

МАГИСТРАЛЬНЫЙ ПРОТОКОЛ М4

Руководство программиста

РАЖГ.00293-33

Отдельные изменения, связанные с дальнейшим совершенствованием изделий, поддерживающих магистральный протокол М4, могут быть не отражены в настоящем 1-м издании.

РОССИЯ, 190020, г. Санкт-Петербург, наб. Обводного канала, 150
Тел./факс: (812) 2522940, 4452745; adm@logika.spb.ru; www.logika.spb.ru

СОДЕРЖАНИЕ

1 Общие сведения	4
1.1 Область применения	4
1.2 Принятые обозначения	4
1.3 Принципы организации магистрали.....	4
2 Структура сообщений.....	5
2.1 Базовый формат сообщений	5
2.2 Сокращенный формат сообщений.....	6
2.3 Теги.....	6
2.4 Основные типы структурны элементов.....	8
3 Процедуры взаимодействия абонентов.....	10
3.1 Инициализация сеанса обмена	11
3.2 Обработка ошибок	11
3.3 Чтение параметров	12
3.4 Запись параметров	13
3.5 Чтение архивов.....	14
3.6 Управление счетом	16
3.7 Выбор скорости обмена	17
4 Перечень ссылок.....	18
ПРИЛОЖЕНИЕ А – Контрольные коды сообщений	19
ПРИЛОЖЕНИЕ Б – Аппаратное построение магистрального интерфейса.....	21

1 Общие сведения

1.1 Область применения

Магистральный протокол М4 (далее – протокол М4 или протокол) предназначен для организации передачи данных в информационно-измерительных системах, построенных с применением средств коммерческого энергоучета ЗАО НПФ ЛОГИКА. Протокол ориентирован на использование в качестве системообразующего для энергонезависимых приборов семейства СПТ94х и СПГ74х.

1.2 Принятые обозначения

В настоящем описании используются следующие формы представления чисел: шестнадцатеричная, двоичная и десятичная. Шестнадцатеричные числа записываются с префиксом “0x”. Для записи двоичных чисел используется суффикс “b”. Числа, записанные без перечисленных атрибутов, следует рассматривать как десятичные.

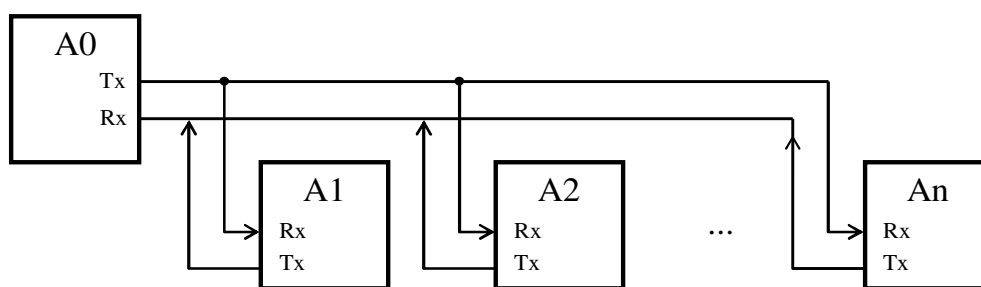
Например, десятичное число 137 эквивалентно двоичному 10001001b и шестнадцатеричному 0x89.

1.3 Принципы организации магистрали

Обмен данными по протоколу М4 может вестись как с одиночным устройством в режиме «точка-точка», так и с множеством устройств (далее – абонентов), подключенных к единой магистрали.

Логически магистраль строится по принципу «ведущий/ведомый» с одним ведущим и множеством ведомых абонентов. Инициатором активности на магистрали является ведущий абонент.

Физическое объединение абонентов осуществляется с помощью RS232-совместимого интерфейса (далее – магистральный интерфейс или интерфейс М4). Принцип организации магистрали иллюстрирует рисунок 1.



A0 – ведущий абонент;
A1...An – ведомые абоненты.

Рисунок 1 – Построение магистрали

Обмен осуществляется структурированными последовательностями байтов – сообщениями.

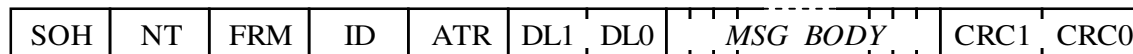
Передача байтов выполняется в асинхронном старт-стоповом полудуплексном режиме с одним стартовым, одним стоповым битами и восемью битами данных. Контроль четности не используется. Порядок передачи битов – «младшим битом вперед». Скорость передачи выбирается из ряда: 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с. Подробное описание аппаратного уровня построения магистрали приведено в приложении Б.

Обмен между двумя абонентами в режиме «точка-точка» может быть организован посредством интерфейса RS232, оптического интерфейса МЭК1107 и магистрального интерфейса. Возможность выбора того или иного способа подключения определяется конкретным типом устройства-абонента.

2 Структура сообщений

2.1 Базовый формат сообщений

Сообщение протокола М4 состоит из заголовка, тела, и блока контрольной информации. Общий вид сообщения приведен ниже.



Где:

SOH – управляющий код начала сообщения;

NT – сетевой номер абонента-адресата сообщения;

FRM – код формата сообщения;

ID – байт идентификатора сообщения;

ATR – байт атрибутов сообщения;

DL1, DL0 – длина тела сообщения (младший и старший байты);

MSG BODY – тело сообщения;

CRC1, CRC0 – шестнадцатитбитный циклический контрольный код (младший и старший байты).

Управляющий код начала сообщения SOH служит для синхронизации приема на фоне помех. Его значение фиксировано:

$$\text{SOH} = 0x10$$

Сетевой номер NT может принимать значения 0...255, причем NT=255 используется для безадресного обращения к абоненту. При получении запроса с NT=255 абонент выполняет его обработку независимо от действительного значения своего сетевого адреса. Во всех остальных случаях запросы, содержащие NT, не соответствующие сетевому адресу абонента, игнорируются.

Поле формата сообщения FRM предназначено для выделения сообщений, передача которых ведется в сокращенной форме (2.2). В таких сообщениях значение этого поля отлично от 0x90. В остальных случаях его значение фиксировано:

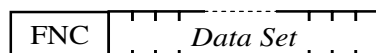
$$\text{FRM} = 0x90$$

Идентификатор ID служит для организации упорядоченного разбора сообщений при их приеме. Байт ID, полученный абонентом в запросе, без изменений копируется в соответствующее поле ответа. Механизм идентификации сообщений позволяет восстановить исходный порядок их следования при использовании сред передачи информации, где такой порядок не гарантируется.

Поле атрибутов ATR зарезервировано для использования в последующих версиях протокола. В него должно подставляться значение:

$$\text{ATR} = 0x00$$

Тело сообщения MSG BODY может иметь произвольную длину в пределах 0...65535 байтов. При этом его структура всегда детерминирована: тело сообщения состоит из однобайтового кода запроса FNC и блока данных (Data Set):



Циклический контрольный код охватывает все сообщение за исключением поля SOH. Для его вычисления используется алгоритм CRC-16 ССИТТ с порождающим полиномом:

$$P = X^{16} + X^{12} + X^5 + 1$$

Пример реализации алгоритма вычисления контрольного кода на языке Си приведен в приложении А.

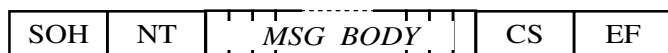
2.2 Сокращенный формат сообщений

Большинство сообщений протокола М4 могут быть представлены в сокращенном формате.

Сокращенный формат сообщений введен для обеспечения совместимости «снизу вверх» с некоторыми, не поддерживающими протокол М4, устройствами, выпускаемыми фирмой ЛОГИКА.

Сокращенный формат сообщения отличается от базового формата отсутствием полей FRM, ID, ATR и DL. Вместо шестнадцатибитого циклического контрольного кода используется восьмидесятибитная контрольная сумма. К концу сообщения приписывается управляющий код конца кадра EF.

На запрос, представленный в сокращенном формате, ответ формируется также в сокращенном формате. Общий вид сообщения:



Управляющие поля SOH и EF имеют значения:

SOH=0x10;

EF=0x16.

Контрольная сумма CS вычисляется путем арифметического сложения всех байтов сообщения за исключением полей SOH и EF. Из полученной суммы выделяется младший байт и производится его побитовое инвертирование. Пример реализации такого алгоритма на языке Си приведен в приложении А.

2.3 Теги

Протоколом М4 предусмотрена передача в одном сообщении набора разнородных параметров, имеющих различный тип, размер, и формат представления. Однозначное декодирование таких сообщений получателем обеспечивается путем дополнительного структурирования информации по определенным правилам с помощью специального набора управляющих символов – тегов.

Каждый тег представляет собой однобайтовый код, который приписывается к началу соответствующего ему структурного элемента сообщения. Принципы структурирования данных построены на основе требований системы стандартов ASN.1 [1...4].

Номенклатура предусмотренных протоколом М4 тегов приведена в таблице 1.

Таблица 1 – Набор тегов

Тег	Код	Тип структурного элемента
OCTET STRING	0x04	Строка октетов
NULL	0x05	Нет данных
ASCIIString	0x16	Строка ASCII-символов
SEQUENCE	0x30	Последовательность
IntU	0x41	Беззнаковое целое (unsigned int)
IntS	0x42	Целое со знаком (int)
IEEEFloat	0x43	Число с плавающей точкой IEEE 754 Float
MIXED	0x44	Параметр со комбинированным значением int+float
Operative	0x45	Оперативный параметр настроечной БД
ACK	0x46	Подтверждение
TIME	0x47	Текущее время
DATE	0x48	Текущая календарная дата
ARCHDATE	0x49	Дата архивной записи
PNUM	0x4A	Номер параметра
FLAGS	0x4B	Сборка флагов
ERR	0x55	Ошибка

С учетом тегов тело сообщения может быть представлено в следующем виде:

FNC

TAG ₁	DL ₁	DATA ₁
TAG ₂	DL ₂	DATA ₂

...

TAG _N	DL _N	DATA _N
------------------	-----------------	-------------------

Где: TAG₁...TAG_N – теги;

DL₁...DL_N – поля длины последующих сегментов данных;

DATA₁...DATA_N – данные структурного элемента.

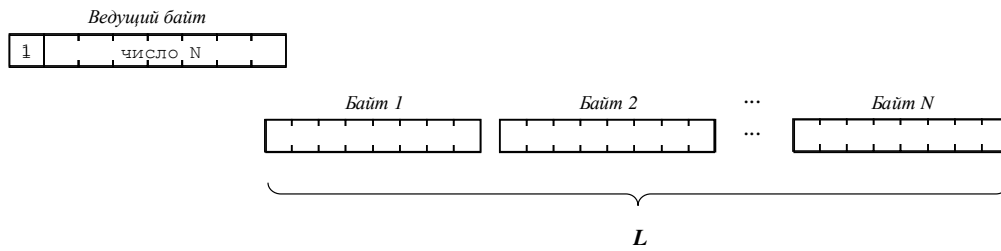
Поле длины может кодироваться как одним, так и несколькими байтами. Первый способ кодирования будем называть простым, второй – составным.

При простом кодировании старший бит поля длины должен быть равен нулю. Оставшиеся семь битов интерпретируются собственно как двоичный код значения длины, таким образом, в простой форме могут быть представлены длины от 0 до 127 байтов.

Например, значение длины L=52 (0x34) будет представлено как:

00110100b

При составном кодировании поля длины старший бит ведущего байта должен быть равен единице. Оставшиеся семь битов содержат двоичный код числа N последующих байтов, которыми собственно кодируется значение длины L. Значение L интерпретируется как двоичное целое без знака. Байт 1 при этом является старшим, байт N – младшим:



По усмотрению отправителя сообщения количество байтов в составном поле длины может не быть минимально необходимым для передачи того или иного значения.

Для примера рассмотрим кодирование величины $L=421$.

Чтобы представить число 421 (0x1A5) в двоичной форме необходимо два байта. Всего же, с учетом ведущего байта, при минимальном кодировании, поле длины будет состоять из трех байтов:

10000010b 00000001b 10100101b

При кодировании с избыточностью, например, четырьмя байтами, получим следующее представление поля длины:

10000100b 00000000b 00000000b 00000001b 10100101b

2.4 Основные типы структурны элементов

Ниже приводится описание некоторых из перечисленных выше структурных элементов.

При этом рассматриваются только элементы, жестко не связанные с тем или иным типом сообщений настоящего протокола обмена. Речь о таких элементах пойдет далее – в детальном описании процедур взаимодействия абонентов на магистралах.

2.4.1 IntU, IntS, IEEFloat

Элементы IntU, IntS, IEEFloat используются соответственно для передачи чисел в форматах: целое без знака (unsigned int), целое (int), число с плавающей точкой IEEE 754 float.

Длина блока данных элемента IEEFloat фиксирована и составляет четыре байта

Длина блока данных для элементов IntU и IntS, как правило, определяется передаваемым значением исходя из минимально необходимого для его кодирования числа байтов. Однако, по усмотрению отправителя сообщения, кодирование может осуществляться с избыточностью в виде нулевых старших байтов.

Для перечисленных структурных элементов соблюдается общее правило: данные передаются «младшим байтом вперед», т.е. старший байт передается последним. Для чисел с плавающей точкой сначала в указанной последовательности передаются три байта мантииссы, а затем – байт двоичного порядка.

Приведем для примера вид структурного элемента IntU, с помощью которого представлено число 421 (0x1A5):

TAG	DL	DATA
0x41	0x02	0xA5 0x01

2.4.2 MIXED

Структурный элемент MIXED является комбинацией из двух составляющих: целого числа (int) и числа с плавающей точкой IEEE754 float. Размер каждой из составляющих фиксирован и составляет 4 байта, поэтому длина поля данных элемента MIXED всегда составляет 8 байтов. Общий вид структурного элемента MIXED показан ниже:

<i>TAG</i>	<i>DL</i>	<i>DATA</i>			
0x44	0x08		<i>int</i>		<i>float</i>

При декодировании элемента MIXED необходимо сложить его составные части. Для этого они должны быть преобразованы единый формат, обеспечивающий адекватную точность вычислений, например, IEEE754 double.

2.4.3 ASCIIString

Элемент ASCIIString используется для передачи текстовой информации. Набор символов соответствует кодировке Windows-1251.

2.4.4 TIME, DATE

Элементы TIME и DATE предназначены соответственно для передачи значений текущего времени, и даты.

Элемент TIME имеет формат:

<i>TAG</i>	<i>DL</i>	<i>DATA</i>			
0x47	0x04	SS/256	SS	MM	HH

Где:

SS/256 – тики таймера, кратные 1/256 доле секунды;

SS – секунды;

MM – минуты;

HH – часы.

Формат элемента DATE:

<i>TAG</i>	<i>DL</i>	<i>DATA</i>			
0x48	0x04	DD	MM	YY	DW

Где:

DD – календарный день;

MM – календарный месяц;

YY – год относительно начала тысячелетия;

DW – день недели.

Значение байта YY вычисляется как:

$$YY = (\text{год} - 2000)$$

Например, год=2012 кодируется как YY=12.

Дни недели кодируются числами от нуля до шести в порядке возрастания. При этом понедельнику соответствует значение 0.

2.4.5 FLAGS

Элементы типа FLAGS предназначены для передачи позиционно-кодированных логических сборок. Каждый бит такой сборки определяет состояние флага, с номером, соответствующим позиции этого бита относительно младшего бита сборки. Младшим битом, т.е. битом с нулевым номером, считается младший бит младшего байта сборки.

Порядок следования байтов в элементах FLAGS соответствует общему правилу: «младшим байтом вперед».

Пример:

<i>TAG</i>	<i>DL</i>	<i>DATA</i>	
0x4B	0x02	0x61	0x80

Показанный выше элемент FLAGS передает значение сборки из двух байтов: 01100001b 10000000b, что соответствует активным флагам с номерами 0,5,6,15.

3 Процедуры взаимодействия абонентов

Перечень сообщений, предусмотренных текущей версией протокола приведен в таблице 2.

Таблица 2 – Типы сообщений протокола M4

Сообщение	Код сообщения (FNC)	Наличие сокращенной формы
Ошибка	0x21	+
Запрос сеанса связи	0x3F	+
Запрос изменения скорости обмена	0x42	+
Запрос управления счетом	0x4F	+
Запрос поиска архивной записи	0x61	-
Запрос чтения параметра	0x72	-
Запрос записи параметра	0x77	-

Все сообщения можно условно разделить на два типа: информативные и управляющие.

С помощью управляющих сообщений осуществляется контроль над процессом обмена данными по магистрали, а также, управление основными режимами работы устройств-абонентов. К таким сообщениям относятся запрос сеанса связи (0x3F), сообщение об ошибке (0x21), запрос изменения скорости обмена (0x42), а также, запрос управления счетом (0x4F).

Управляющие сообщения имеют блоки данных фиксированной длины от одного до четырех байтов (в зависимости от типа сообщения). Какого-либо структурирования информации с помощью тегов для таких сообщений не предусмотрено.

К информативным относятся сообщения, обеспечивающие запись и чтение параметров, а также, чтение архивных данных (0x61, 0x72, 0x77).

Ниже представлено развернутое описание сообщений протокола M4 и процедур, связанных с их обработкой. При этом показаны только тела сообщений. Управляющие поля заголовков и контрольные коды условно не приводятся.

При рассмотрении сообщений, данные в которых структурированы тегами, подразумевается, что вся необходимая пользователю информация о принципах расстановки тегов, изложена ранее (3). Для определения значений тегов следует руководствоваться таблицей 1.

3.1 Инициализация сеанса обмена

Обмен с абонентом магистрали должен начинаться со специальной процедуры инициализации сеанса. При выполнении указанной процедуры абонент, с которым предстоит вести обмен, переходит в режим ожидания адресованных ему запросов. Остальные абоненты магистрали напротив – игнорируют весь трафик вплоть до начала новой процедуры инициализации сеанса с одним из них.

Процедура инициализации сеанса начинается с передачи в магистраль стартовой последовательности из 16 байтов 0xFF. По завершении передачи, перед началом дальнейшего обмена, должен быть выдержан некоторый таймаут T_{start} .

Далее передается запрос сеанса:

0x3F

0x00	0x00	0x00	0x00
------	------	------	------

На запрос сеанса абонент отвечает сообщением вида:

0x3F

DVC_L	DVC_H	VX
-------	-------	----

Где: DVC_L, DVC_H – код устройства;
VX – идентификатор исполнения.

Обмен данными в рамках одного сеанса может вестись сколь угодно долго при соблюдении одного условия: обмен не должен прерываться более чем на 1 минуту. В противном случае, сеанс связи автоматически закрывается абонентом и для его возобновления необходимо повторно выполнить описанную выше процедуру.

Длительность таймаута T_{start} определяется конкретным типом устройства-абонента и содержится в соответствующих спецификациях устройства.

При наличии в на магистрали нескольких устройств разного типа с разными требованиями к величине T_{start} , следует выбирать наибольшее значение. Общий подход к определению данного параметра следующий: величина T_{start} не должна быть меньше, но может быть сколь угодно больше значения, указанного в спецификациях устройства выбранного типа.

3.2 Обработка ошибок

Обработка полученного сообщения абонентом выполняется в следующем порядке:

- проверка контрольного кода сообщения;
- анализ принятого запроса;
- выполнение запроса и формирование ответа.

На первом этапе происходит отсев разрушенных в ходе передачи сообщений. Дальнейшая обработка таких сообщений не выполняется. Ответы на них не формируются.

Второй этап предусматривает формальную проверку структуры данных принятого запроса. На этом этапе выполняется проверка значений передаваемых параметров на соответствие допускемым пределам, целостность отдельных структурных элементов сообщения, выделенных тегами и т.д.

На третьем этапе выполняются предусмотренные запросом операции, и формируется ответ.

При возникновении ошибок на втором или третьем этапах обработки входящего сообщения, абонент формирует ответ:

0x21

Код ошибки

Коды возможных ошибок приведены в таблице 3.

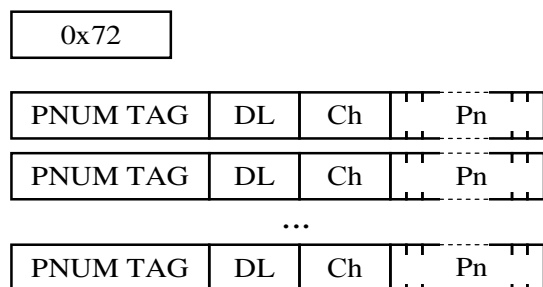
Таблица 3 – Коды ошибок

Код	Ошибка
0x00	Нарушение структуры запроса
0x01	Защита от записи
0x02	Недопустимые значения параметров запроса

3.3 Чтение параметров

Чтение параметров абонента осуществляется произвольными наборами, порядок следования, тип и количество параметров в которых определяются отправителем запроса.

Запрос состоит из структурированного тегами PNUM TAG набора указателей на параметры, значения которых должны быть получены. Каждый указатель состоит из номера канала Ch и номера параметра Pn. Общий вид запроса:

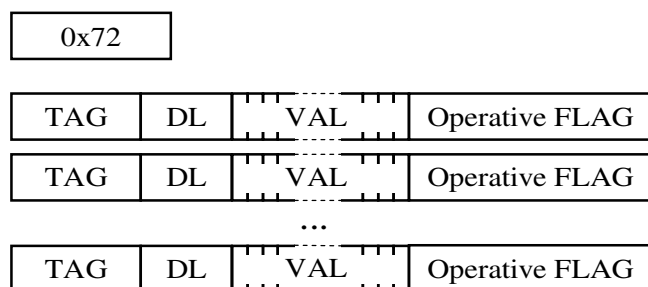


Номер канала Ch кодируется одним байтом.

Номер параметра может быть представлен одним или несколькими байтами. В последнем случае порядок следования байтов подчиняется правилу: «младшим байтом вперед».

Ответ абонента содержит набор значений запрошенных параметров. Порядок их следования в ответе соответствует порядку следования указателей в запросе.

Общий вид ответа:



Где:

VAL – значение параметра;

TAG – тег, соответствующий типу параметра;

Operative FLAG – поле, индицирующее оперативные настроечные параметры.

Понятие «оперативный параметр» присуще устройствам, возможность изменения настроек которых аппаратно блокируется после ввода устройства в эксплуатацию. Возможность изменять значения оперативных параметров при этом остается.

Поле Operative FLAG является опциональным. Оно подключается к полю VAL при одновременном соблюдении следующих условий:

- параметр является настроечным параметром абонента;
- параметр сконфигурирован как оперативный.

Структура поля показана ниже:

Operative TAG	DL=1	Flag
---------------	------	------

Значение байта Flag интерпретируется следующим образом:

- 0 – параметр не оперативный;
- 1 – параметр оперативный.

По усмотрению отправителя сообщения, к неоперативным параметрам поле Operative FLAG может не подключаться вообще или подключаться со значением байта FLAG=0.

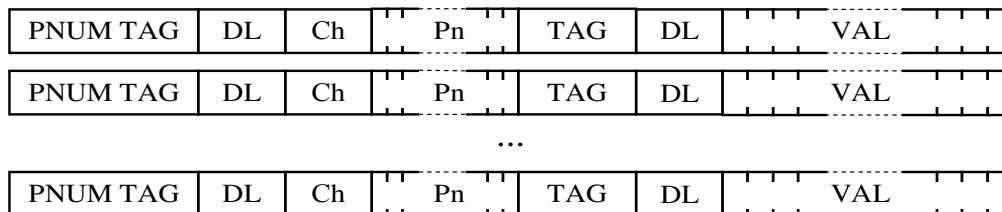
Поля сообщения TAG в зависимости от типа соответствующего им структурного элемента могут принимать значения:

- OCTET STRING;
- MIXED;
- ASCIIStrng;
- IEEFloat;
- IntU;
- IntS;
- NULL.

3.4 Запись параметров

Запись параметров осуществляется с помощью запроса:

0x77



Где:

- Ch – номер канала;
- Pn – номер параметра;
- VAL – значение параметра.

Запрос состоит из произвольного набора параметров, запись которых должна быть выполнена.

Структурирование элементов запроса осуществляется тегами PNUM TAG и TAG. Значение последнего определяется типом передаваемого параметра и в общем случае может быть равным:

- OCTET STRING;
- MIXED;
- ASCIIStrng;
- IEEFloat;
- IntU;
- IntS;
- NULL.

При записи настроечного параметра последний может быть одновременно сконфигурирован как оперативный или неоперативный. Для этого вслед за полем VAL к описывающей параметр структуре может быть присоединено поле Operative FLAG. Таким образом, каждая из строк представленного выше запроса может иметь вид:

PNUM TAG	DL	Ch	Pn	TAG	DL	VAL	Operative FLAG
----------	----	----	----	-----	----	-----	----------------

Формат поля Operative FLAG описан в (3.3).

Процедура назначения параметров оперативными или неоперативными может быть использована самостоятельно, т.е. без передачи параметру нового значения. Для этого элемент VAL может не включаться в запрос, с учетом чего каждая строка представленного выше запроса записи может иметь в вид:

PNUM TAG	DL	Ch	Pn	Operative FLAG
----------	----	----	----	----------------

После обработки запроса записи абонент возвращает отчет вида:

0x77		
ACK/ERR TAG	DL	Option
ACK/ERR TAG	DL	Option
...		
ACK/ERR TAG	DL	Option

Отчет содержит результаты записи каждого параметра из полученного набора. В зависимости от результата, поле ACK/ERR TAG может содержать тег подтверждения (ACK TAG) или тег ошибки (ERR TAG).

Поле Option передает коды ошибок, если таковые возникают в процессе записи. Перечень кодов ошибок соответствуют таблице 3. При успешной записи поле Option имеет нулевую длину.

3.5 Чтение архивов

Для получения от абонента накопленных им архивных данных используется запрос:

0x61						
OCTET STRING TAG	DL	CNT_L	CNT_H	Ch	Rectype	N
ARCHDATE TAG	DL	Start Date				
ARCHDATE TAG	DL	End Date				

Где:

CNT_L, CNT_H – номер архивного раздела (младший и старший байты);

N – максимальное количество включаемых в ответ записей;

Ch – канал.

Поле Rectype определяет тип запрашиваемых записей:

- 0 – часовые;
- 1 – суточные;
- 2 – декадные;
- 3 – месячные;
- 4 – изменения БД;
- 5 – перерывы электропитания;
- 6 – НС.

Поля Start Date, End Date определяют начальную и конечную границы интервала времени, в котором должны находиться запрашиваемые записи. Развернутая структура этих полей следующая:

ARCHDATE TAG	DL=8	YY	MH	DD	HH	MM	SS	ms_l	ms_h
--------------	------	----	----	----	----	----	----	------	------

Где:

YY-MH-DD-HH-MM-SS – год, месяц, день, час, минута, секунда;
 ms_l, ms_h – миллисекунда (младший и старший байты).

В общем случае поля Start Date и End Date могут иметь переменную длину. Если длина поля менее восьми байтов, то это означает, что из него исключены один или несколько элементов по принципу «от менее весомого к более весомому». Например, в запросе чтения месячных записей поля Start Date и End Date могут передаваться в виде:

ARCHDATE TAG	DL=2	YY	MH
--------------	------	----	----

Поле End Date может отсутствовать в сообщении. В этом случае чтение будет всегда выполняться в сторону возрастания дат (от прошлого к настоящему), а количество записей будет ограничено значением поля N.

Использование в сообщении обоих полей позволяет выбирать направление чтения:

При Start Date<End Date включение архивных записей в ответ абонента будет выполняться в направлении «от более ранних записей к более поздним»;

При Start Date>End Date – «от более поздних записей к более ранним».

Байт YY содержит значение года относительно начала тысячелетия. Его формат аналогичен формату, описанному в (2.4.4) для структурного элемента DATE.

Понятие «архивный раздел» относится к устройствам, архивы которых имеют соответствующее структурное деление. Архивные разделы обозначаются номерами от 0 до 65535. При формировании запроса данных из последнего, т.е. текущего открытого раздела, поле CNT следует задавать равным 0xFFFF.

Ответ абонента имеет вид:

0x61

ARCHDATE TAG	DL	Date	SEQUENCE TAG	DL	Record
--------------	----	------	--------------	----	--------

ARCHDATE TAG	DL	Date	SEQUENCE TAG	DL	Record
--------------	----	------	--------------	----	--------

...

ARCHDATE TAG	DL	Date	SEQUENCE TAG	DL	Record
--------------	----	------	--------------	----	--------

ARCHDATE TAG	DL	Date (+1)	SEQUENCE TAG	DL=0
--------------	----	-----------	--------------	------

Поле Date(+1) содержит дату записи, ближайшую к заданному в запросе интервалу времени: при Start Date < End Date – в сторону возрастания; при Start Date > End Date – в сторону убывания. При отсутствии таковых поле Date(+1) будут иметь нулевую длину.

Например, при полном отсутствии записей в архиве будет сформирован ответ:

0x61

ARCHDATE TAG	DL=0	SEQUENCE TAG	DL=0
--------------	------	--------------	------

Поле Record содержит собственно архивную запись.

Внутри записи отдельные параметры разделены тегами, т.е. каждый параметр представлен структурой:

TAG	DL	VAL
-----	----	-----

Поле TAG, в зависимости от типа параметра, может принимать значения:

- OCTET STRING;
- ASCIIString;
- IEEEFloat;
- IntU;
- IntS;
- NULL.

Поле N в запросе может принимать значения 0...255.

В ответ на запрос с N=0 будет возвращен лишь указатель Date(+1). Собственно архивные данные в ответ на такой запрос не включаются.

Запрос с N=0 удобно использовать для определения даты первой архивной записи. Для этого поле Start Date заполняется нулями. Поле End Date не подключается к запросу или заполняется кодом 0xff.

В зависимости от конкретного типа устройства набор формируемых им архивов может быть неполным. В этом случае, при запросе данных из непредусмотренного для данного устройства архива будет сформировано сообщение с кодом ошибки 0x02 – «недопустимые значения параметров запроса».

3.6 Управление счетом

Запрос предназначен для выполнения пуска/останова счета, сброса накопленных абонентом архивных данных и тотальных значений параметров. Общий вид запроса:

0x4F

CMD	OP	Ch	0x00
-----	----	----	------

Где: CMD – код команды;

OP – опции;

Ch – код канала.

Предусмотрены следующие значения кодов команд:

0 – останов счета;

1 – пуск счета (начало нового раздела);

0xFF – сброс архивов.

Поле OP может принимать значения:

0 – не сбрасывать накопленные абонентом значения тотальных параметров;

1 – сбросить (занулить) значения тотальных параметров.

После обработки запроса абонент формирует подтверждение вида:

0x4F

В зависимости от типа устройства-абонента те или иные команды, предусмотренные запросом, могут не поддерживаться. В этом случае абонентом будет сформировано сообщение с кодом ошибки 0x02 «Недопустимые значения параметров запроса».

3.7 Выбор скорости обмена

Скорость обмена с абонентом может быть изменена с помощью описанной ниже процедуры. При этом начальная скорость обмена, т.е. скорость на этапе установления сеанса связи определяется начальными установками абонента.

Запрос выбора скорости обмена:

0x42

S	0x00	0x00	0x00
---	------	------	------

Где: байт S – код скорости обмена.

Значение 0x00 соответствует скорости 2400, 0x01 – 4800, 0x02 – 9600, 0x03 – 19200, 4 – 38400, 5 – 57600, 6 – 115200 бит/с.

Подтверждение:

0x42

Подтверждение передается на неизменной скорости, после чего вступает в силу ее новое значение.

В случае, если запрашиваемая скорость не поддерживается абонентом, будет сформирован ответ с кодом ошибки 0x02 «Недопустимые значения параметров запроса».

При отсутствии обмена в течение 10 с абонент автоматически перестраивает свой интерфейс на исходную скорость.

Описанная процедура изменения скорости обмена может применяться только при обмене с одиноким абонентом. При наличии на магистрали нескольких абонентов изменять скорость обмена недопустимо.

4 Перечень ссылок

1. ITU-T X680 – Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.
2. ITU-T X681 – Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.
3. ITU-T X690 – Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).
4. ГОСТ Р ИСО/МЭК 8825-1-2003 - Информационная технология. Правила кодирования АСН.1. Часть 1. Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования.

ПРИЛОЖЕНИЕ А

Контрольные коды сообщений

Ниже приводятся реализации алгоритмов вычисления контрольных кодов сообщений на языке Си.

1 CRC16

```
// Функция вычисляет циклический код для
// последовательности из len байтов, указанной *msg.
// Алгоритм CRC-16 ССИТТ.
// Используется порождающий полином:
// (X в степени 16)+(X в степени 12)+(X в степени 5)+1.
// Полиному соответствует битовая маска 0x1021.
//
INT16U CRC16Calc(unsigned char *msg, INT16U len)
{
    INT16U crc=0;
    INT16U j;

    while (len-- > 0)
    {
        crc = crc ^ (INT16U)*msg++ << 8;
        for ( j=0; j < 8; j++ )
        {
            if(crc & 0x8000)
                crc = (crc << 1) ^ 0x1021;
            else
                crc <<= 1;
        }
    }

    crc=crc*256+crc/256;
    return crc;
}
```

Пример

В качестве примера рассмотрим сообщение протокола М4, содержащее запрос сеанса связи:

```
0x10 0xff 0x90 0x00 0x00 0x05 0x00 0x3f 0x00 0x00 0x00 0x00 0xd9 0x19
```

Контрольный код сообщения 0x19D9 получен посредством вычислений по приведенному алгоритму. Он охватывает все сообщение за исключением кода начала кадра (0x10), т.е. байты:

```
0xff 0x90 0x00 0x00 0x05 0x00 0x3f 0x00 0x00 0x00 0x00
```

При выполнении получателем сообщения проверки контрольного кода CRC16, удобно пользоваться следующим свойством последнего: если приписать к произвольному блоку информации его CRC16, а затем пересчитать CRC16 полученного таким образом нового блока, то результатом будет ноль. Для приведенного примера нулевой результат дадут вычисления CRC16 следующего массива:

```
0x90 0x00 0x00 0x05 0x00 0x3f 0x00 0x00 0x00 0x00 0xd9 0x19
```

2 Контрольная сумма КС8

```
// Функция вычисляет контрольную сумму
// последовательности из len байтов, указанной *msg.

unsigned char CS8Calc(unsigned char *msg, unsigned char len)
{
    unsigned char sum=0;

    while(len>0)
    {
        sum+=*msg++;
        len--;
    }
    sum^=0xff;
    return sum;
}
```

Пример

Приведем в качестве примера сокращенную форму уже рассмотренного выше запроса сеанса связи:

```
0x10 0xff 0x3f 0x00 0x00 0x00 0x00 0xc1 0x16
```

Контрольная сумма сообщения 0xC1 получена по приведенному алгоритму путем обработки следующего массива байтов:

```
0xff 0x3f 0x00 0x00 0x00 0x00
```

При проверке КС8 на стороне получателя сообщения удобно пользоваться правилом, аналогичным описанному для CRC16: если вычислить КС8 блока информации, последним байтом которого является КС8 всех предшествующих байтов, то результатом будет ноль. Т.е., для приведенного примера, нулевой результат даст вычисление КС8 следующего массива байтов:

```
0x3f 0x00 0x00 0x00 0x00 0xc1
```

ПРИЛОЖЕНИЕ Б

Аппаратное построение магистрального интерфейса

На уровне аппаратной реализации интерфейса М4 решаются следующие задачи:

- минимизация энергопотребления интерфейсными драйверами ведомых абонентов;
- помехоустойчивая передача данных по физической линии;
- гальваническая развязка интерфейсных цепей;
- совместимость с уровнями сигналов интерфейса RS232C.

1 Цепи и сигналы

Интерфейсные цепи ведущего и ведомого абонентов имеют различную структуру.

Интерфейс ведущего абонента состоит из трех линий:

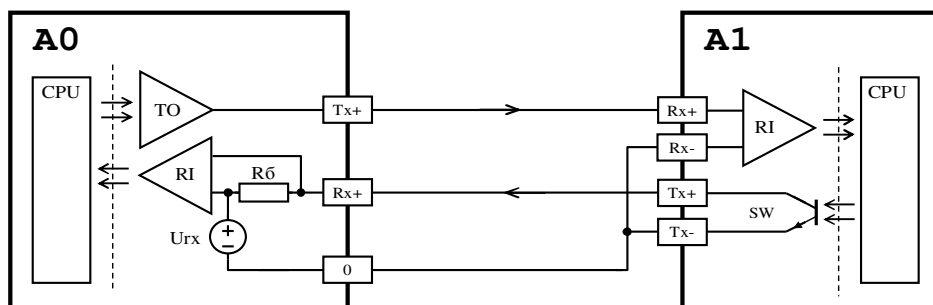
- «Tx+» – выходная цепь передатчика;
- «Rx+» – входная цепь приемника;
- «0» – общая цепь.

Интерфейс ведомого абонента образован четырьмя линиями:

- «Tx+» – положительная выходная цепь передатчика;
- «Tx-» – отрицательная выходная цепь передатчика;
- «Rx+» – положительная входная цепь приемника;
- «Rx-» – отрицательная входная цепь приемника.

Объединение абонентов в магистраль осуществляется с помощью трехпроводной линии связи.

Построение приемопередающих цепей, а также способ взаимного подключения ведущего и ведомого абонентов иллюстрирует рисунок 1.



А0 – ведущий абонент; А1 – ведомый абонент
 ТО – выходной буфер передатчика; RI – входной буфер приемника; CPU – процессор;
 Urx – источник напряжения приемной цепи; Rб – балластный резистор
 SW – передающий ключ.

Рисунок 1 – Структура взаимодействия абонентов

Интерфейсные цепи абонентов отделены от центральных (процессорных) цепей с помощью гальванической развязки, условно показанной на рисунке пунктирной линией. Аналоговая обработка сигналов ведется с помощью буферов-формирователей RI и ТО.

Передача данных ведущим абонентом осуществляется с помощью сигнала напряжения. Состояние лог. «1» передается низким уровнем, а состояние лог. «0» – высоким уровнем напряжения.

Ведомые абоненты осуществляют передачу манипуляцией состоянием выходного ключа SW. Состояние «замкнуто» соответствует передаче лог. «1», состояние «разомкнуто» - лог. «0». При обработке принимаемого сигнала ведущим абонентом, ток, протекающий через балластный резистор Rб, с помощью входного буфера RI сравнивается с некоторым пороговым значением. Превышение порога интерпретируется как лог. «1».

Диаграммы сигналов на интерфейсных линиях ведущего абонента показаны на рисунке 2.

Схема подключения группы абонентов к магистрали показана на рисунке 3.

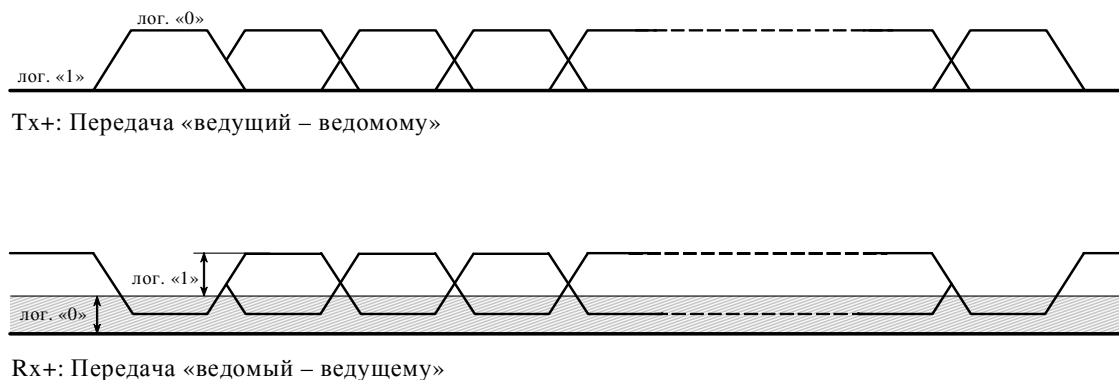
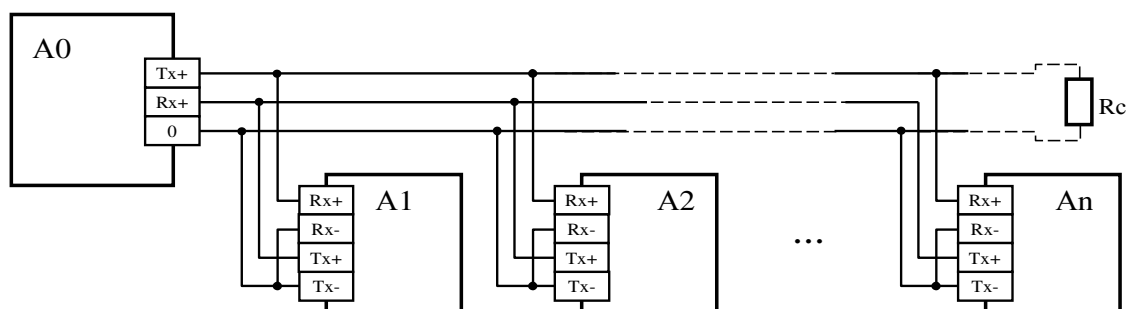


Рисунок 2 – Диаграммы напряжений на интерфейсных цепях ведущего абонента



A0 – ведущий абонент; A1...An – ведомые абоненты;
Rc – согласующий резистор 470 Ом±10%

Рисунок 3 – Схема подключения группы абонентов к магистрали

2 Согласование параметров линии связи

Для оптимального согласования характеристик интерфейсных цепей абонентов с распределенными параметрами линии связи, к самому удаленному от ведущего абонента концу магистрали должен быть подключен согласующий резистор (см. рисунок 3). При прочих равных условиях, применение согласующего резистора позволяет обеспечить максимально возможную протяженность магистрали.

На магистралях, протяженность которых относительно невелика, согласующий резистор не требуется. Для оценки необходимости его подключения можно использовать следующий критерий: согласующий резистор необходим, если суммарная длина сегментов магистрали составляет более $\frac{1}{4}$ от предельного расчетного значения (3).

3 Протяженность магистрали

В общем случае, максимально возможная протяженность магистрали определяется количеством подключенных к ней абонентов, параметрами кабеля и скоростью передачи данных. Стабильная работа магистрали может быть гарантирована при выполнении следующих требований:

- сопротивление каждого проводника линии связи не должно превышать 300 Ом;
- суммарная емкость между любыми двумя проводниками должна быть не более 0,3 мкФ;
- количество абонентов на магистрали – до 32.

Оптимальными характеристиками для использования в качестве магистрального кабеля М4 обладают кабели UTP и FTP категории 5 («витая пара» LAN) с медными жилами диаметром не менее 0,5 мм. Приведенные ниже данные о максимальной протяженности линии связи относятся к этим типам кабелей. При прокладке магистрали кабелем UTP (FTP) используются две пары проводников, соединенные как показано на рисунке 4.



Рисунок 4 – Соединение проводников кабеля UTP

Расчетные значения протяженности магистрали приведены на рисунке 5.

Близкие результаты могут быть получены при использовании любого другого кабеля с сопротивлением жилы до 0,01 Ом/м и погонной емкостью не более 100 пФ/м, например – сигнального кабеля марки J-Y(St)Y 2x2x0,8.

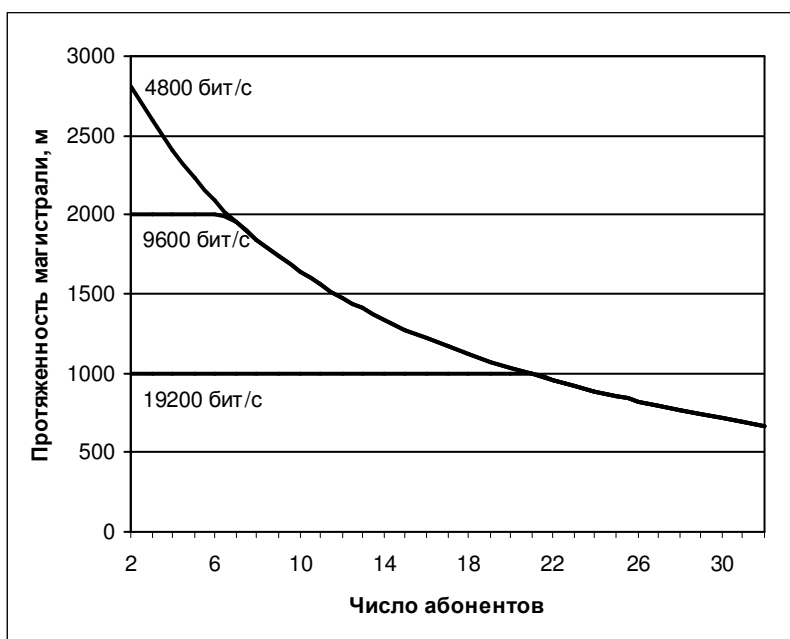
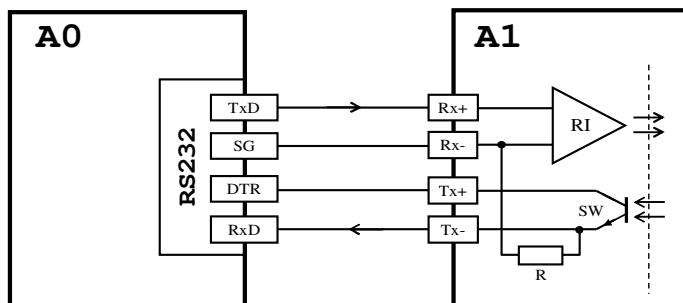


Рисунок 5 – Протяженность магистрали при ее прокладке кабелем UTP (FTP) категории 5

4 Подключение к порту RS232

При локальном обмене данными в качестве ведущего абонента может выступать устройство, оснащенное портом RS232C. Схема подключения ведомого абонента к порту RS232C показана на рисунке 6.



A0 – ведущий абонент с интерфейсом RS232; A1 – ведомый абонент;
 RI – приемный буфер; SW – передающий ключ;
 R – балластное сопротивление передатчика

Рисунок 6 – Подключение абонентов по интерфейсу RS232

Для обеспечения передачи данных цепь DTR порта RS232 должна быть установлена в активное состояние (положительный уровень напряжения).

Протяженность линии связи не должна превышать 100 м.

5 Электрические параметры

Основные требования к электрическим параметрам цепей, установленные для интерфейса М4 приведены в таблице 1.

Таблица 1 – Электрические параметры цепей интерфейса М4

Параметр	Диапазон значений
<u>Ведущий абонент</u>	
Напряжение на выходе передатчика, В: при передаче лог. «0» при передаче лог. «1»	8...11 0...0,2
Выходное сопротивление передатчика, Ом	27... 60
Напряжение между контактами 0 и Rx+ приемника при отсутствии передачи со стороны ведомого абонента, U _{rx} , В	8... 11
Входное сопротивление приемника, Ом	220... 330
Чувствительность приемника, мА	4,5... 10
<u>Ведомый абонент</u>	
Чувствительность приемника, В	1,8...2,2
Входное сопротивление приемника, кОм, не менее	4,7
Сопротивление выходного ключа передатчика: при передаче лог. «0», Ом при передаче лог «1», МОм	не более 30 не менее 1