

Протокол обмена с адаптером модема АМ-01 версия 1.4

При подключении модема АМ-01 начинает его инициализацию. АМ определяет наличие модема по сигналу DSR и CTS и отключенному DCD. При успешной инициализации переходит в режим ожидания звонка. При поступлении звонка проверяет время и если связь разрешена, выдает команду на ответ. При успешном соединении модем устанавливает сигнал DCD и АМ переходит в командный режим.

Сигналом перехода в командный режим служит высокий уровень на линии DCD. При подключении к компьютеру нуль-модемным кабелем с порта компьютера устанавливается сигнал DTR (который устанавливает DCD и DSR на АМ-01) затем RTS (который устанавливает CTS на АМ-01). АМ-01 устанавливает ответные сигналы. При обмене с АМ-01 программа или модем может управлять потоком с помощью RTS сигнала, а также должна следить о готовности прибора принимать данные по сигналу CTS.

Протокол обмена с АМ-01

Команды: 0x03 READ_REGISTER
0x10 WRITE_REGISTER
0x00 SYS_COMMAND

Адрес 0x15	Код команды 1 байт 0x00-0x7F	Номер регистра 1 байт	Номер Команды 1 байт	Длина Поля данных n 1 байт	Данные (Параметры) n байт	CRC 2 Байта
---------------	------------------------------------	--------------------------	-------------------------	----------------------------------	---------------------------------	----------------

Ответ:

Аналогичен команде. В зависимости от функции возвращается либо пустой пакет, либо данные

Ошибка:

Адрес 0x15	Код команды 1 байт 0x80-0x8F с битом ошибки	Номер Команды 1 байт	Код ошибки 1 байт	CRC 2 Байта
---------------	---	-------------------------	----------------------	----------------

В расчет CRC входят все поля

Таймауты:

- Ответ от прибора поступает не ранее чем через 8 байтный интервал времени после приема последнего байта команды.
- Ответ от АМ-01 на команды ТМК запросов ограничен ответом от тепловычислителей. АМ во время ТМК запроса способен получать системные команды. При любой другой команде во время запроса АМ-01 вернет ошибку о занятости устройства.
- В случае ошибки ТМК запроса, ошибка будет выдана не позднее чем через 8 сек (если прибор ответил пакетом но не сошлась сумма и делались повторные запросы), в большинстве случаев прибор не отвечает на команду и АМ-01 возвращает ошибку через 3 сек. При запросе версии прибора время ответа от 3 (в большинстве случаев) до 9 (при ошибке) секунд.

Алгоритм обмена:

1. Запрашиваем типа устройства (**MAIN_PARAM**) по коду устройства и версии прошивки определяем длину ответа 11 или 4 байта. Определяем правильность времени, при необходимости корректируем.
2. Запрашиваем версию устройства **TMK_VER** указывая адрес и скорость, либо только адрес, либо ничего. АМ посылает пакеты для пробуждения прибора и при ответе запрашивает версию и время. АМ автоматически определяет тип подключенного прибора.
3. Запрашиваем текущие параметры **TMK_CURR_PARAM**
4. Запрашиваем текущий суточный архив **TMK_DAY_CURR** затем последующий **TMK_DAY_NEXT**. При ошибочном ответе можно еще раз запросить текущий архив (кроме **TMK_H1**). Запрос можно начинать с последующего архива **TMK_DAY_NEXT** при первом запросе будет возвращен текущий архив.
5. Для выхода из локального режима посылаем **TMK_END** Эта команда обнуляет флаг четности и флаг следующей страницы после этого, команды запросы архива **TMK_DAY_NEXT** и **TMK_DAY_CURR** выдадут последнюю запись.
6. Послав в глобальном режиме **TMK_END** **TMK** переключается в **SLEEP** без ответа
7. Флаг четности и флаг следующей страницы обнуляются после команды **TMK_VER** а также при ошибке во время запроса.
8. Запрашиваем аналогично часовые архивы.
9. Для работы с другим прибором переходим к пункту 1
10. Отменить любой запрос можно командой **RESET_COMMAND_STATUS**
11. Завершить обмен по модему можно командой **RESET_MODEM** либо **RESET_DEVICE**

Команда **0x00 SYS_COMMAND** (Системные команды)

Системные команды. Выполняются при любой ситуации, служат для сброса и переключения режимов.

Регистр	Название	Параметры	Ответ	Описание
RESET_DEVICE	Перезагрузить устройство	-	Нет ответа	Программный Reset
RESET_MODEM	Сбросить модем	-	Нет ответа	Сбрасывает модем на 2 секунды. Переходит в режим работы с модемом, заново инициализирует модем и переходит в режим ожидания звонка
RESET_COMMAND_STATUS	Сбросить командный статус	-	-	Завершает все операции запроса. Переходит в режим ожидания команд

Команда **0x03 READ_REGISTER** (Чтение регистра)

Регистр	Название	Параметры	Ответ	Описание
MAIN_PARAM	Общие параметры	-	11 байт АМ-01 4 байта АЛ-01	Код устройства 2 байта Версия прошивки 2 байта Дата и время 7 байт Формат BCD Секунды 0-59 Минуты 0-59 Часы 0-23 Дата 1-31 Месяц 1-12 День недели 1-7 Год 0 – 99
TERMINAL_PARAM	Тип и скорость подключенного устройства	-	1 байт	Выдает версию оконечного устройства и скорость работы с ним. 0-2 бит – тип устройства ТМК-Н -0x00, МК-Н – 0x01, ТМК-Н2 – 0x02, ТМН-Н3 – 0x03 3 бит – скорость 0 – 4800, 1 – 9600
TMK_VER	Активировать, запросить версию и время оконечного устройства	1 байт – адрес устройства (default 0x00) 2 байт – скорость (default 0x00) 0-4800/1-9600	11 байт	Вывод из состояния SLEEP оконечного устройства с заданным адресом и скоростью. Время ожидания ответа до 9 сек. При неуспешном результате выдает ошибку Запрашивает версию и время и посылает 0x0D (окончание обмена). При успешном результате автоматически обновляет параметры подключенного устройства.
TMK_END	Запрос на окончание обмена	-	Пустой пакет	Посылает байт 0x0D для выхода из локальных режимов запроса. Сбрасывает внутренние флаги локальных запросов. Локальные запросы начинаются с первой страницы
TMK_DIRECT_REQUEST	Прямой запрос	Команды запроса	Данные запроса	АМ-01 выполняет действия согласно запросу
TMK_CURR_PARAM	Запрос текущих параметров	-	Текущие параметры. Длина зависит от версии	Считывает текущие параметры и посылает 0x0D окончание обмена

TMK_HOUR_CURR	Запросить текущую запись часового архива	-	Текущая запись. Длина зависит от версии	Считывает текущую запись часового архива. Повторный вызов будет неверным для версии ТМК-Н При ошибке чтение начинается с первой страницы
TMK_DAY_CURR	Запросить следующую запись часового архива	-	Следующая запись. Длина зависит от версии	Считывает следующую запись часового архива. Если до этого не была запрошена текущая запись. То выводится первая страница затем последующие. При ошибке чтение начинается с первой страницы
TMK_HOUR_NEXT	Запросить текущую запись суточного архива	-	Текущая запись. Длина зависит от версии	Считывает текущую запись суточного архива. Повторный вызов будет неверным для версии ТМК-Н. При ошибке чтение начинается с первой страницы
TMK_DAY_NEXT	Запросить следующую запись суточного архива	-	Следующая запись. Длина зависит от версии	Считывает следующую запись суточного архива. Если до этого не была запрошена текущая запись. То выводится первая страница затем последующие. При ошибке чтение начинается с первой страницы
ANSWER_TIME * только для(AM-01)	Время захвата линии и освобождения линии	-	4 байта	Два двухбайтовых числа минут начиная с 0:00 Максимальное значение 1339 0x053B
DEVICE_ARRAY	Список подключенных приборов	-	10 байт	Массив из двухбайтовых величин 0 байт – 7 бит – скорость 0 – 4800, 1 – 9600 0 – 6 – адрес устройства 1 байт – тип прибора ТМК-Н -0x00, МК-Н – 0x01, ТМК-Н2 – 0x02, ТМН-Н3 – 0x03 Ячейка пустая если оба байта равны 0
INIT_STRING * только для(AM-01)	Строка инициализации	-	Строка без 0 В конце	Возвращает строку инициализации модема
OK_STRING * только для(AM-01)	Строка успешного ответа	-	Строка без 0 В конце	Возвращает строку успешного ответа модема
ANSWER_STRING * только для(AM-01)	Строка ответа на входящий звонок	-	Строка без 0 В конце	Возвращает строку ответа на входящий звонок модема

RING_STRING * только для(AM-01)	Строка входящего звонка	-	Строка без 0 В конце	Возвращает строку входящего звонка модема
RTC_CORRECT_VALUE * только для(AM-01)	Величина корректировки RTC	-	1 байт	Возвращает величину корректировки RTC в секундах 0 соответствует -30 сек/сутки 60 соответствует +30 сек/сутки

Команда **0x10 WRITE_REGISTER** (Запись регистра)

Регистр	Название	Параметры	Ответ	Описание
CURRENT_TIME * только для(AM-01)	Текущее время	7 байт	-	Запись внутреннего времени. Формат BCD Секунды 0-59 Минуты 0-59 Часы 0-23 Дата 1-31 Месяц 1-12 День недели 1-7 Год 0 – 99
ANSWER_TIME * только для(AM-01)	Время захвата линии и освобождения линии	4 байта	-	Два двухбайтовых числа минут начиная с 0:00 Максимальное значение 1339 0x053B
DEVICE_ARRAY	Список подключенных приборов	10 байт	-	Массив из двухбайтовых величин 0 байт – 7 бит – скорость 0 – 4800, 1 – 9600 0 – 6 – адрес устройства 1 байт – тип прибора ТМК-Н -0x00, МК-Н – 0x01, ТМК-Н2 – 0x02, ТМН-Н3 – 0x03 Ячейка пустая если оба байта равны 0
INIT_STRING * только для(AM-01)	Строка инициализации	Строка без 0 В конце	-	Задаёт строку инициализации модема. В конце строки должен быть символ возврата каретки 0x0D
OK_STRING * только для(AM-01)	Строка успешного ответа	Строка без 0 В конце	-	Задаёт строку успешного ответа модема
ANSWER_STRING * только для(AM-01)	Строка ответа на входящий звонок	Строка без 0 В конце	-	Задаёт строку ответа на входящий звонок модема. В конце строки должен быть символ возврата каретки 0x0D

RING_STRING * только для(АМ-01)	Строка входящего звонка	Строка без 0 В конце	-	Задаёт строку входящего звонка модема
TMK_SET_TEMPR	Задать температуру холодной воды	1 байт	-	Задаёт (Тхв * 10) для ТМК2 и ТМК3
RTC_CORRECT_VALUE * только для(АМ-01)	Величина корректировки RTC	1 байт	-	Задаёт величину корректировки RTC в секундах 0 соответствует -30 сек/сутки 60 соответствует +30 сек/сутки
NET_ADDRESS * только для(АЛ-01)	Сетевой адрес	1 байт	-	Задать сетевой адрес для модификации с 485 интерфейсом

Команды прямого запроса

```

#define PUT_HEADER 0x01 // Очистить буфер. Добавить заголовок в буфер отправки указав длину пакета
#define SET_ATTEMPTS 0x02 // Установить количество попыток
#define SEND_COMMAND 0x03 // Послать команду
#define PREPARE_WAKE_UP_TMK 0x04 // Подготовка к пробуждению
#define WAKE_UP_TMK 0x05 // Вывод из состояния sleep
#define RECV_TMK_PACKET 0x06 // Принять N байт пакета ТМК (записывать в буфер проверять CRC)
#define DEFINE_VERSION 0x07 // Определить версию прибора
#define TRANSFER_PACKET 0x08 // Отослать сформированный пакет
#define SEND_HOLD 0x09 // Отослать байт для удержания линии
#define REPEAT_COMMAND 0x0A // Повторить команду через одну назад N раз не считая уже запрошенной
#define TRANSFER_COMPLETE_PACKET 0x0B // Отослать сформированный пакет обновив его длину
#define TRANSFER_PACKET_REQ 0x0C // Отослать пакет с затребованной ранее длиной
#define TRANSFER_BACK_BUFFER 0x0D // Отослать предыдущую копию пакета
#define SEND_TMK_PACKET 0x0E // Отослать пакет

```

При работе командами запроса необходимо подставлять размеры пакетов данных для каждого типа прибора.

При работе командами прямого запроса можно считывать несколько страниц архивных данных

Для работы со всеми приборами достаточно несколько наборов запросов. Запрос формируется последовательными командами с параметром.

Далее приводятся массивы данных для нескольких типов прямого запроса.

Разбудить ТМК, МК и считать время и версию. Аналогия команды **ТМК_VER**

BYTE tmk_80[] =

```
{
    PUT_HEADER,11,                // Подготовить заголовок
    PREPARE_WAKE_UP_TMK,0,
    WAKE_UP_TMK,0x80,            // Разбудить ТМК по 0 адресу. Если адрес не 0 то подставляем (0x80 | адрес)
    SET_ATTEMPTS,3,             // Количество попыток - 3
    SEND_COMMAND,0x06,          // Команда 0x06
    RECV_TMK_PACKET,11,         // Принять данные и проверить CRC. При ошибке делается повторы заданными попытками
    SEND_COMMAND,0x0D,          // Команда 0x0D - закончить обмен
    TRANSFER_PACKET, 0x00        // Отправить запрос
};
```

По 11 байту определяем тип прибора и версию протокола обмена

Версия протокола	Тип прибора
0x00...0x07	ТМК-Н1
0x08	ТМК-Н2
0x09, 0x0A	ТМК-Н3, ТМК-Н5
0xF0	МК-Н
0x0C	ТМК-Н12
0x0B	ТМК-Н13

Размеры считываемых данных в байтах:

Тип прибора	Текущие данные	Суточные данные	Часовые данные
ТМК-Н1	81	23 + 23 + 23	18 + 18
ТМК-Н2	57	26	20
ТМК-Н3	99	46	36
ТМК-Н5	103	48	37
МК-Н	29	21	10
ТМК-Н12	67	27	21
ТМК-Н13	112	50	39

Считать текущие параметры. Аналогия команды **TMK_CURR_PARAM**. Пример для ТМК-Н12

```
BYTE tmk_08[] =
{
    PUT_HEADER, 67,                // Подготовить заголовок 67 байт
    SET_ATTEMPTS, 3,               // Количество попыток - 3
    SEND_COMMAND, 0x08,           // Команда 0x08
    RECV_TMK_PACKET, 67,          // Принять данные и проверить CRC. При ошибке делается повторы заданными попытками
    SEND_COMMAND, 0x0D,           // Команда 0x0D - закончить обмен
    TRANSFER_PACKET, 0x00         // Отправить запрос
};
```

Считать первую страницу суточного архива. Пример для ТМК-Н12

```
BYTE tmk_day[] =
{
    PUT_HEADER, 27,                // Подготовить заголовок 27 байт
    SET_ATTEMPTS, 3,               // Количество попыток - 3
    SEND_COMMAND, 0x0C,           // Команда 0x0C
    RECV_TMK_PACKET, 27,          // Принять данные и проверить CRC. При ошибке делается повторы заданными попытками
    TRANSFER_PACKET, 0x00         // Отправить запрос
};
```

Последующие страницы считываем, заменяя команду 0x0C на 0x0B, 0x0A, 0x0B, 0x0A, и т.д.

Считать несколько страниц суточного архива с первой. Пример для ТМК-Н12

```
BYTE tmk_day_packet[] =
{
    PUT_HEADER, 27 * n,            // Подготовить заголовок 27 байт * n страниц
    SET_ATTEMPTS, 3,               // Количество попыток - 3
    SEND_COMMAND, 0x0C,           // Команда 0x0C
    RECV_TMK_PACKET, 27,          // Принять данные и проверить CRC. При ошибке делается повторы заданными попытками
    REPEAT_COMMAND, n-1,          // Повторить 2 предыдущие команды n-1 раз
    TRANSFER_PACKET, 0x00         // Отправить запрос
};
```

Слеует отметить что при каждом повторе команда 0x0C автоматически заменяется на 0x0B, 0x0A, 0x0B, 0x0A, и т.д. Для запроса следующих страниц необходимо начинать с команды 0x0A или 0x0B в зависимости от того на какой команде закончился запрос. Количество запрашиваемых страниц ограничено размером буфера АМ-01 равным 256 байт на данные ответа. Аналогично производится считывание часового архива.

!!! При считывании архива, перед началом считывания другого типа архива, считывания с архива с начала либо запроса другой команды необходимо выйти из режима считывания архива командой **TMK_END**

```

//-----Функции-----//
#define SYS_COMMAND      0x00          // Системные команды
#define READ_REGISTER    0x03          // Чтение регистра
#define WRITE_REGISTER   0x10          // Запись регистра
//-----Регистры-----//
#define MAIN_PARAM       0x00          // Общие параметры (тип устройства, версия, время)
#define TMK_SET_TEMPR    0x01          // Задать температуру холодной воды
#define TERMINAL_PARAM   0x02          // Тип и скорость подключенного устройства
#define RTC_CORRECT_VALUE 0x03          // Величина корректировки RTC
#define NET_ADDRESS      0x04          // Сетевой адрес
#define ANSWER_TIME      0x05          // Время захвата и освобождения линии
#define CURRENT_TIME     0x06          // Текущее время
#define DEVICE_ARRAY     0x07          // Список подключенных приборов
#define INIT_STRING      0x08          // Строка инициализации
#define OK_STRING        0x09          // Строка успешного ответа
#define ANSWER_STRING    0x0A          // Строка ответа
#define RING_STRING      0x0B          // Строка входящего звонка
#define TMK_CURR_PARAM   0x10          // Запросить текущие параметры
#define TMK_HOUR_CURR    0x12          // Запросить текущую запись часового архива
#define TMK_DAY_CURR     0x13          // Запросить текущую запись суточный архива
#define TMK_HOUR_NEXT    0x22          // Запросить следующую запись часового архива
#define TMK_DAY_NEXT     0x23          // Запросить следующую запись суточного архива
#define TMK_VER          0xF0          // Запросить версию и время
#define TMK_END          0xF1          // Запрос на окончание обмена
#define TMK_DIRECT_REQUEST 0xF2          // Прямой запрос
//-----//
#define RESET_DEVICE     0x00          // Сброс устройства
#define RESET_MODEM      0x01          // Сброс модема
#define RESET_COMMAND_STATUS 0x02          // Завершить все операции перейти в режим ожидания команды
//-----Коды ошибок-----//
#define UNKNOWN_ERROR    0x00          // Общая ошибка(без конкретизации причины)
#define ILLEGAL_FUNCTION 0x01          // Недопустимый(неподдерживаемый) код команды
#define ILLEGAL_DATA_ADDRESS 0x02          // Недопустимый(неверный) номер регистра
#define ILLEGAL_DATA_VALUE 0x03          // Недопустимое значение в поле данных
#define SLAVE_DEVICE_FAILURE 0x04          // Внутренняя ошибка прибора
#define SLAVE_DEVICE_BUSY 0x06          // Устройство занято
#define GATEWAY_TARGET_FAILED 0x0B          // Оконечное устройство не ответило

```

Алгоритм расчета CRC

```
unsigned short CalculateCrc(unsigned char *buf, unsigned short size)
{
    WORD retval=0xFFFF;
    BYTE j,carry;
    for( ; size != 0 ; buf++,size-- ) {
        retval ^= *buf;
        for (j = 0; j < 8; j++){
            carry = retval & 1;
            retval >>= 1;
            if (carry)
                retval ^= 0xA001;
        }
    }
    return retval;
}
```