



СЧЕТЧИКИ ИМПУЛЬСОВ ПРОВОДНЫЕ УНИВЕРСАЛЬНЫЕ (СИПУ)

Протокол обмена

Оглавление

Система команд	2
Порядок хранения и передачи байт данных	3
Алгоритм генерации CRC	4
Адресация регистров	5
Коды ошибок	6
Коды скоростей обмена	6
Назначение входа	6
Маски типов событий	6
Алгоритм получения архивных данных	6

Система команд

Командно-информационный обмен управляющего компьютера со счетчиком осуществляется в пакетном режиме по принципу “команда-ответ”. В качестве физической среды передачи информации используется канал RS485 со следующими параметрами:

- * Режим передачи – 8 бит без проверки на четность, 2 стоп-бита.
- * Скорость обмена – 9600.
- * Максимальная длина посылки 128 байта.
- * Порядок следования байтов: старший байт вперед.

Протокол обмена соответствует стандарту Modbus режим RTU.

Формат кадра сообщения

Адрес устройства	Команда	Данные	Контрольная сумма
1 Байт	1 Байт	N Байт	2 Байта

Для обмена данными со счётчиками используются следующие команды:
0x03 - Получение текущего значения одного или нескольких регистров,
0x10 - Установить новые значения одного или нескольких последовательных регистров.

03h - Получение текущего значения одного или нескольких регистров

Формат запроса:

Адрес устройства	Команда	Начальный адрес регистра	Количество регистров	Контрольная сумма
A	0x03	XX (2 Байта)	NN (2 Байта)	CC

Формат ответа при выполнении без ошибок:

Адрес устройства	Команда	Количество байт данных	Данные	Контрольная сумма
A	0x03	D (1 Байт)	D Байт	CC

10h - Установить новые значения нескольких последовательных регистров

Формат запроса:

Адрес устройства	Команда	Начальный адрес регистра	Количество регистров	Количество байт данных	Данные	КС
A	0x10	XX	NN	D (D = NN * 2)	D Байт	CC

Формат ответа при выполнении без ошибок:

Адрес устройства	Команда	Начальный адрес регистра	Количество регистров	Контрольная сумма
A	0x10	XX	NN	CC

Формат ответа при выполнении с ошибкой:

Адрес устройства	Команда	Код ошибки	Контрольная сумма
A	Установлен старший бит	E	CC

Порядок хранения и передачи байт данных

Для чтения и записи регистров в стандарте Modbus предусмотрены специальные функции, которые оперируют содержимым шестнадцатиразрядных регистров. Эти функции предполагают, что прибор хранит данные только типа шестнадцатиразрядное беззнаковое целое и ничего не «знают» о тех типах данных, которыми действительно представлены параметры прибора. Таким образом, получается, что в приборе данные хранятся в некоем исходном формате, а передаются по сети в виде набора шестнадцатиразрядных регистров. При передаче данных, чей размер в исходном формате превышает 16 бит (long, float, double и т.д.), используются несколько последовательных регистров. При этом младшие слова передаются в первую очередь, старшие - в последнюю. Т.о., для преобразования к порядку байт, естественному для платформы PC, требуется для каждого прочитанного/записываемого регистра изменить порядок байт.

Пример размещения данных для типа Int32

Регистр	Регистр A0		Регистр A1	
Порядок передачи	первый			последний
Байт	B1	B0(LSB)	B3(MSB)	B2

Алгоритм генерации CRC

1. 16-ти битовый регистр загружается числом FFFF hex (все 1), и используется далее как регистр CRC.
2. Первый байт сообщения складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с содержимым регистра CRC. Результат помещается в регистр CRC.
3. Регистр CRC сдвигается вправо(в направлении младшего бита) на 1 бит, старший бит заполняется 0.
4. (Если младший бит 0): Повторяется шаг 3 (сдвиг)
(Если младший бит 1): Делается операция ИСКЛЮЧАЮЩЕЕ ИЛИ регистра CRC и полиномиального числа A001 hex.
5. Шаги 3 и 4 повторяются восемь раз.
6. Повторяются шаги со 2 по 5 для следующего сообщения. Это повторяется до тех пор пока все байты сообщения не будут обработаны.
7. Финальное содержание регистра CRC и есть контрольная сумма.
Контрольная сумма передаётся младшим байтом вперёд.

// Расчет контрольной суммы кадра Modbus RTU

```
int modbus_crc( char* pdata, int len )
{
    int crc = 0xFFFF;

    while( len-- )
    {
        crc ^= *pdata++;
        for (int i=0; i<8; i++)
        {
            if (crc & 1)
                crc = (crc>>1)^0xA001;
            else
                crc >>= 1;
        }
    }

    return crc;
}
```

Табл.1 Адресация регистров

Адрес	Тип данных	Назначение	Чтен	Зап
0x0000	Int32	Серийный номер	✓	✓
0x0002	Int16	Версия ПО	✓	
0x0003	Int16	Идентификатор ПО	✓	
0x0004	Int16	Номер сборки ПО	✓	
0x0005	Int8	Сетевой адрес	✓	✓
0x0006	Int8	Код скорости обмена	✓	✓
0x0007	Int8	День месяца сохранения данных в журнал	✓	✓
0x0008	Int32	Текущее время в формате UTC	✓	✓
0x000A	Int16	Регистр состояний	✓	
0x000B	Int16	Командный регистр		✓
0x0100	Struct	Структура настроек канала 1	✓	✓
0x0200	Struct	Структура настроек канала 2	✓	✓
0x0300	Struct	Структура настроек канала 3	✓	✓
0x0400	Struct	Структура настроек канала 4	✓	✓
0x0500	Struct	Структура настроек канала 5	✓	✓
0x0600	Struct	Структура настроек канала 6	✓	✓
0x0700	Struct	Структура настроек канала 7	✓	✓
0x0800	Struct	Структура настроек канала 8	✓	✓
0x0900	Struct	Структура настроек канала 9	✓	✓
0x0A00	Struct	Структура настроек канала 10	✓	✓
0x0B00	Struct	Структура настроек канала 11	✓	✓
0x0C00	Struct	Структура настроек канала 12	✓	✓
0x0D00	Struct	Структура настроек канала 13	✓	✓
0x0E00	Struct	Структура настроек канала 14	✓	✓
0x0F00	Struct	Структура настроек канала 15	✓	✓
0x1000	Struct	Структура настроек канала 16	✓	✓
0x2000	Int32[N]	Показания по каналам в импульсах	✓	✓
0x2050	Float32[N]	Показания по каналам вычисленное	✓	✓
0x20A0	Int32	Состояния входов	✓	
0x2100	Int16	Счётчик несчитанных часовых записей в журнал	✓	✓
0x2101	Int16	Счётчик несчитанных событий	✓	✓
0x2102	Int32	Журнальное время в формате UTC	✓	✓
0x2110	Float32[N]	Показания по часам	✓	
0x2150	Float32[N]	Показания по месяцам	✓	
0x2200	Struct	Структура описания события	✓	

Табл. 2 Структура настроек канала

Смещение	Тип данных	Назначение
0	Int16	Код производителя
1	Int32	Серийный номер устройства
3	Int8	Версия
4	Int8	Тип устройства
5	Int16	DIF
6	Int16	VIF
7	Int8	Назначение входа
8	Float32	Вес импульса
10	Int16	Минимальная длительность импульса (мс)

Табл. 3 Структура описания события

Смещение	Тип данных	Назначение
0	Int32	Время в формате UTC
2	Int16	Тип события
3	Int32	Состояние входов
5	Float32[N]	Показания по каналам вычисленное

N – количество каналов

Коды ошибок

- 0x01 – Неизвестная команда;
- 0x02 – Неизвестный адрес регистра;
- 0x03 – Неверное значение параметра;
- 0x04 – Переполнение буфера данных;
- 0x05 – Запись в журнале отсутствует.

Версия ПО

Версия встроенного программного обеспечения определяет количество каналов у счётчика:

- 0x0110 – 2 канала,
- 0x0100 – 4 канала,
- 0x0120 – 10 каналов,
- 0x0130 – 16 каналов.

Коды скоростей обмена

- 0 -> 1200
- 1 -> 2400
- 2 -> 4800
- 3 -> 9600
- 4 -> 19200*
- 5 -> 38400*
- 6 -> 57600*
- 7 -> 115200*

* только для счётчиков без гальванической изоляции интерфейса.

Регистр состояний

- 0x0000 – нет ошибок,
- 0x0001 – напряжение между интерфейсными и счётными входами более 250В.

Командный регистр

При записи в этот регистр, счётчик выполняет следующие действия:

- 0x0001 – стирание часового журнала.

Тип устройства

Тип подключаемого счётчика кодируется согласно стандарту mBus:

- 02 – электричество,
- 03 – газ,
- 04 – теплосчётчик,
- 06 – горячая вода,
- 07 – вода,
- 16 – холодная вода.

Поле DIF

Поле DIF используется для указания типа данных и разбиения по тарифам:

- 0x0005 - число с плавающей точкой без указания тарифа
- 0x1085 - 1-й тариф,
- 0x2085 - 2-й тариф,
- 0x3085 - 3-й тариф,
- 0x4085 - энергия отпущенная,
- 0x5085 - энергия отпущенная 1-й тариф,
- 0x6085 - - - 2-й тариф,
- 0x7085 - - - 3-й тариф.

Поле VIF

Поле VIF определяет единицы измерения параметров:

- 0x0003 - 1 Втч,
- 0x0004 - 10 Втч,
- 0x0013 - 1 л,
- 0x0014 - 10л,
- 0x09FB - 1 ГДж,
- 0x0DFB - 1 Мкал

Назначение входа

- 0 – не подключен,
- 1 – импульсный счётный вход,
- 2 – импульсный аварийный вход,
- 3 – счётный вход намур,
- 4 – аварийный вход намур.

Маски типов событий

- 0x0001 – перезапуск устройства,
- 0x0002 – отключение внешнего питания,
- 0x0004 – включение внешнего питания,
- 0x0008 – изменение состояния входов защиты.

Алгоритм получения архивных данных.

Для получения архивных данных на определённую дату, необходимо в регистр «Журнальное время в формате UTC» записать эту дату. После этого можно считать данные из регистров «Показания по часам» и «Показания по месяцам», причём загрузка данных из журнала в память происходит в момент считывания первого регистра показаний. В случае отсутствия данных на указанную дату, при чтении первого регистра показаний будет возвращена ошибка 0x05. При удачном считывании часовых показаний Журнальное время автоматически будет увеличиваться на 1 час, а счётчик несчитанных записей в журнал уменьшаться на 1.

При считывании структуры описания событий, счётчик несчитанных событий будет уменьшаться на 1, а указатель будет смещаться на следующее событие. Т.о. при повторном считывании структуры описания события в регистрах будет содержаться уже следующее событие. Чтобы считать более ранние события, необходимо записать в счётчик событий число, на которое

нужно сместить назад указатель событий. Загрузка события из журнала в память происходит при чтении первого байта «Структуры описания события».

Размеры журналов в зависимости от количества каналов у счётчика указаны в таблице 4.

Табл. 4 Максимальное количество записей в журналы

Журнал	2 канала	4 канала	10 каналов	16 каналов
Часовой	4437	2662	1210	783
Месячный	85	51	23	15
Событий	455	315	146	110