**Описание протокола**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Start 1** | **Start 2** | **Address D** | **Address S** | **Command** | **Parameter/Error** | **Data Size** | **Data** | **CRC16** | **Stop1** | **Stop2** |
| **Значение** | 0x10 | 0x02 |  |  |  |  |  |  |  | 0x10 | 0x03 |
| **Количество байт** | 1 | 1 | 4 | 4 | 2 | 2 | 4 | 0..Data Size | 2 | 1 | 1 |

**Из этого вычисляется CRC16**

**Все это подлежит байтстаффингу**

**Start1, Start2:** Начало пакета (0х10 0х02).

**Stop1, Stop2:** Конец пакета (0х10 0x03).

**Address D (Address Destination):**  IP-адрес получателя

**Address S (Address Source):** IP-адрес отправителя

**Command:** Код команды (см. Таблицу 2). Старший бит всегда отвечает за направление передачи. Если 0 то запрос, если 1 то ответ.

**Parameter/Error:** Код ошибки в ответном пакете. При запросе всегда равен 0. При ответе если больше 0, то ошибка (см. Таблицу 2) иначе поле Data содержит ответные данные.

**Data Size:** Размер поля данных (Data) в байтах.

**Data:** Поле для байтов данных (может быть от 0 до Data Size байт).

**CRC16:** Контрольная сумма, вычисляется как полином = 0х1021, стартовое значение = 0x0000.

**Внимание!**

**Пакет *перед посылкой* подлежит обработке механизмом *байтстаффинга*.**

**Пакет *при приеме* подлежит обработке механизмом *обратного байтстаффинга*.**

Байтстаффингу подвергаются все байты пакета за исключением стартовых и стоповых.

Стартовая последовательность - 0х10 0х02, стоповая - 0х10 0x03 все, что между ними, подлежит байтстаффингу.

При байтстаффинге анализируется каждый байт, и к каждому байту 0x10 добавляется байт 0x10

При обратном байтстаффинге вместо пары последовательных байт (0x10 0x10) остаётся только один (0x10).

*Все значения более 1-го байта передаются как:* первый байт всегда младший и идет первым

**Описание команд**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование**  **Команды** | **Код, hex** | **Описание команды** | **Запрос: передаваемые данные (Поле Data)** | **Ответ: передаваемые данные (Поле Data)** |
| Gate Request | 0x102 | Команда организации канала связи с указанным шлюзом через М2М | 1. Номер шлюза – 4 байта (Номер шлюза указывается в прямом шестнадцатеричном формате. То есть если номер шлюза 10007, то первые четыре байта должны иметь вид: 0х 07 00 01 00); 2. Номер канала обмена – 1 байт: 0 по умолчанию; 3. Описание подключения – 30 байт ASCII | Один байт:  0x00 – канал открыт;  0x01 – шлюза нет на связи;  0xF1 – м2м не имеет лицензии |
| Find Gate | 0x104 | Команда запроса статуса указанного шлюза | 1. Номер шлюза – 4 байта (Номер шлюза указывается в прямом шестнадцатеричном формате. То есть если номер шлюза 10007, то первые четыре байта должны иметь вид: 0х 07 00 01 00) | 1. Тип структуры данных – 2 байта 2. Статус шлюза – 2 байта (0 – не зарегистрирован, 1 – на связи, 2- не на связи, 3 – канал занят клиентом) 3. Регистрационный ID – 2 байта 4. Структура данных о шлюзе |
| End Of Exchange | 0xFFFE | Команда окончания обмена данными | 1. Тип программного обеспечения - 1 байт 2. Код компьютера - *SHORT\_STRING* 3. Параметр команды от 0 до 65535 – 2 байта (UINT16)   0 – окончание обмена | Ответ не предусмотрен |

STRING - массив байт (первый байт длина в символах и далее тело строки; на каждый символ тела строки выделяется два байта в кодировке UNICODE)