### 3.7.2 Запись календаря выходных дней

Изменение календаря выходных дней является привилегированной операцией, поэтому для ее выполнения требуется коммерческий пароль. Формат запроса для записи нового календаря приведен на рисунке 33.

HDR	ком. пароль	M{год}(0...254)	M{месяц}(1...12)	A[год][месяц]	CRC
-----	-------------	-----------------	------------------	---------------	-----

Рисунок 33

Поле запроса «A[год][месяц]» должно содержать массив четырехбайтных значений *days*, соответствующих множествам «год» и «месяц». После приема корректного запроса для каждого календаря ССДУ выполняет следующие действия:

- создает объект типа *TWeekend*;
- копирует в параметр *days* очередное значение из поля «A[год][месяц]»;
- копирует в параметр *year* очередное значение из множества «год»;
- присваивает параметру *valid* значение *true*;
- помещает созданный объект в свой собственный массив по индексу, соответствующему месяцу из множества «месяц».

### 3.7.3 Обнуление календаря выходных дней

Так же как и для записи, для операции обнуления требуется коммерческий пароль. Название операции не совсем точно отражает выполняемые действия, на самом деле после выполнения этой операции, для тех месяцев, которые указаны в запросе, ССДУ возобновляет автоматический расчет календаря выходных дней. Эта операция может быть полезна, если по каким-то причинам для каких-то месяцев был записан календарь, а потом возникла необходимость вернуться к автоматическому расчету. Формат запроса для восстановления автоматического расчета календаря приведен на рисунке 34.

HDR	ком. пароль	M{год}(0..254)	M{месяц}(1..12)	CRC
-----	-------------	----------------	-----------------	-----

Рисунок 34

После выполнения запроса для всех объектов структуры *TWeekend*, соответствующих множеству «месяц», параметр *valid* будет установлен в значение *false*.

### 3.8 Дата и время переключения сезонов

Номер параметра: 11

Допустимые операции: чтение, запись, обнуление, расчет размера ответа

По умолчанию, ССДУ автоматически переходит на зимний сезон в 3 часа ночи последнего воскресенья октября и на летний сезон в 2 часа ночи последнего воскресенья марта. Переход на зимний сезон сопровождается вычитанием одного часа из текущего времени, а на летний сезон – добавлением одного часа. Для реализации такого поведения используются два экземпляра структуры типа *TDateAndTime*. Один из них хранит дату и время перехода на зимний сезон, а второй – дату и время перехода на летний сезон.

Перерасчет даты и времени переключения сезонов выполняется в следующих случаях:

- при установке времени (раздел «Дата и время ССДУ»);
- при включении питания;
- при наступлении момента переключения сезона.

#### 3.8.1 Чтение дат и времени переключения сезонов

Для контроля даты и времени переключения сезонов можно воспользоваться операцией чтения, которая имеет следующий формат:

HDR	M{сезоны}(0,1)	CRC
-----	----------------	-----

Рисунок 35

Множество с именем «сезоны» задает перечисление сезонов (0 – зимний сезон, 1 – летний). В ответном пакете будет возвращен массив структур типа *TDateAndTime*. Размерность массива зависит от количества считываемых сезонов.

#### 3.8.2 Запись дат и времени переключения сезонов

Как уже упоминалось выше, расчет начала сезонов выполняется автоматически и не требует вмешательства извне. Однако может возникнуть ситуация, когда требуется изменить начало сезона или отключить их отслеживание. Для этого по каналу связи можно изменить дату и время начала того или другого сезона. При этом автоматический расчет будет отключен, а переход на соответствующий сезон будет выполнен при

совпадении текущей даты-времени и записанного значения. Если значение начала сезона будет содержать дату и время меньшую текущей, то переход на соответствующий сезон никогда не произойдет.

Изменение даты и времени начала сезона является привилегированной операцией, поэтому для ее выполнения требуется коммерческий пароль. Формат запроса для записи даты и времени начала сезонов приведен на рисунке 36.

HDR	ком. пароль	M{сезоны}(0,1)	A[сезоны]	CRC
-----	-------------	----------------	-----------	-----

Рисунок 36

Кодировка множества с именем «сезоны» такая же как и для операции чтения. Новые значения должны быть типа *TDateAndTime*.

### 3.8.3 Обнуление даты и времени начала сезонов

Если значение начала сезона было изменено операцией записи, то автоматический расчет для этого сезона отключается. Для того чтобы восстановить автоматический расчет, необходимо выполнить операцию обнуления для обоих сезонов. Формат запроса на выполнение операции обнуления приведен на рисунке 37.

HDR	ком. пароль	M{сезоны}(0,1)	CRC
-----	-------------	----------------	-----

Рисунок 37

## 3.9 Архивы ССДУ

Номер параметра: 12

Допустимые операции: чтение, расчет размера ответа

ССДУ имеет в своем составе 2 архива: архив состояний обмена и архив корректировок. Каждый из архивов может хранить до 100 записей. Если архив полностью заполнен и возникла необходимость помещения в архив ещё одной записи, то самая старая запись будет удалена. Каждая запись в архиве описывается следующей структурой:

```
struct TArhivRecord
{
    TDateAndTime    dt;
    unsigned short  reason;
    unsigned char   define;
    unsigned char   dummy;
};
```

где: **dt** – дата и время фиксации события с точностью до секунды;  
**reason** – код текущего события;  
**define** – уточнение события;  
**dummy** – пустой байт (дополнение до четного размера структуры в байтах).

### 3.9.1 Архив состояний обмена

Архив состояния обмена зарезервирован для будущих расширений ССДУ и на данный момент (версия 1.0) не используется.

### 3.9.2 Архив корректировок

Архив корректировок предназначен для отслеживания хронологии внешних воздействий на ССДУ. Кодировка воздействий приведена в таблице 1.

Таблица 1

№ бита	Название
0	Сетевое питание включено
1	Сетевое питание отключено
2	Проведена установка времени с паролем (параметр 8)
3	Зарезервировано
4	Изменена конфигурация портов (параметр 5)
5	Изменена конфигурация узлов (параметр 6)
6	Изменено тарифное расписание (параметр 9)
7	Изменен календарь выходных дней (параметр 10)
8	Изменена дата и время начала сезона (параметр 11)
9	Изменен пароль
10	Зафиксирована попытка сканирования пароля

При возникновении любого из вышеперечисленных событий выполняются следующие действия:

- создается экземпляр структуры *TArhivRecord*;
- поле *dt* заполняется текущей датой и временем;
- в поле *reason* устанавливается соответствующий бит (№ бита в таблице 2).
- в поле *define* записывается 0;
- экземпляр *TArhivRecord* помещается в архив.

### 3.9.3 Чтение архивов

Формат запроса для чтения архивов представлен на рисунке 38.

HDR	M{тип архива}(0,1)	M{смещение}(0,-1...-100)	CRC
-----	--------------------	--------------------------	-----

Рисунок 38

Как видно из рисунка, в запросе обязательно должны присутствовать два множества. Множество с именем «тип архива» определяет перечисление типов архивов. Значение 0 присвоено архиву состояния обмена, а 1 – архиву корректировок. Множество с именем «смещение» позволяет выбрать требуемое количество записей. Каждый член этого множества задает относительное положение записи по отношению к текущему времени. Смещение с кодом -1 указывает на последнюю запись в архиве, -2 – на предпоследнюю и т.д. Смещение с кодом 0 определяет текущее состояние события.

Например, для архива состояний обмена поля записи со смещением 0 будет содержать:

- *dt* – текущие дату и время;
- *reason* – состояние обмена всех опрашиваемых узлов по ИЛИ;
- *define* – 0.

Запись с нулевым смещением в архиве событий будет содержать:

- *dt* – текущие дату и время;
- *reason* – состояние 0;
- *define* – 0.

Если запись по указанному в запросе смещению отсутствует или она недостоверна, то все члены записи (структура *TArhivRecord*) будут иметь нулевое значение, в том числе все члены структуры *TDateAndTime* будут равны 0.

Пример запроса для ССДУ с сетевым адресом 1 при использовании расширенного заголовка (номер пакета 7) для чтения всех записей архива состояний обмена приведен на рисунке 39.

расширенный заголовок								тип арх.		смещения			CRC	
1	2	15	0	7	0	12	0	0	0	4	255	100	147	51

Рисунок 39

### **3.10 Архивные данные по узлам за период времени**

Номер параметра: 13

Допустимые операции: чтение, подсчет размера ответа, чтение в текстовом потоке

ССДУ в автоматическом режиме считывает показания с подключенных к нему устройств. Считывание происходит в моменты времени, заданные при конфигурировании расписания ССДУ. Глубина архива зависит от частоты считываний, заданной в расписании системы, и от количества узлов, с которых считываются показания. Всего в архиве ССДУ может храниться до 1785 записей, с максимальной длиной, предусмотренной спецификациями М-Bus.

Для считывания архивных записей в протоколе ССДУ предусмотрена команда чтения архивных данных. Считывать архивные записи можно как в бинарном виде, так и в текстовом.

Операция «подсчет размера ответа» выдает размер ответа для чтения в бинарном виде. Размер ответа для чтения в текстовом виде не ограничен.

При считывании данных в бинарном виде архивные записи выдаются в виде стандартных пакетов М-Bus в том виде, как они были считаны с приборов учета. Для расшифровки и декодирования пакетов необходимо руководствоваться спецификациями М-Bus «<http://www.m-bus.com/>».

При считывании данных в текстовом виде ССДУ сама производит расшифровку архивных записей и выдает данные в виде текстового потока с определенными правилами форматирования. Данный способ является более предпочтительным, т.к. позволяет более просто произвести разбор полученных данных; кроме того, при передаче в текстовом виде не накладывается ограничение на длину пакета (1024 байта).

Формат запроса на чтение архивных данных приведен на рисунке 40.

HDR	TimeStart	TimeEnd	M{узлы}(1...399)	CRC
-----	-----------	---------	------------------	-----

Рисунок 40

TimeStart – структура типа *TDateAndTime*, задающая нижнюю границу времени создания архива;

TimeEnd – структура типа *TDateAndTime*, задающая верхнюю границу времени создания архива;

M{узлы} – множество узлов.

Формат ответа зависит от операции.

Если код операции – подсчет размера ответа, то в ответном сообщении будет возвращена ожидаемая длина ответа при чтении в бинарном виде.

Если код операции – чтение, то в ответном сообщении будут помещены архивные записи по узлам, заданных множеством М, время создания которых лежит между TimeStart и TimeEnd. Архивные записи будут представлены в протоколе, определенном спецификациями М-Bus.

Если код операции – чтение в текстовом виде, то ССДУ выполнит следующие действия:

- Порт, на который пришел запрос, будет переведен в текстовый режим; при этом в порт будет выведена строка «"**<< Text mode begin\r\n**"».
- В порт будут выведены архивные данные по запрашиваемым узлам за заданный период времени. Данные будут выведены в текстовом виде, в формате оговоренном в приложении Д.
- Порт будет выведен из текстового режима, при этом в порт будет выведена строка «"**Text mode end >>\r\n**"».

При чтении данных в текстовом виде следует иметь в виду:

1. Если чтение происходит с опто-порта, то действует временной лимит на длительность чтения равный 5 минутам. По истечении этого времени ССДУ переключит порт в режим платы расширения.
2. Если чтение происходит со встроенного GSM-модуля, и по расписанию ССДУ настало время передачи данных по E-Mail, FTP, либо синхронизации времени по серверу NTP, то чтение будет прервано.

### **3.11 Конфигурация расписания**

Номер параметра: 14

Допустимые операции: чтение, запись, обнуление, расчет размера ответа

С помощью расписания можно задать время, в которое ССДУ будет выполнять те или иные задания.

Каждому заданию соответствует бит в 32-разрядной битовой маске.

Перечень обрабатываемых ССДУ заданий представлен в следующей таблице:

Таблица 2

<b>Маска задания</b>	<b>Название задания</b>
0x0001	Синхронизация времени
0x0002	Считать данные с М-Bus устройств в архив
0x0004	Считать данные с М-Bus устройств в архив
0x0008	Считать данные с М-Bus устройств в архив
0x0010	Считать данные с М-Bus устройств в архив
0x0100	Считать данные с М-Bus устройств и передать их в E-Mail сообщении
0x0200	Считать данные с М-Bus устройств и передать их в E-Mail сообщении
0x0400	Передать архив в E-Mail сообщении
0x0800	Передать архив в E-Mail сообщении
0x1000	Передать архив на FTP сервер
0x2000	Передать архив на FTP сервер



Каждое задание описывается следующей структурой:

```
struct TTaskCfg
{
    TCron    cron;        // расписание запуска
    uchar    param[4];    // дополнительные параметры
};
```

Поле cron описывает расписание запуска данного задания.

```
struct TCron
{
    uchar    sec;        // секунды
    uchar    min;        // минуты
    uchar    hour;       // часы
    uchar    day;        // день
    uchar    month;      // месяц
    uchar    year;       // год
};
```

Данное поле используется для сравнения с текущим временем ССДУ. Если результат сравнения положительный, то происходит запуск соответствующего задания.

Сравнение структур TCron и TDateAndTime происходит поэлементно, т.е. TCron.sec сравнивается с TDateAndTime.sec, TCron.min сравнивается с TDateAndTime.min и так далее. Если хотя бы одно поэлементное сравнение отрицательно, то результат сравнения отрицательный, если же все поэлементные сравнения положительны, то результат сравнения положительный. При поэлементном сравнении действуют следующие правила:

- Если элемент в структуре TCron равен 0xFF, то он считается равным любому значению соответствующего элемента структуры TDateAndTime;
- Если старший бит элемента в структуре TCron равен нулю, то этот элемент проверяется на равенство соответствующему элементу структуры TDateAndTime;
- Если старший бит элемента в структуре TCron равен единице, то выполняется проверка, что соответствующий элемент структуры TDateAndTime делится без остатка на число, закодированное в младших 7 битах элемента структуры TCron.

Пример функции сравнения в стандарте языка «C»:

```
bool CompareCron(TDateAndTime *tm, TCron *crn)
{
    uchar *bufTm = (uchar*)tm;
    uchar *bufCron = (uchar*)crn;
    for(int i = 0; i < sizeof(TDateAndTime); i++)
        if(bufCron[i] != 0xFF)
            if((bufCron[i] & 0x80) == 0)
                if(bufCron[i] != bufTm[i]) return false;
            else
                if((bufTm[i] % (bufCron[i] & 0x7F)) != 0) return false;
```

```
    return true;  
}
```

В структуре TTaskCfg присутствует поле «*param*». С помощью данного поля можно задать дополнительные параметры для каждого задания. Кодировка поля зависит от типа задания и приведена ниже.

Для заданий типа «Синхронизация времени»: (uchar)param[0] содержит минимальное допустимое рассогласование времени, (ushort)(param[1]+param[2]\*0x100) содержит максимальное недопустимое рассогласование времени, (char)param[3] содержит часовой пояс.

Для заданий типа «Считать данные с M-Bus устройств в архив», «Считать данные с M-Bus устройств и передать их в E-Mail сообщении», «Передать архив в E-Mail сообщении», «Передать архив на FTP сервер»: (ushort)(param[0]+param[1]\*0x100) содержит маску группы узлов, к которым будет применяться данное задание.

### 3.11.1 Чтение конфигурации расписания

Для чтения конфигурации расписания используется запрос, представленный на рисунке 41.

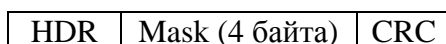


Рисунок 41

Поле Mask содержит побитно маски всех расписаний, которые нужно прочитать. Ответ от ССДУ будет иметь вид, представленный на рисунке 42.

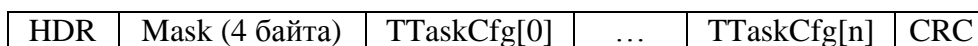


Рисунок 42

Поле Mask содержит побитно маски всех расписаний, которые присутствуют в ответе. Далее следуют сами расписания, начиная с младшей маски.

### 3.11.2 Запись конфигурации расписания

Для записи конфигурации расписания используется пакет следующего вида:

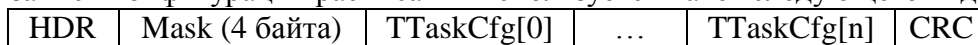


Рисунок 43

Поле Mask содержит побитно маски всех расписаний, которые необходимо записать в ССДУ. Далее следуют сами расписания, начиная с младшей маски.

### 3.11.3 Очистка конфигурации расписания

Для очистки конфигурации расписания используется запрос, представленный на рисунке 44.

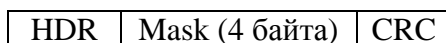


Рисунок 44

Поле Mask содержит побитно маски всех расписаний, которые нужно очистить.

### 3.12 Текущие данные по множеству узлов

*Номер параметра:* 15

*Допустимые операции:* чтение в текстовом виде

В ССДУ есть возможность считать данные с узлов по запросу пользователя. Формат запроса представлен на рисунке 45.

HDR	M{узлы}(1...399)	CRC
-----	------------------	-----

Рисунок 45

Получив данный запрос, ССДУ начнет по очереди считывать данные с узлов, заданных множеством «узлы», и передавать полученные данные в текстовом потоке.

Текстовый поток формируется по правилам, описанным в пункте «3.10 Архивные данные по узлам за период времени».

### 3.13 Конфигурация встроенного GSM-модуля

#### 3.13.1 Параметры GSM-модуля

*Номер параметра:* 19

*Допустимые операции:* чтение, запись

Данный параметр предназначен для конфигурации параметров встроенного GSM-модуля.

Конфигурация GSM-модуля описывается следующей структурой:

```
struct TConfGSM
{
    unsigned char    ApnLoginPassword[101];    //[0]
    unsigned char    SMTP_Server[51];         //[1]
    unsigned char    Login[51];               //[2]
    unsigned char    Password[31];            //[3]
    unsigned char    Sender[61];              //[4]
    unsigned char    Recepient1[61];          //[5]
    unsigned char    Recepient2[61];          //[6]
    unsigned char    Recepient3[61];          //[7]
    unsigned char    NtpSrv[51];              //[8]
    unsigned short   TCP_Port;                 //[9]
    unsigned char    FTPSrv[51];              //[10]
    unsigned char    FTPLogin[51];            //[11]
    unsigned char    FTPPass[31];             //[12]
    unsigned char    FTPDir[51];              //[13]
};
```

Поля данной структуры заполняются следующим образом:

- **ApnLoginPassword** – точка доступа [0 – длина строки, 1..101 – строка], формат строки: “Apn”, “Login”, “Pass”;
- **SMTP\_Server** – сервер исходящей почты [0 – длина строки, 1..51 – строка], пример формата строки: “smtp.inbox.ru”, 25;

- **Login** – логин почтового ящика [0 – длина строки, 1..51 – строка];
- **Password** – пароль почтового ящика [0 – длина строки, 1..31 – строка];
- **Sender** – отправитель [0 – длина строки, 1..61 – строка];
- **Recepient1** – получатель 1 [0 – длина строки, 1..61 – строка];
- **Recepient2** – получатель 2 [0 – длина строки, 1..61 – строка];
- **Recepient3** – получатель 3 [0 – длина строки, 1..61 – строка];
- **NtpSrv** – сервер точного времени [0 – длина строки, 1..51 – строка], пример формата строки: “www.belgim.by”,123;
- **TCP\_Port** – порт для входящего TCP-соединения. Нулевое значение отключает режим TCP-сервер;
- **FTPSrv** – сервер FTP [0 – длина строки, 1..51 – строка], пример формата строки: “ftp.drivehq.com”,21;
- **FTPLogin** – логин FTP-сервера [0 – длина строки, 1..51 – строка];
- **FTPPass** – пароль FTP-сервера [0 – длина строки, 1..31 – строка];
- **FTPSrvDir** – каталог FTP [0 – длина строки, 1..51 – строка].

Для чтения параметров используется запрос следующего формата:

HDR	M{параметры}(0...13)	CRC
-----	----------------------	-----

**Рисунок 46**

Для записи параметров используется запрос следующего формата:

HDR	M{параметры}(0...13)	Значения параметров	CRC
-----	----------------------	---------------------	-----

**Рисунок 47**

### **3.14 Другие команды**

Кроме вышеперечисленных команд ССДУ поддерживает запросы, предназначенные для электросчетчиков СС101 и СС301. Для ознакомления с их применением необходимо руководствоваться документами [1] и [2]. Перечень параметров, обрабатываемых ССДУ, приведен в приложении Г.

**Приложение А**  
(справочное)  
**Циклический избыточный код (CRC)**

Контроль циклическим кодом применяется для повышения надежности передачи данных. Смысл контроля заключается в следующем. Запрос подвергается шифровке циклическим кодом. Полученный результат добавляется в конец запроса, и весь пакет отправляется подчиненному устройству. Подчиненное устройство выполняет те же действия над байтами запроса и сравнивает полученный результат с CRC принятого пакета, и, в случае положительного результата, выполняет требуемое действие. Затем оно формирует ответное сообщение, подвергает его той же процедуре шифровки, добавляет 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, 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,
```

**СИСТЕМА СБОРА ДАННЫХ, ДИСПЕТЧЕРИЗАЦИИ И УПРАВЛЕНИЯ «ГРАН-ЭЛЕКТРО ССДУ»**  
**ИНСТРУКЦИЯ ОПЕРАТОРА ПО РАБОТЕ С ПОСЛЕДОВАТЕЛЬНЫМ КАНАЛОМ СВЯЗИ**  
**СИФП 46.00.000 И1**

---

```
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,  
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,  
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,  
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,  
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,  
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,  
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,  
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,  
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,  
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,  
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,  
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,  
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,  
0x40  
};
```

```
unsigned short __fastcall CRC16(unsigned char *msg, unsigned short len)  
{  
    unsigned short idx;  
    unsigned char CRChi = 0xFF;  
    unsigned char CRClo = 0xFF;  
  
    while(len--)  
    {  
        idx = (CRChi ^ *msg++) & 0xFF;  
        CRChi = CRClo ^ tblCRChi[idx];  
        CRClo = tblCRClo[idx];  
    }  
  
    return ((CRChi << 8) | CRClo);  
}
```

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

**Приложение Б**  
(обязательное)  
**Кодировка типов множеств**

Таблица Б.1

<b>Тип</b>	<b>Название</b>
0	Одноэлементное множество
1	Упорядоченное множество по возрастанию (первый, последний)
2	Упорядоченное множество по возрастанию (первый, количество)
3	Упорядоченное множество по убыванию (первый, последний)
4	Упорядоченное множество по убыванию (первый, количество)
5	Неупорядоченное множество
6	16-ти разрядная битовая маска
7	Двухбайтное одноэлементное множество
8	Двухбайтное упорядоченное множество по возрастанию (первый, последний)
9	Двухбайтное упорядоченное множество по возрастанию (первый, количество)
10	Двухбайтное упорядоченное множество по убыванию (первый, последний)
11	Двухбайтное упорядоченное множество по убыванию (первый, количество)

**Кодировка операций**

Таблица Б.2

<b>Операция</b>	<b>Название</b>
0	Чтение
1	Запись
2	Обнуление
3	Расчет размера ответа при чтении
4	Операция суммирования (СУМ)
5	Операция ИЛИ
6	Чтение в текстовом виде

**Кодировка типов параметров**

Таблица Б.3

<b>Параметр</b>	<b>Название</b>	<b>Допустимые операции</b>
2	Изменение паролей	запись
3	Управление режимами работы ССДУ	чтение, запись
5	Параметры последовательных портов	чтение, запись, размер ответа
6	Конфигурация узлов	чтение, запись, размер ответа
8	Дата и время	чтение, запись
10	Календарь выходных дней	чтение, запись, обнуление, размер ответа
11	Дата и время переключения сезонов	чтение, запись, обнуление, размер ответа
12	Архивы	чтение, размер ответа
13	Архивные данные по узлам за период времени	чтение, размер ответа, чтение в текстовом виде
14	Конфигурация расписания	чтение, запись, размер ответа, обнуление
15	Текущие данные по узлам за период времени	чтение в текстовом виде
19	Параметры GSM-модуля	чтение, запись



**Приложение В**  
(обязательное)  
**Кодировка поля 'результат' ответа**

Таблица В.1

<b>Код</b>	<b>Название</b>	<b>Описание</b>
0	ОК	Запрос выполнен успешно.
1	Неизвестная функция	Номер функции, указанный в поле « <i>функция</i> » запроса не поддерживается.
2	Неизвестный параметр	Номер параметра, указанный в поле « <i>параметр</i> » недоступен.
3	Ошибочный аргумент	Ошибка в поле данных запроса.
4	Несанкционированный доступ	Для выполнения этой функции требуется отключение защиты.
5	Блок поврежден	Некоторые параметры хранятся в виде блоков, защищенных контрольной суммой. Ошибка возникает при чтении блока, в котором обнаружено несовпадение контрольной суммы.
6	Ошибка памяти	Невозможно выполнение команды из-за неисправности памяти.
7	ССДУ занят	Такая ошибка может возникнуть, если переполнилась очередь запросов на запись в энергонезависимую память. При приеме такой ошибки необходимо повторить запрос.
8	Переполнение буфера	Размер ответа превышает 1024 байта.
9	Ошибочная операция	Поле « <i>операция</i> » содержит код недопустимый для данного параметра.
10	Недостаточно памяти	Такая ошибка может возникнуть при обработке параметра N6 с операцией «запись», если недостаточно памяти для размещения новых узлов.
100	Нет ответа	Только для транзитного обмена. Нет ответа от устройства, адресуемого вложенным запросом.
101	Ошибка CRC	Только для транзитного обмена. Принят ответ от устройства, адресуемого вложенным запросом, с ошибкой CRC.

**Приложение Г**  
 (справочное)

**Перечень параметров электросчетчиков СС101, СС301, обрабатываемых ССДУ**

Таблица Г.1

№	Название параметра	Смещение	Тариф	Уточнение	Модификация	Обнуление
0	Идентификационный номер устройства	0	0	0	-	-
14	Архив событий состояния фаз	0,-1...-100	0	0	-	-
15	Архив событий состояния прибора	0,-1...-100	0	0	-	-
16	Архив событий коррекций	0,-1...-100	0	0	-	-
17	Тип прибора	0	0	0	-	-
18	Заводской номер	0	0	0	-	-
19	Дата выпуска прибора	0	0	0	-	-
20	Версия программы	0	0	0	-	-
21	Сетевой адрес прибора	0	0	0	+	-
22	Идентификатор пользователя	0	0	0	+	-
23	Конфигурация порта связи	0	0	0	+	-
27	Дата и время перехода на летний сезон	0	0	0	+	+
28	Дата и время перехода на зимний сезон	0	0	0	+	+
29	Календарь выходных дней	0...255	0	1...12	+	-
30	Тарифное расписание для рабочих дней	0	0	1...12	+	-
31	Тарифное расписание для выходных дней	0	0	1...12	+	-
32	Текущее значение даты и времени	0	0	0	+	-
37	Пароль (только запись)	0	0	0,1	+	-

Примечание:

- В полях 'Смещение', 'Тариф' и 'Уточнение' приведены допустимые значения для каждого из параметров.
- Символ '+' в поле 'Модификация' означает, что этот параметр можно изменить.
- Символ '+' в поле 'Обнуление' означает, что этот параметр можно обнулить.

**Приложение Д**  
(справочное)

**Формат передачи данных в текстовом виде**

Передача данных в текстовом виде используется при чтении показаний с точек учета, подключенных к ССДУ (команды с параметрами 13 и 15), а так же при передаче E-Mail сообщений и при передаче данных на FTP.

При передаче данных в текстовом виде используются следующие соглашения:

- Данные передаются в виде строк;
- Каждая строка заканчивается символами 0x0D, 0x0A;
- Данные в одной строке разбиты на ячейки, которые отделены друг от друга символом «;» (точка с запятой);
- Количество ячеек постоянно для всех строк.

Для удобства далее будут приводиться примеры строк, заключенные в двойные кавычки без символов 0x0D, 0x0A.

Первая строка в потоке данных содержит название ячеек (столбцов) и имеет следующий вид:

**“N Node;TimePoint;Node ID;Man.;Type dev.;N Param;Name;Value;Unit;Type;Module;Storage; Tariff;CRC”**

где:

- **N Node** – номер узла в ССДУ;
- **TimePoint** – время считывания данных с точки учета;
- **Node ID** – идентификационный номер устройства (заводской номер);
- **Man.** – изготовитель устройства по спецификации M-Bus;
- **Type dev.** – тип устройства по спецификации M-Bus;
- **N Param** – порядковый номер параметра в пакете M-Bus (в пределах одного пакета);
- **Name** – название параметра по спецификации M-Bus;
- **Value** – значение параметра;
- **Unit** – единица измерения параметра;
- **Type** – тип параметра по спецификации M-Bus;
- **Module** – номер модуля по спецификации M-Bus;
- **Storage** – номер хранилища данных по спецификации M-Bus;
- **Tariff** – номер тарифа;
- **CRC** – контрольная сумма строки.

Далее в потоке данных передаются требуемые пакеты данных в текстовом виде. Правила передачи пакетов данных следующие:

- Каждый пакет данных содержит несколько параметров; каждый параметр передается в отдельной строке;
- Номер параметра в ячейке «N Param» соответствует номеру параметра в пакете данных;

- Ячейки «N Node», «TimePoint», «Node ID», «Man.» и «Type dev.» заполняются только для первого параметра в пакете, для последующих параметров пакета данные ячейки содержат пустую строку.

Пример передачи пакета данных от узла с номером 263 в текстовом виде:

```
263;11:0:42 30.10.2013;66308245;LUG;Heat outlet;1;Status;No Error;;;;;0x3AEA
;;;;;2;Actuality Duration;8;seconds;Instant.;0;0;0;0xAC67
;;;;;3;Averaging Duration;8;seconds;Instant.;0;0;0;0x822E
;;;;;4;Energy;33274000000;J;Instant.;0;0;0;0xF1F3
;;;;;5;Volume;1000.65;m^3;Instant.;0;0;0;0xDC78
;;;;;6;Power;300;W;Instant.;0;0;0;0x8F88
;;;;;7;Volume Flow;0.084;m^3/h;Instant.;0;0;0;0xAC04
;;;;;8;Flow Temperature;35;*C;Instant.;0;0;0;0xA157
;;;;;9;Return Temperature;32;*C;Instant.;0;0;0;0x2FA7
;;;;;10;Temperature Difference;2.7;K;Instant.;0;0;0;0x1F58
;;;;;11;Volume;796.57;m^3;Instant.;0;1;0;0x72CD
;;;;;12;Energy;24929000000;J;Instant.;0;1;0;0xD294
;;;;;13;Fabrication No;66308245;Instant.;0;0;0;0xCD8C
;;;;;14;On Time;25677;hours;Instant.;0;0;0;0x4D16
;;;;;15;On Time;5683;hours;Error;0;0;0;0xC843
;;;;;16;On Time;5683;hours;Error;0;1;0;0xBE92
;;;;;17;Time Point;1.1.0;Instant.;0;1;0;0xB743
;;;;;18;Energy;32655000000;J;Instant.;0;2;0;0x3368
;;;;;19;Energy;32655000000;J;Instant.;0;3;0;0x9F7F
;;;;;20;Energy;32655000000;J;Instant.;0;4;0;0x587C
;;;;;21;Energy;32655000000;J;Instant.;0;5;0;0xF46B
;;;;;22;Energy;32655000000;J;Instant.;0;6;0;0x0053
;;;;;23;Energy;32655000000;J;Instant.;0;7;0;0xAC44
;;;;;24;Energy;31790000000;J;Instant.;0;8;0;0x3752
;;;;;25;Energy;29436000000;J;Instant.;0;9;0;0x5749
;;;;;26;Energy;27701000000;J;Instant.;0;10;0;0xC4B2
;;;;;27;Energy;24929000000;J;Instant.;0;11;0;0xEE3C
;;;;;28;Energy;22635000000;J;Instant.;0;12;0;0x7D26
;;;;;29;Energy;21240000000;J;Instant.;0;13;0;0x854E
;;;;;30;Energy;20381000000;J;Instant.;0;14;0;0xD877
;;;;;31;Energy;20381000000;J;Instant.;0;15;0;0xDE1B
;;;;;32;Energy;20381000000;J;Instant.;0;16;0;0xD4AF
```

Теперь рассмотрим по порядку столбцы ячеек и значения, которые могут принимать данные в этих столбцах.

Столбец «**N Node**» содержит номер узла в ССДУ и может принимать значения от 1 до 399 (нулевой узел в БДК закреплен за самим ССДУ).

Столбец «**TimePoint**» содержит время, в которое был считан пакет данных. Формат представления времени «часы:минуты:секунды день:месяц:год».

Столбец «**Node ID**» содержит уникальный идентификационный номер устройства, который, как правило, совпадает с заводским номером.

Столбец «**Man.**» содержит трехсимвольное обозначение производителя устройства. Данное обозначение является уникальным идентификатором, закрепленным за данным производителем.

Столбец «**Type dev.**» содержит тип устройства (разделение идет по типу измеряемых ресурсов). Данное поле может принимать следующие значения:

- Other – устройство, не определенное в спецификации M-Bus;
- Oil – нефть;
- Electricity – электричество;
- Gas – газ;
- Heat outlet – отдаваемое тепло;
- Steam – пар;

- Hot Water – горячая вода;
- Water – вода;
- Heat Cost Alloc. – стоимость выделенного тепла;
- Compressed Air – сжатый воздух;
- Cool.l.meter:out – (Volume measured at return temperature: outlet) измеренный объем при температуре в обратной магистрали, выход;
- Cool.l.meter:in – (Volume measured at flow temperature: inlet) измеренный объем при температуре в потоке, вход;
- Heat inlet – принимаемое тепло;
- Heat/Cooling l.m – (Heat / Cooling load meter) измеритель нагрузки отопления/охлаждения;
- Bus/System – вспомогательное устройство M-Bus;
- Unknown Medium – неопределенная среда;
- Cold Water – холодная вода;
- Dual Water – двойная вода;
- Pressure – давление;
- A/D Converter – аналогово-цифровой преобразователь;
- Reserved – зарезервировано.

Столбец «**N Param**» содержит порядковый номер параметра в пакете.

Столбец «**Name**» содержит название параметра. Данное поле может принимать следующие значения:

- Energy – энергия;
- Volume – объем;
- Mass – масса;
- On Time – время работы;
- Operating Time – текущее время;
- Power – мощность;
- Volume Flow – объемный расход;
- Volume Flow ext. – объемный расход расширенный;
- Mass flow – массовый расход;
- Flow Temperature – температура в подающем трубопроводе;
- Return Temperature – температура в обратном трубопроводе;
- Temperature Difference – разность температур;
- External Temperature – внешняя температура;
- Pressure – давление;
- Time Point – временная метка;
- Reserved – зарезервировано для будущих расширений протокола;
- Averaging Duration – продолжительность усреднения;
- Actuality Duration – актуальная длительность;
- Fabrication No – заводской номер;
- Bus Address – адрес, назначенный устройству в сети M-Bus;
- Manufacturer Specific – параметр, определяемый производителем устройства;
- Credit – кредит;
- Debit – дебет;

- Access Number – номер доступа;
- Medium – среда;
- Manufacturer – производитель;
- Param.set ident. – (Parameter set identification) идентификация набора параметров;
- Model/Version – модель/версия;
- Hardware version – аппаратная версия;
- Firmware version – версия прошивки;
- Software version – версия программного обеспечения;
- Customer location – местоположения клиента;
- Customer – клиент;
- Access Code User – пользовательский код доступа;
- Access Code Operator – код доступа оператора;
- Access Code Sys.Operator – код доступа системного оператора;
- Access Code Developer – код доступа разработчика;
- Password – пароль;
- Err.flags – флаги ошибок;
- Err.mask – маска ошибок;
- Digital Output – цифровой выход;
- Digital Input – цифровой вход;
- Baudrate – скорость передачи;
- resp.timeout – таймаут ожидания ответа;
- Retry – повтор;
- First storage – первое хранилище;
- Last storage – последнее хранилище;
- Size of storage block – размер одного хранилища;
- Storage interval [sec] – интервал сохранения в секундах;
- Storage interval [min] – интервал сохранения в минутах;
- Storage interval [hour] – интервал сохранения в часах;
- Storage interval [day] – интервал сохранения в днях;
- Storage interval month – интервал сохранения в месяцах;
- Storage interval year – интервал сохранения в годах;
- Duration since last readout [sec] – время с момента последнего считывания [секунд];
- Duration since last readout [min] – время с момента последнего считывания [минут];
- Duration since last readout [hour] – время с момента последнего считывания [часов];
- Duration since last readout [day] – время с момента последнего считывания [дней];
- Start tariff – начало тарифа;
- Duration of tariff [min] – длительность тарифа [минуты];
- Duration of tariff [hour] – длительность тарифа [часы];
- Duration of tariff [day] – длительность тарифа [дни];
- Period of tariff [sec] – период тарифа [секунд];

- Period of tariff [min] – период тарифа [минуты];
- Period of tariff [hour] – период тарифа [часы];
- Period of tariff [day] – период тарифа [дни];
- Period of tariff [month] – период тарифа [месяцы];
- Period of tariff [year] – период тарифа [года];
- dimensionless – безразмерный;
- Voltage – напряжение;
- Current – ток;
- Cold/Warm Temperature Limit – холодный/теплый температурный предел;
- simul. count max power – совокупное количество максимальной мощности.

Столбец «**Value**» содержит значение параметра. Формат поля зависит от типа параметра и единиц измерения.

Столбец «**Unit**» содержит единицы измерения параметра.

Столбец «**Type**» содержит тип параметра. Данное поле может принимать следующие значения:

- пустая строка – для данного параметра не предусмотрен тип;
- Instant. – мгновенное значение;
- Maximum – максимальное значение;
- Minimum – минимальное значение;
- Error – данный параметр относится к ошибке.

Столбец «**Module**» содержит номер модуля, к которому относится параметр. По спецификации M-Bus в одном устройстве могут находиться несколько модулей, собирающих разнотипные данные. Для того чтобы разделить эти данные в пакете для каждого параметра выставляется номер модуля, к которому он относится.

Столбец «**Storage**» задает номер хранилища, к которому относится данный параметр. Как правило, с помощью данного поля задается номер архивной записи.

Столбец «**Tariff**» задает номер тарифа, к которому относится данный параметр.

Столбец «**CRC**» содержит контрольную сумму текущей строки. Контрольная сумма считается от начала строки до поля контрольной суммы, алгоритм расчета контрольной суммы описан в приложении А. Контрольная сумма передается как шестнадцатеричное число в текстовом виде с префиксом 0x.

**Для заметок**