

ОПИСАНИЕ

протокола обмена между ПЭВМ

и приборами ОВЕН

СОДЕРЖАНИЕ

1	Основные положения	2
1.1	<i>Построение сети RS-485</i>	3
1.2	<i>Технические требования</i>	3
2	Основные принципы обмена по сети	3
3	КАНАЛЬНЫЙ уровень протокола ОВЕН	5
3.1	<i>Метод передачи байта "Тетрада-в-ASCII-символ"</i>	5
3.2	<i>Маркировка начала и конца сообщения (кадра)</i>	5
3.3	<i>Структура кадра</i>	6
3.3.1	<i>Старшая часть адреса сети и расширение.</i>	6
3.3.2	<i>Признак удаленного запроса.....</i>	6
3.3.3	<i>Локальный идентификатор параметра.....</i>	6
3.3.4	<i>Контрольная сумма.....</i>	6
3.3.5	<i>Блок данных канального уровня.....</i>	6
3.3.6	<i>Структура квитанции.....</i>	7
4	Уровень ПРЕДСТАВЛЕНИЯ протокола ОВЕН	7
4.1	<i>Структура данных.....</i>	7
4.2	<i>Локальный идентификатор параметра</i>	7
4.3	<i>Соглашение об именах параметров.....</i>	7
4.4	<i>Преобразование из ASCII-кодов имени параметра в код символа.....</i>	8
4.5	<i>Хеширование имен параметров и вычисление контрольной суммы сообщения.....</i>	8
4.6	<i>Поле данных уровня представления.....</i>	9
4.7	<i>Структура квитанции.....</i>	9
5	Уровень ПРИЛОЖЕНИЯ протокола ОВЕН.....	9
5.1	<i>Формат представления поля данных</i>	9
5.1.1	<i>Формат числа с плавающей точкой.....</i>	10
5.1.2	<i>Знаковое число с односторонней десятичной точкой</i>	10
5.1.3	<i>Часовой формат данных</i>	10
5.1.4	<i>Нетипизированные целые числа или наборы битов</i>	11
5.1.5	<i>Строковый тип</i>	11
5.1.6	<i>Составной тип данных с информацией о времени.....</i>	12
5.1.7	<i>Составной тип данных с дополнительным полем индексации.....</i>	12
5.2	<i>Типы параметров</i>	12
5.3	<i>Индексирование параметров разных типов</i>	12
5.3.1	<i>Индексация конфигурационных параметров</i>	13
5.3.2	<i>Индексация оперативных параметров</i>	14
5.4	<i>Передача ошибок.....</i>	14
5.4.1	<i>Ошибки и исключительные ситуации.....</i>	14
5.4.2	<i>Передача сетевых ошибок.....</i>	14
5.4.3	<i>Передача исключительной ситуации</i>	14
6	Сетевые параметры протокола ОВЕН, Атрибуты параметров.	15
6.1	<i>Сетевые параметры протокола.....</i>	15
6.2	<i>Команды смены сетевых настроек прибора</i>	15
6.3	<i>Атрибуты параметров. Смена атрибутов.....</i>	16
Приложение 1	17	
<i>Таблица №1. Список сетевых ошибок</i>	17	
Приложение 2	19	
<i>Алгоритм обработки исключительной ситуации при передаче прибором</i>	19	
	Синим цветом отмечены отличия от предыдущей версии описания протокола	

1 Основные положения

Данный протокол предназначен для описания процесса обмена информацией между приборами фирмы «Овен» и между приборами и ПЭВМ на базе сети RS-485.

1.1 Построение сети RS-485

1.1.1. Все приборы связаны по сети RS-485 и с ПЭВМ через коммутатор (преобразователь RS-232/485). В частном случае, роль ПЭВМ может выполнять прибор стороннего производителя, поддерживающий сеть RS-232 или RS-485 и имеющий технические возможности приема и обработки информации с произвольного количества приборов, соединенных в сеть RS-485.

1.1.2. Ограничения на кол-во микроконтроллеров определяется размерностью отведенного под базовый адрес кол-ва бит и спецификациями RS-485.

1.2 Технические требования

1.2.1. Устройство, сопряженное с компьютером, поддерживающее на выходе RS-485

1.2.2. Прибор любой модификации, поддерживающий RS-485 и сетевой протокол, описанный в данном документе.

2 Основные принципы обмена по сети

2.1. Сеть имеет единственное ведущее устройство, инициирующее процесс обмена (master). Чаще всего этим устройством является компьютер. Все остальные устройства являются ведомыми (slave) узлами.

2.2. Все операции (команды, обмен данными) производятся к одностороннему обмену сообщениями.

2.3. Параметр однозначно определяется в сети своим идентификатором, составленным из адреса узла, локального идентификатора параметра в приборе и его индексом.

2.4. Каждый кадр в сети должен содержать локальный идентификатор параметра hash-код, который будет рассмотрен дальше.

2.5. Параметры подразделяются на собственно параметры и на команды, которые служат для управляющего воздействия на прибор.

2.6. Запрос ведущего устройства прослушивается всеми ведомыми устройствами и происходит фильтрация принадлежности посылки.

Если отфильтрованное сообщение не содержит запрос, то оно может дальше использоваться двумя способами:

- а)** Если часть полного адреса узла совпадает с базовым адресом прибора, то:
 - переданное значение параметра заменяет старое значение этого параметра в приборе;
 - переданное значение используется в качестве команды управления прибором, узлом прибора или процессом в приборе.
- б)** Если не совпадает с базовым адресом, а отфильтровано сетевыми входами прибора, то переданное значение параметра используется для текущих нужд прибора (например, как значение температуры на входе ПИД-регулятора).

Если отфильтрованное сообщение содержит запрос, но базовая часть адреса в сообщении не совпадает с базовым адресом принявшего его прибора, то данное сообщение отвергается. Иначе, при совпадении базовой части и наличии запроса, в ответ, в сеть, посылается сообщение, содержащее тот же идентификатор и требуемую информацию в поле данных, но без бита удаленного запроса.

2.7. Поскольку по собственной инициативе ни один из slave-узлов не может послать запрос, то сообщение со значением собственного параметра может посылаться slave-узлом только в ответ на запросе от master-узла.

2.8. Каждое пришедшее сообщение должно квитироваться. Квитанция должна быть послана после задержки, большей или равной времени, указанной в параметре rS.dL прибора, но не более максимального тайм-аута в 50 мс. При этом типовая задержка между отправкой

последнего байта посылки до приема первого байта квитанции должна быть не менее rS.dL и не более rS.dL + 5мс. При отсутствии квитанции от прибора в течение 50 мс транзакцию приема-передачи считать сбойной.

2.9. При посылке сообщения или квитанции допускается прерывание потока посылаемых байт на время, не превышающее 50 мс.

2.10. После посылки сообщения или квитанции посылающая сторона должна освободить линию передачи и перейти в режим приема не более чем через 1.1 мс после посылки последнего байта.

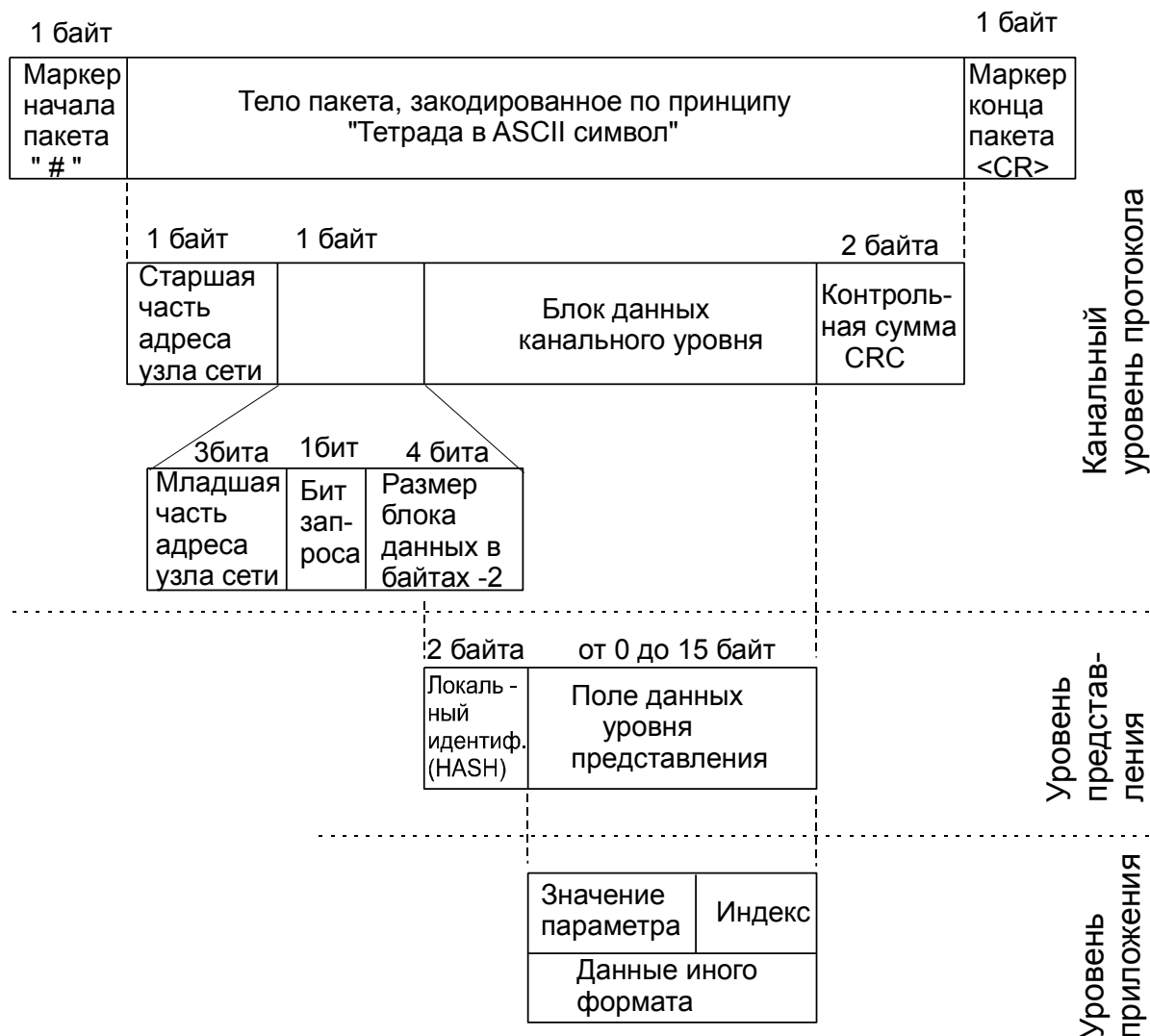
2.11. Каждое сообщение и квитанция передается старшими байтами вперед.

Работа с преобразователем RS-232/RS-485

Для сигнализации полуавтоматическому коммутатору-преобразователю RS-232/RS-485 фирмы OVEN AC3 о приеме/передаче информации по RS-232 на линии устанавливается сигнал RTS в «0»/«1». После выдачи посылки в течении времени не большем, чем установлено в параметре rS.dL принимающей стороны, должно произойти переключение RTS с «1» на «0». В противном случае имеется опасность потери начала квитанции. Полуавтоматические преобразователи сторонних фирм могут использовать дополнительно сигнал DTR для переключения приема/передачи.

Для автоматических преобразователей работа с линиями RTS или DTR не требуется. Однако существует время, в течение которого автоматический преобразователь удерживает линию в состоянии передачи и может потерять начало квитанции прибора. Для работы с автоматическими преобразователями в параметре rS.dL приборов необходимо установить задержку, большую, чем время переключения автоматического преобразователя.

СТРУКТУРНАЯ СХЕМА ПРОТОКОЛА ОВЕН



Физический уровень протокола ОВЕН не описывается, т.к. является открытым стандартом сети RS-485.

3 КАНАЛЬНЫЙ уровень протокола ОВЕН

Каждый байт передаваемого или принимаемого сообщения упаковываются по методу "тетрада-в-ASCII символ";

3.1 Метод передачи байта "Тетрада-в-ASCII-символ"

Передающее устройство должно преобразовать каждую тетраду исходного сообщения в ASCII символ, в зависимости от требуемого формата передачи байта закодированный 7-ми или 8-ми битовым кодом, добавить или не добавить к нему соответствующий бит четности, и снабдить каждый пересылаемый байт старт-битом и соответствующим количеством стоповых бит.

В задачу данного уровня входит преобразование "чистой" двоичной информации в допустимые для передачи по каналу ASCII-символы.

Выбран вариант кодирования тетрад, позволяющий нам иметь собственное адресное пространство, которое не зависит от адресного пространства чужих приборов. Тетрады кодируются следующим образом:

```
0000 -> 'G' = 0x47
0001 -> 'H' = 0x48
0010 -> 'I' = 0x49
0011 -> 'J' = 0x4a
0100 -> 'K' = 0x4b
0101 -> 'L' = 0x4c
0110 -> 'M' = 0x4d
0111 -> 'N' = 0x4e
1000 -> 'O' = 0x4f
1001 -> 'P' = 0x50
1010 -> 'Q' = 0x51
1011 -> 'R' = 0x52
1100 -> 'S' = 0x53
1101 -> 'T' = 0x54
1110 -> 'U' = 0x55
1111 -> 'V' = 0x56
```

На приемной стороне после отбрасывания символов начала и конца кадра производится обратное преобразование ASCII байтов в тетрады с последующим "склеиванием" каждой пары тетрад в полноценный байт двоичной информации. На этом этапе избыточность можно использовать для дополнительного контроля правильности приема, поскольку все символы, не входящие в набор из 16-ти вышеперечисленных, являются ошибочными, символы с не совпавшей вычисленной и переданной четностью также являются сбойными.

3.2 Маркировка начала и конца сообщения (кадра)

Начало и конец кадра обозначаются специальными маркерами. Для этого используются символы, не встречающиеся в других местах кадра, и поэтому позволяющие легко осуществлять локализацию полей принимаемых сообщений и отделять сообщения друг от друга. Хотя для данных целей достаточно использовать лишь один из данных маркеров, для увеличения надежности обмена лучше использовать два. В качестве маркера начала кадра в данном протоколе используется ASCII-символ "#" (код 0x23), а в качестве маркера конца кадра - ASCII символ <CR> (символ возврата каретки, код 0x0d).

3.3 Структура кадра

Сообщение (кадр) имеет следующую структуру:

- 8-битовая старшая часть адреса узла сети (прибора)
- 3-битовое расширение адреса узла сети
- 1-битовый признак удаленного запроса
- 4-битовый размер блока данных канального уровня в байтах –2
- Блок данных канального уровня длиной от 2 до 17 байт
- 16 битовая контрольная сумма

3.3.1 Старшая часть адреса сети и расширение.

В протоколе используются разно-размерная адресация узлов сети: 11-и и 8-и битная.

11-битный адрес узла сети

8-битовый адрес узла сети (старшие биты полного адреса) и 3-битовое расширение адреса узла сети (младшие биты полного адреса) представляют собой полный 11-битный адрес узла сети (прибора или "канала").

В одноканальном приборе такой узел единственный, и поэтому 11-битный адрес относится ко всему прибору. Многоканальный прибор с любой сложностью внутренней иерархической структуры благодаря использованию одномерной индексации, с точки зрения сети, рассматривается как несколько одноканальных приборов с расположенными подряд адресами. В качестве "общего", или Базового адреса прибора берется 11-битный адрес "младшего" канала.

Единственным исключением является посылка широковещательного адреса, который служит для передачи сообщения всем приборам одновременно. В этом случае в базовом адресе прибора посылаются все «1», а значение дополнительного игнорируется.

8-битный адрес узла сети

Для упрощения, как частный случай, применяется и 8-битная адресация. При этом младшие биты полного адреса заполняются нулями. Порядок рассмотрения многоканального прибора с точки зрения одномерной индексации, сохраняется, как и с 11-битной адресацией.

3.3.2 Признак удаленного запроса

Он служит для маркировки команды запроса значения параметра («1») или кадра, содержащего значение параметра («0»).

Запрос на чтение параметра должен содержать признак удаленного запроса в «1», при этом поле данных пустое или содержит индекс запрошенного параметра. При ответе или при команде на запись значения параметра признак удаленного запроса устанавливается в «0», а поле данных содержит данные.

3.3.3 Локальный идентификатор параметра

Локальный идентификатор параметра представляет из себя Hash-свертку имени передаваемого параметра.

3.3.4 Контрольная сумма

Вычисляется согласно алгоритма, указанного в п.4.5. В контрольную сумму входят все полученные от передающего устройства вышеперечисленные биты информации (в т.ч. адрес, длина и поле данных) и не входят маркеры начала и конца пакета.

3.3.5 Блок данных канального уровня

Блок данных содержит данные, передаваемые следующим уровням.

3.3.6 Структура квитанции

Если в принятом сообщении не обнаружены ошибки, то ведомом узлом (прибором) передается квитанция, аналогичная посылке, но с признаком удаленного запроса равным «0».

4 Уровень ПРЕДСТАВЛЕНИЯ протокола ОВЕН

4.1 Структура данных

Блок данных, принятый от канального уровня протокола, состоит из:

- 16-битового локального идентификатора параметра (HASH-кода)
- n-байт (от 0 до 15) поля данных

4.2 Локальный идентификатор параметра

Локальный идентификатор параметра в протоколе ОВЕН представляет из себя Hash-свертку имени передаваемых параметров.

4.3 Соглашение об именах параметров

Используем 4-х символьные имена пользовательских параметров в виде следующего набора разрешенных символов:

- 1) **26** букв английского алфавита (от "А" до "Z") без различия строчных и прописных букв
- 2) **10** арабских цифр (от "0" до "9")

Примечание: На 7-ми сегментном индикаторе начертание цифры "5" совпадает с начертанием буквы "S"

- 3) Следующие специальные символы:

пробел
"_"
" "
"/"

Итого - набор из 40 символов. Каждый символ может иметь или не иметь "при себе" точку, таким образом, количество кодовых комбинаций удваивается, и в именах можно будет использовать и точки, не занимая "отдельного" символьного места.

Вышеозначенные символы кодируются (нумеруются) подряд идущими **ЧЕТНЫМИ** цифрами, начиная с 0000000b, причем сначала нумеруются символы-цифры ("0"..."9"), затем без перерыва в алфавитном (английский) порядке буквы (от "А" до "Z"), а затем специальные знаки "-", "_", "/" и, наконец, пробел.

Символам с точкой присваивается код символа без точки, увеличенный на 1 (нечетные коды). Пример:

"0"	-	0000000b
"0."	-	0000001b
"1"	-	0000010b
"1."	-	0000011b
...		
"a"="А"	-	0010100b
"a."="А."	-	0010101b

Таким образом, имена "Comp", "comp", "COMP" будут относиться к одному и тому же параметру, но имена "com.P" или "Co.mp" являются уже именами других параметров и никак не связаны с первым параметром.

Пробелы в имени возможны в рамках 4-х символов, но в конце имени.

Далее с помощью хеширования имя параметра превращается в его локальный идентификатор, который в совокупности с адресом узла образуют глобальный идентификатор параметра (или сообщения).


4.4 Преобразование из ASCII-кодов имени параметра в код символа

Преобразование из ASCII кодов (делается на PC перед hash-сверткой или внутри прибора):

```
char ch; //преобразуемый ASCII символ
char char_for_hash; //символ для обработки с помощью hash-
функции
int toupper(int c); //преобразование в символы верхнего
регистра
ch=toupper(ch); //преобразуем все в верхний регистр
switch (ch)
{
    case "0".."9": char_for_hash=ch-"0"; break;
    case "A".."Z": char_for_hash=ch-"A"+10; break;
    case "-":char_for_hash=10+26; break;
    case "_":char_for_hash=10+26+1; break;
    case "/":char_for_hash=10+26+2; break;
    case " ":char_for_hash=10+26+3; break;
    default: обработка_ошибки(); break;
}
char_for_hash*=2; //удвоить код символа
if (за_ним_следует_точка) char_for_hash+=1; //прибавить 1
```

Также можно воспользоваться готовой таблицей (однако следует понимать, что работа с таблицей происходит медленнее):

Код	Символ	Код	Символ	Код	Символ	Код	Символ
0	0	20	А	40	К	60	U
2	1	22	Ь	42	Л	62	v
4	2	24	С	44	м	64	w
6	3	26	д	46	п	66	х
8	4	28	Е	48	о	68	Y
10	5	30	Ф	50	Р	70	z
12	6	32	Г	52	q	72	-
14	7	34	Н	54	r	74	_(подч)
16	8	36	и	56	S	76	/
18	9	38	Ж	58	t	78	пробел

 При наличии точки после символа к его коду прибавляется единица.

4.5 Хеширование имен параметров и вычисление контрольной суммы сообщения

Для вычисления hash-функции и CRC-суммы используется один и тот же полином, а именно:

$$x^{16}+x^{15}+x^{11}+x^{10}+x^9+x^8+x^6+x^4+x^2+x^1+x^0.$$

Алгоритм вычисления hash-функции и контрольной суммы следующий:

```
unsigned Hash(char Byte, char nbit, unsigned CRC)
{
    for (int i = 0; i < nbit; i++, Byte <<= 1)
    {
```



```

        if ( ( Byte ^ (CRC>>8) ) & 0x80 )
        {
            CRC <<= 1;  CRC ^= 0x8F57;
        }
        else CRC <<= 1;
    }
    return CRC;
}

```

Обратите внимание на то, что «добавление» очередного байта к hash-свертке или к контрольной сумме начинается со старшего бита. Для представления символа сетевого имени параметра используется 7 бит, вследствие чего, при расчете hash-свертки имени, старшим битом считается **именно 7-й, а не 8-й**, всегда равный нулю бит. 8-й бит символа в вычислении hash-свертки имени никогда не участвует:

```

crc = Hash(command [3] << 1, 7,
           Hash(command [2] << 1, 7,
           Hash(command [1] << 1, 7,
           Hash(command [0] << 1, 7, 0)))));

```

4.6 Поле данных уровня представления

Формат представления поля данных описан в п.5.1. Структура поля данных зависит от формата и конкретного параметра.

4.7 Структура квитанции

Если в сообщении, принятом ведомым узлом, не обнаружены ошибки, передается квитанция, аналогичная посылке, но с признаком удаленного запроса равным «0».

5 Уровень ПРИЛОЖЕНИЯ протокола ОВЕН

5.1 Формат представления поля данных

Передаваемая в 15 байтах поля данных информация содержит различные типы данных. Это позволяет использовать в сети информацию и от достаточно простых приборов, которые оперируют лишь ограниченным количеством типов данных. Устройства приемники данных должны уметь преобразовывать данные принимаемых типов в требуемый внутренний формат в случае их несовпадения. Информация о типе данных источника вводится в приемник на этапе конфигурирования, одновременно с информацией об идентификаторе принимаемого параметра.

Базовые типы:

- Формат числа с плавающей точкой
- Знаковое число с односторонней десятичной точкой в двоичном виде
- Знаковое число с односторонней десятичной точкой в двоично-десятичном виде
- Часовой формат данных в двоичном виде
- Часовой формат данных в двоично-десятичном виде
- Нетипизированные целые числа в двоичном виде
- Нетипизированные целые числа в двоично-десятичном виде
- Строковый тип

Модификаторы базовых типов:

- 1) Временной модификатор (+2 байта времени в младших байтах поля данных, но перед (в более старшем байте) индексом, если наличествуют оба модификатора)
- 2) Индексный модификатор (+2 байт индекса в младших байтах поля данных)

Для удобства базовым типам присвоены номера, которые должен использовать каждый прибор в сети при задании формата данных фильтруемого параметра. С учетом того, что существуют также два модификатора базовых типов (времени и индексов) получится всего $8 \cdot 4 = 32$ формата.

5.1.1 Формат числа с плавающей точкой

Формат числа с плавающей точкой - это просто стандартное 4-байтовое плавающее число в IEEE-формате, или усеченное 3-х байтовое (PIC-формат) - то же, что и IEEE, но с отброшенным младшим байтом мантиссы. Если известно, что формат плавающий, то усеченный и полный IEEE-форматы можно различить по битам размера информационного поля. Внутренние поля представлены только в двоичном, но не в двоично-десятичном виде.

5.1.2 Знаковое число с односторонней десятичной точкой

Имеет знак (S), десятичный порядок (Exponent), выраженный БЕЗЗНАКОВЫМ числом и указывающий положение ДЕСЯТИЧНОЙ точки (0 для целого числа, 1 для числа с десятичными долями и т.п.) и ненормализованную мантиссу (Mantissa).

S - старший бит
Exponent - 3 следующих бита
Mantissa - остальные биты (длина мантиссы определяется размером поля данных)

Суммарная длина должна составлять целое число байтов

Значение числа можно вычислить, как

$$(-1)^S * 10^{(-Exponent)} * Mantissa$$

Мантисса может выражаться как в двоичном, так и в двоично-десятичном виде.

Например:

-10.38 выражается как 0xA01038 при двоично-десятичном представлении мантиссы
или как 0xA40E, то есть на байт короче, при ее двоичном представлении.

5.1.3 Часовой формат данных

Шаблон для представления числа в часовом формате является последовательность блоков данных, содержащих следующую информацию о времени (начиная со старшего байта):

Y..Y - год (*от 00 до 99 кодируются только две младшие цифры года)
M..M - месяц (*от 0 до 11)
D..D - день (*от 0 до 30)
h..h - часы (*от 0 до 23)
m..m - минуты (*от 0 до 59)
s..s - секунды (*от 0 до 59)
c..c - сотые доли секунд (*от 0 до 99)



* Ограничения не относятся к первому блоку посылки, который к тому же может иметь длину более одного байта.

Каждый из блоков может состоять из одного (для любого блока в посылке) или нескольких байт (для самого старшего блока в посылке). Формат числа в часовом формате предусматривает передачу любой "сплошной" части этого шаблона не нарушающей границ байтов. Число в данном формате (от "старших" цифр к младшим) представляется в виде байтов шаблона, начиная с произвольного байта и кончая произвольным байтом шаблона, вслед за которыми идет байт, старшая тетрада которого содержит количество (Number) передаваемых блоков в посылке, а младшая - цену младшего блока (Price).

Количество передаваемых блоков может быть от 1 до 7.

Цена младшего разряда может принимать следующие значения:

0 если младшим байтом являются сотые доли секунды
1 если младшим байтом являются секунды

- 2 если младшим байтом являются минуты
- 3 если младшим байтом являются часы
- 4 если младшим байтом являются дни
- 5 если младшим байтом являются месяцы
- 6 если младшим байтом являются годы

Пример 1:

Посылка вида **hh|mm|ss|0x31** - будет иметь размер поля данных Size= 4 байта. При приеме такой посылки младшая тетрада байта под номером 0 (в данном случае 1) указывает на то, что в 1-ом байте находятся секунды, в следующем (если таковой есть) минуты и затем часы. hh (поле часов) является последним полем, на что указывает количество блоков данных - 3 в старшей тетраде байта под номером 0. Длина этого последнего поля в принципе может быть произвольной, но в данном случае она, как и у остальных полей составляет один байт. Длина последнего поля вычисляется по формуле Size-Number=4-3=1.

Пример 2:

Посылка вида **mmmm|ss|0x21** - также будет иметь размер поля данных Size= 4 байта. При приеме такой посылки младшая тетрада байта под номером 0 (в данном случае 1) указывает на то, что в 1-ом байте находятся секунды). Второе поле mm (поле минут) является последним полем, на что указывает количество блоков данных - 2 в старшей тетраде байта под номером 0. Длина этого последнего поля в данном случае составляет 2 байта. Длина последнего поля вычисляется по формуле Size-Number=4-2=2.

Каждое из полей шаблона может быть выражено как в двоичном, так и в двоично-десятичном формате, при этом общая длина данных от этого не зависит.

5.1.4 Нетипизированные целые числа или наборы битов

Если нетипизированные целые задаются как тип данных чужого параметра, фильтруемого фильтром, то в зависимости от длины поля данных могут соотноситься с типами:

```
unsigned char;      //(1 байт)
unsigned short int; //(2 байта)
unsigned long int;  //(4 байта),
```

но могут иметь и любой нестандартный размер в пределах от 1 до 15 байтов.

Выделенных полей (типа знака, порядка и т.п.) не имеют, хотя фактически, как мы увидим в дальнейшем, такие поля могут в них быть. Байты могут кодироваться как в двоичном, так и в двоично-десятичном форматах.

Нетипизированные целые числа могут применяться также для обмена по сети произвольными данными при "двухточечном" соединении (в том числе, данными типа float), если их типы одинаковы или заведомо известны на приемной и передающей сторонах, и если эти данные никогда не используются третьими лицами. Примером может служить случай, когда master-устройство устанавливает конфигурационные параметры в slave-устройстве.

Таким образом, нетипизированные целые числа могут представлять и плавающие числа, и данные с фиксированной запятой, и даже знаковые числа, в двоичном случае трактуемые как числа в дополнительном коде, а в двоично-десятичном случае - как число со знаком, заданным в старшей тетраде (<0xa - все число считается положительным, а цифра засчитывается как цифра числа; >=0xa - тетрада считается отрицательным знаком всего числа и как цифра не засчитывается, т.е. считается равной 0).

5.1.5 Строковый тип

Используется для передачи строк или данных непредусмотренных выше типов с помощью конвертирования их в строку.

Строка может состоять из от 1 до 15 ASCII-символов. Несмотря на то, что она сама составлена из ASCII символов, каждый символ (байт) строки и в этом случае разбивается на

тетрады нижним уровнем и представляется 2-мя ASCII символами (из разрешенного набора "G".."V").

☞ Строка передается байтом с наибольшим индексом (старшим байтом) вперед.

5.1.6 Составной тип данных с информацией о времени

Такие параметры, как температура, влажность, давление и т.п. как правило, должны сопровождаться и временем, сообщаящим о моменте, когда данное измерение было сделано. В этом случае прибор, имеющий в своем составе ПИД-регулятор, прослушивая сеть, может перехватывать данные параметры и осуществлять полноценное регулирование, в том числе и по интегральному и дифференциальному каналам, не внося дополнительных ошибок, связанных с несоответствием действительного времени измерения, которое определяется внутренней синхронизацией прибора-источника, и фактическим временем передачи параметра по сети, которое определяется моментом запроса параметра master'ом. Причем синхронизации времен разных приборов в сети не потребуется, поскольку для преследуемых нами целей используется только относительное время, т.е. разность между временем предыдущего и текущего измерений. Важно только иметь единую единицу измерения времени. В качестве такой единицы выбрана 0.01 сек, поскольку, с одной стороны, это время достаточно мало, чтобы обеспечить приемлемую точность расчета производной и интеграла контролируемого сигнала, а с другой стороны оно достаточно легко получается при обычно используемых частотах регенерации индикации (обычно "круглые" сотни Гц).

Если по сети передавать 2-байтовое циркулярное значение времени, то этого окажется достаточным для нормальной работы ПИД-регулятора с периодичностью получения результатов измерения до 655,36 секунд = 10,922(6) минут, что представляется вполне приемлемым.

Все перечисленные выше типы данных, кроме, может быть данных в часовом формате, для которых это не имеет смысла, могут иметь и варианты с 2-х байтовым временем с 0,01-секундным разрешением. При этом эти 2 байта являются младшими и всегда выражены в двоичном виде, даже если базовый тип выражен в двоично-десятичном.

5.1.7 Составной тип данных с дополнительным полем индексации

Для передачи индекса конфигурационных параметров (см. п. 5.2 и 5.3.1) используются составной тип данных с дополнительным полем индекса, равным 2-м байтам. Такие комбинированные типы данных могут быть образованы от любого из вышеперечисленных базовых типов с добавлением к ним со стороны младшего байта дополнительного одномерного индекса.

5.2 Типы параметров

Параметры приборов разделяются на два типа: конфигурационные и оперативные.

Конфигурационные параметры – это параметры, задающиеся пользователем и определяющие настройки прибора. Значения конфигурационных параметров, как правило, сохраняются в энергонезависимой памяти и не пропадают при выключении питания прибора.

Оперативные параметры переносят информацию о текущем состоянии прибора или объекта регулирования. В оперативных параметрах передаются измеренные или вычисленные значения, выходные мощности регуляторов, номера запущенных в данный момент программ, текущие состояния выходных элементов и т.д.

Также к оперативным параметрам относятся команды, каким-то образом управляющие прибором.

5.3 Индексирование параметров разных типов

Индексация параметров необходима для разделения одинаковых параметров, относящихся разным однотипным элементам прибора. Например, в приборе имеется два регулятора. Конфигурационный параметр "Уставка регулятора" будет один, например, с именем SP. А

отличить, к какому регулятору эта уставка относится можно будет по индексу: параметр SP с индексом 0 будет относиться к первому регулятору, SP с индексом 1 – ко второму регулятору.

Индексирование параметров осуществляется различными способами для различных типов параметров. Оперативные и конфигурационные параметры имеют различные способы индексации.

5.3.1 Индексация конфигурационных параметров

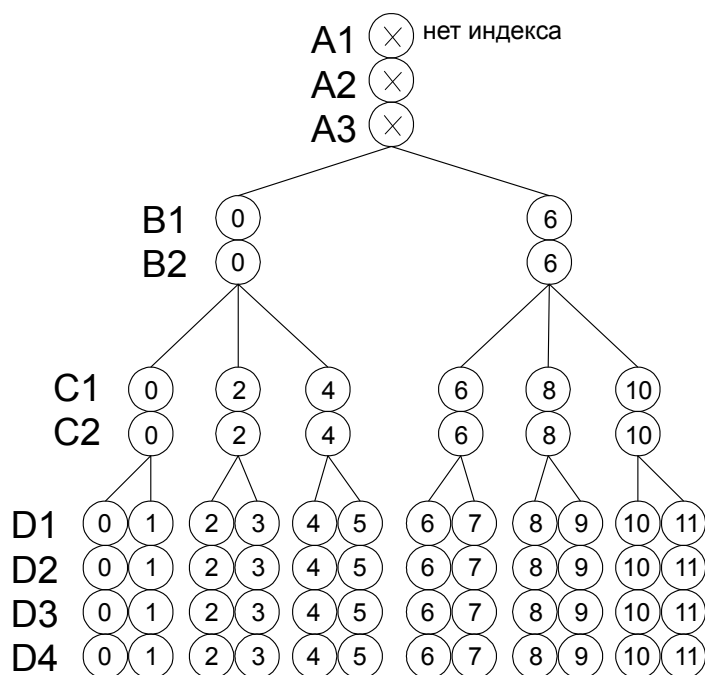
Конфигурационные параметры могут иметь индекс или не иметь такового. Индекс не требуется, если конфигурационный параметр представлен в приборе в единственном числе.

Индекс конфигурационного параметра передается в поле данных в 2-х последних байтах. Допускается только один индекс, по этому при необходимости индексировать сложную структуру данных необходимо организовать ее в виде дерева, пример которого рассмотрен ниже.

Пример дерева индексации. На рисунке показано "дерево" параметров, входящих в общую группу индексации.

Каждый круг соответствует отдельному параметру, но не все параметры имеют собственные уникальные имена. Параметры, расположенные на одном горизонтальном уровне (но на разных ветвях) имеют одно имя (являются тезками), а различаются индексом "ветви, на которой сидят". Будем считать, что это параметры группы "программы технолога". Тогда параметры "корня" (A1, A2, A3) - это параметры общие для всей группы, т.е. для всех объектов, всех программ технолога и всех шагов этих программ. Параметры следующего уровня – это параметры, относящиеся к объектам (B1, B2). В нашем примере имеются 2 объекта и, соответственно, дерево разделяется на 2 ветви. Далее идут параметры для каждой программы технолога (C1, C2). В примере у каждого объекта есть по 3 независимые программы технолога и соответственно дерево расходится на 3 ветви. Последние параметры – это параметры шагов программы. В примере у программ 2 шага и дерево опять ветвиться надвое.

Индексы параметров указаны в центре кружков. Индекс верхней ветви (предка) равен индексу самого левого элемента нижней ветви (потомка). Корневые параметры дерева (в данном примере A1, A2, A3) не имеют индексов (поле индекса передавать не надо).



В данном примере группа параметров имеет 3 уровня вложенности. В приборах, построенных на основе AVR-процессоров, допускается иметь до 8 таких уровней вложенности. В принципе дерево может содержать произвольное количество ветвей на каждом из уровней. Корень может отсутствовать, тогда дерево может начинаться сразу с ветвления 1-го уровня. Могут отсутствовать и несколько первых уровней, тогда "дерево" превращается в несколько

одинаковых деревьев, что однако не мешает индексировать их точно таким же образом, как и в случае полного дерева.

Наличие индексов конфигурационных параметров, а также структура их построения определяется для каждого прибора индивидуально и указывается в списке параметров прибора.

5.3.2 Индексация оперативных параметров

Индекс оперативного параметра передается в адресе прибора, в поле данных индекс не ставится.

Пример. Есть восьмиканальный измеритель температуры. И его адрес в сети RS-485 (Базовый адрес) равен 16. Тогда измеренную температуру с первого канала прибор должен передавать с адреса 16, температуру второго канала с адреса 17, третьего – 18 и т.д. до 23-го адреса. Т.е. восьмиканальный прибор представляется как 8 независимых одноканальных приборов, каждый со своим сетевым адресом. И под него резервируются 8 адресов (независимо от длины адреса: 8 или 11 бит).

Индекс оперативного параметра должен быть всегда одномерный, линейный. Сложная индексация не допускается. Если необходимо передать оперативные данные сложной структуры – то надо разбить ее на ряд простых подструктур с одномерной индексацией и передавать эти структуры с разными именами.

5.4 Передача ошибок

5.4.1 Ошибки и исключительные ситуации

В приборе выделяется две группы ошибок: Сетевые ошибки и Исключительные ситуации при передаче оперативных параметров.

Сетевая ошибка – это ошибка, возникающая при передаче или приеме данных по сети RS-485. Сетевые ошибки происходят при работе с конфигурационными и оперативными параметрами и могут возникнуть при посылке, приеме или обработке пакета на любой из сторон. Также как сетевая ошибка могут быть интерпретированы ошибки, связанные с невозможностью прибора дать ответ на посылаемый запрос, например при сбое памяти, хранящей запрашиваемый параметр.

Исключительная ситуация – это ситуация, возникающая при невозможности прибора корректно ответить на запрос оперативного параметра или на посылку ему управляющей команды. Эта ситуация может возникнуть в целом ряде случаев: из-за обрыва датчика, из-за того, что измерение еще не успело произвестись, из-за попытки вычислить квадратный корень из отрицательного числа, при попытке выдать сигнал тока с дискретного выходного элемента и т.д. При этом сетевой обмен прибор вполне может осуществлять, только корректно обработать запрос или команду не может.

Список оперативных параметров и присущих им исключительных ситуаций содержится в описании на каждый конкретный прибор.

Сетевые ошибки и исключительные ситуации передаются и обрабатываются **по разному**.

5.4.2 Передача сетевых ошибок

При наличии сетевых ошибок ведомый узел передает на месте hash-кода параметра – hash-код параметра n.Err, а в поле данных:

Старший байт – код общей ошибки (см. Приложение 1).

Два последующих (младших) байта – hash-код параметра посылки.

5.4.3 Передача исключительной ситуации

Передача при исключительной ситуации не отличается от обычной передаче пакета. Ни адрес, ни Hash-код параметра не меняются. Изменение происходят только в поле данных.

При передаче сообщения об исключительной ситуации в поле данных надо трансформировать так, чтобы передать вместо нормального сообщения такое, чтобы оно:

– либо было такой длины, чтобы невозможно было его представить как нормальный пакет
 – либо передать слишком большое или слишком малое число, такое, какого не бывает при нормальной работе прибора.

Первый тип маскировки подходит для чисел с плавающей точкой, а второй более подходит для целочисленных данных.

Далее, если пакет идентифицируется как несущий сообщение об исключительной ситуации, то он в любом случае представляется как набор байт (или как целое число) с кодом ошибки. Для контроля в первом байте этого набора первые 4 бита должны быть равными единице. Если это условие не выполняется – значит имеет место сетевая ошибка, которая должна обрабатываться отдельно.

Более подробно алгоритм обработки исключительной ситуации дан в **Приложении 2**.

6 Сетевые параметры протокола ОВЕН, Атрибуты параметров.

На этом уровне протокола описываются обязательные параметры и команды, необходимые для работы прибора.

6.1 Общие и Сетевые параметры протокола (обязательные)

Имя сетевое	Hash-код	Тип	Назначение	Значения
dev	D681	ASCII	Название прибора	до 8 символов
ver	2D5B	ASCII	Версия прошивки прибора	до 8 символов
bPS	B760	int	Скорость обмена (кбод)	0 =2,4 kbps; 1= 4,8 kbps; 2= 9,6 kbps; 3=14,4 kbps; 4=19,2 kbps; 5=28,8 kbps; 6=38,4 kbps; 7=57,6 kbps; 8=115,2 kbps
Len	523F	int	Длина слова данных (бит)	0= 7 бит 1= 8 бит
PrtY	E8C4	int	Состояние бита четности в посылке	0= No 1= EuEn 2= Odd
Sbit	B72E	int	Количество стоп-битов в посылке	0 =1 бит 1 = 2 бита
A.Len	1ED2	int	Длина сетевого адреса (бит)	0 = 8 бит 1 = 11бит
Addr	9F62	int	Базовый адрес прибора	0 ...2047
n.Err	0233	int	Код сетевой ошибки при последнем обращении к прибору.	См. таблицу №3
rS.dL	1e25	int	Задержка ответа от прибора по RS485 (мс)	0...45

6.2 Команды смены сетевых настроек прибора

Для смены сетевых настроек прибора необходимо послать в прибор специальный параметр-команду Aply.

Имя сетев.	Hash-код	Назначение	Комментарии
APLY	8403	Команда перехода на новые сетевые настройки.	Посылка: бит ASK = 0, Size = 0 Квитанция: бит ASK = 0, Size = 0 или нет квитанции

Прибор должен фактически иметь две таблицы сетевых настроек: текущую (рабочую) и конфигурационную. При записи новых сетевых параметров они попадают в энергонезависимую

память, в таблицу конфигурационных сетевых настроек. А по команде Arly прибор должен копировать конфигурационную таблицу параметров в рабочую и переходить на работу с новыми сетевыми настройками.

6.3 Атрибуты параметров. Смена атрибутов

Каждый параметр, помимо своего значения может иметь один или два атрибута:

- атрибут "Редактирования", имеющий значения "Редактируемый", "Не редактируемый".
- атрибут "Владелец", имеющий значения "Завод" и "Пользователь".

Атрибут редактирования защищает значение параметра от несанкционированного редактирования. Если установлен атрибут "Не редактируемый", то изменить значение параметра не удастся. Пользователь может изменить этот атрибут по своему усмотрению.

Атрибут "Владелец" защищает от изменения атрибут "Редактирования". Атрибут устанавливается предприятием-изготовителем специальными методами и призван защитить некоторые параметры и их атрибуты от редактирования пользователем.

Для параметров допускается делать только один атрибут "Редактирования" или не делать атрибутов вовсе.

Для смены атрибута редактирования существует команда Attr.

Имя сетев.	Hash-код	Назначение	Комментарии
Attr	749F	Команда чтения/записи атрибута редактирования	<p>Посылка записи атрибутов: бит ASK = 0 SIZE = 3 или 5</p> <p>Информационная часть: В 0-1-м байтах hash-код параметра, которому принадлежат атрибуты. В 2-3-м байтах индекс параметра (если есть). 4-й байт – атрибут: 0-й бит – атрибут редактирования: 1 = редактируемый параметр 0 = не редактируемый параметр</p> <p>Квитанция записи бит ASK = 0 SIZE = 3 или 5 Информационная часть аналогична посылки записи.</p>

Приложение 1

Таблица №1. Список сетевых ошибок

Наименование ошибки (желательное название при написании Progr.)	Значение ошибки, передаваемое по сети	Характеристика
Определение констант ошибок приема		
OK	0	Безошибочный прием кадра
Ошибки записи параметров и атрибутов функцией modifc		
PDOT	2	Задано положение точки, превышающее 3
EROM	3	Попытка модификации ROM-параметра
ESTR	4	Не целое число при записи индекса строки или времени
EDOT	5	Неверно задано положение точки (при фиксированной точке)
ERNG	6	Значение мантиссы превышает ограничения дескриптора
Ошибки записи атрибутов функциями modAllPermis() и modEditPermis()		
EOWNER	7	Несанкционированная попытка редактирования Атрибутов (попытка изменить атрибут пользователем, когда он не является хозяином параметра).
EPERM	8	У запрошенного параметра отсутствуют признаки
Стандартные ошибки, присущие протоколу обмена		
AFE	0x21	Аппаратная ошибка кадрирования
B8E	0x22	Ошибка в 8-ом бите посылки
B9E	0x23	Ошибка в 9-ом бите посылки
SBE	0x24	Ошибка приема стоп-байта (стоп пришел не вовремя)
OVB	0x25	Ошибка переполнения буфера
ERS	0x26	Принят недопустимый символ
CRCE	0x27	Неверная контрольная сумма кадра
EDESC	0x28	Не найден дескриптор
NFNC	0x29	Не найдена сетевая функция, хотя дескриптор найден!!! В нормальном приборе эта ошибка встречаться НЕ ДОЛЖНА!!!
Стандартные ошибки, общие для всех модулей		
EDGT	0x30	Мантисса двоично-десятичного параметра содержит ошибку
SZE	0x31	Размер поля данных не соответствует ожидаемому
EASK	0x32	Значение бита запроса не соответствует ожидаемому
EACC	0x33	Редактирование параметра запрещено индивидуальным атрибутом
IDXOVF	0x34	Недопустимо большой линейный индекс
IDXLIM	0x35	Индекс параметра превышает ограничитель индекса
EXTROM	0x36	Индекс параметра превышает ограничитель индекса
RESERVED	0x37	Данный код не используется
ГРУППА из 8 кодов ошибок в диапазоне: LEVGRATT...LEVGRATT+L...LEVGRATT+7. Ошибка: "Запись запрещена групповым атрибутом уровня L"		
LEVGRATT	0x38	Запрещающий групповой атрибут находится на уровне 0 (в корне)
LEVGRATT1	0x39	Запрещающий групповой атрибут находится на уровне 1
LEVGRATT2	0x3A	Запрещающий групповой атрибут находится на уровне 2
LEVGRATT3	0x3B	Запрещающий групповой атрибут находится на уровне 3
LEVGRATT4	0x3C	Запрещающий групповой атрибут находится на уровне 4
LEVGRATT5	0x3D	Запрещающий групповой атрибут находится на уровне 5
LEVGRATT6	0x3E	Запрещающий групповой атрибут находится на уровне 6
LEVGRATT7	0x3F	Запрещающий групповой атрибут находится на уровне 7
Состояния COMMON-сегмента		
__LOCKSEG	0x41	выполняется другая задача (Сегмент COMMON занят)
__FREESEG	0x42	задача еще не запущена(Сегмент COMMON свободен)
__READYSEG	0x43	запрошенная задача уже выполняется
__DEBUGSEG	0x44	Программе неизвестна запрошенная функция
__NOWHATCOM	0x45	В программе стоит заглушка функции WhatCOMState()
__NORUNCOM	0x46	В программе стоит заглушка функции RunCOMTask()
	0x47	Недопустимое сочетание значений параметров

		(Изменение параметра было запрещено функцией Valid)
	0x48	Ошибка при чтении EEPROM

Ошибки при редактировании графиков		
	0x49	Нарушена упорядоченность узлов X по возрастанию
	0x4A	Попытка записи X при ненулевом числе узлов графика
	0x4B	Ошибка выполнения функции PrevWriteActions()
Ошибки мостов и ретрансляторов.		
GATE_OVR	0x50	Переполнение буфера моста или ретранслятора
GATE_DERR	0x51	Превышение тайм-аута ответа, потеря пакета в дочерней сети (сети, в которую ретранслируется пакет)
GATE_NONET	0x52	Запрошенная дочерняя подсеть не доступна (в случае ретрансляции в одну из нескольких дочерних подсетей)
GATE_MERR	0x53	Ответ из дочерней сети не может быть ретранслирован в материнскую сеть.

Дополнительная информация для справки

7-УРОВНЕВАЯ МОДЕЛЬ ПРОТОКОЛОВ

Уровень приложения (Application) - интерфейс с прикладными процессами.

Уровень представления (Presentation) - согласование представления (форматов, кодировок) данных прикладных процессов.

Сеансовый уровень (Session) - установление, поддержка и закрытие логического сеанса связи между удаленными процессами.

Транспортный уровень (Transport) - обеспечение безошибочного сквозного обмена потоками данных между процессами во время сеанса.

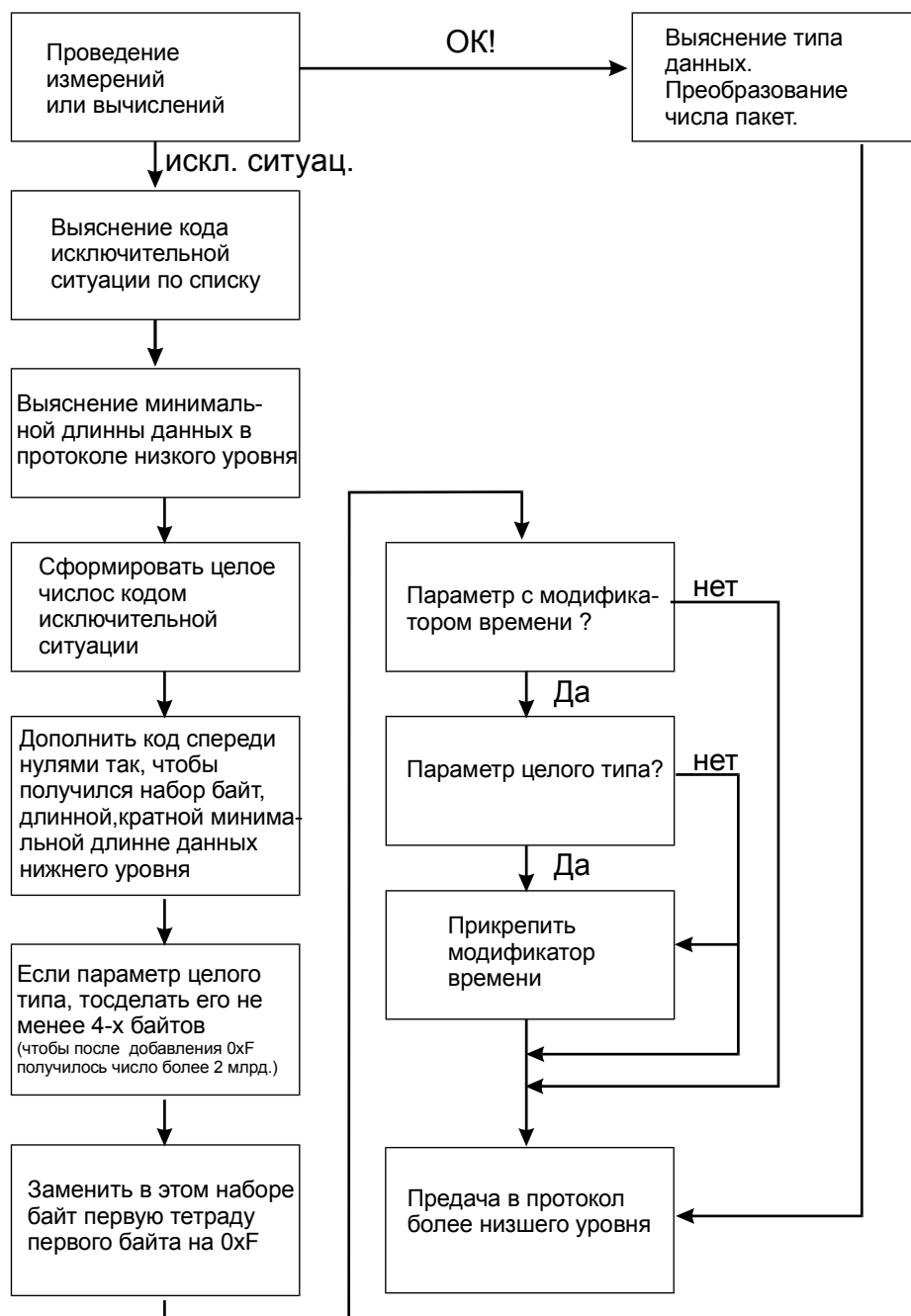
Сетевой уровень (Network) - фрагментация и сборка передаваемых транспортным уровнем данных, маршрутизация и продвижение их по сети от компьютера-отправителя к компьютеру-получателю.

Канальный уровень (Data Link) - управление каналом передачи данных, управление доступом к среде передачи, передача данных по каналу, обнаружение ошибок в канале и их коррекция.

Физический уровень (Physical) - физический интерфейс с каналом передачи данных, представление данных в виде физических сигналов и их кодирование (модуляция).

Приложение 2

Алгоритм обработки исключительной ситуации при **передаче прибором**



Пример передачи кода исключительной ситуации 0xE для параметра типа float:

Для протокола **ОВЕН** (мин.дл.данных = 16байт) пакет будет: 0xFE

Для протокола **ModBus** (мин.дл.данных=2байт) пакет будет: 0xF0 0E

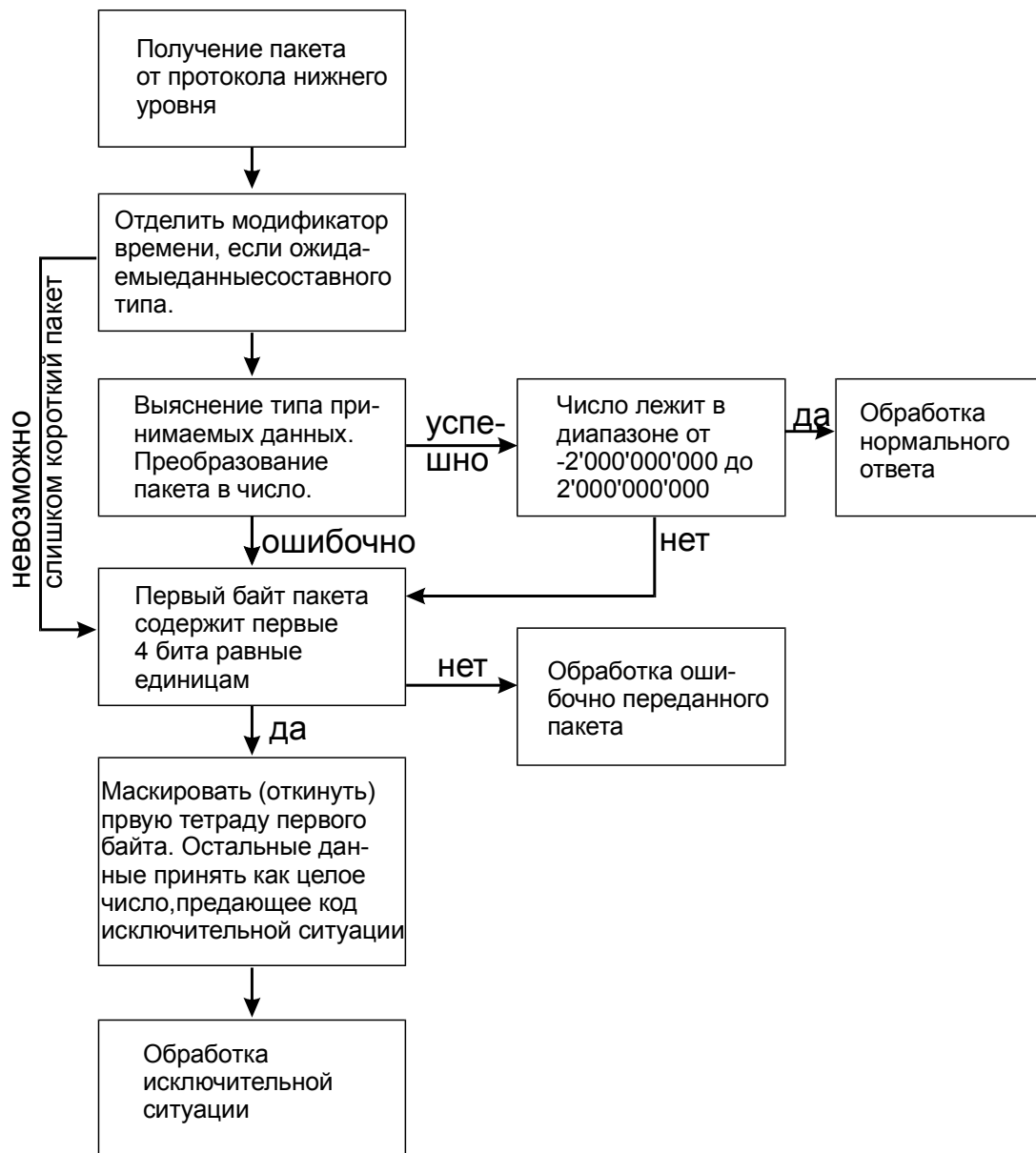
Для протокола с мин. длиной **8 байт** пакет будет: 0xF0 00 00 00 00 00 00 0E

Для кода **искл. ситуации** 0x15E пакет будет

(при минимальной длине = 4 байт): 0xF0 00 01 5E

При передаче данных составного типа (например, с модификатором времени) для целых чисел необходимо передавать полные составные данные, а для чисел с плавающей точкой – можно передавать или не передавать составные данные.

Обработка исключительной ситуации при приеме.



Приложение 3

Типовое обозначение форматов представления данных, принятых для приборов ОВЕН.

№	Сокращенная мнемоника	Характеристика
1	DEC_dot <i>i</i>	Параметр имеет фиксированную (не подлежащую редактированию) десятичную точку в <i>i</i> -й позиции (см. п. 5.1.2). Диапазон изменения <i>i</i> от 0 до 3.
2	STR	Параметр, отображаемый не в виде числа, а как последовательность символов. Его значение соответствует номеру строки в таблице. (см. п. 5.1.4)
3	STORED_DOT	Параметр имеет не фиксированную десятичную точку, которая хранится вместе с параметром в памяти ЕЕПРОМ, что увеличивает длину параметра на 2 бита. (см. п. 5.1.2)
4	INCR	Параметр в формате U отображается на дисплее увеличенным на 1. (см. п. 5.1.4)
5	U UB	Беззнаковое целое. (см. п. 5.1.4)
6	IEEE Float32	В формате IEEE (см. п. 5.1.1)
7	PIC Float24	В формате PIC (см. п. 5.1.1)
8	Time	Время измерения в 2-х байтах. (см. п. 5.1.6)
9	ASCII T	Строка символов из ASCII кодов (WIN1251) (см. п. 5.1.5).
10	I	Знаковое целое. (см. п. 5.1.4)

Примечание:

Для некоторых приборов возможно другое обозначения форматов представления данных и введение иных форматов представления данных. Иные форматы представления данных указываются дополнительно в списке конфигурационных параметров приборов.