

# Введение в микроконтроллеры

Домашнее задание №5

Евгений Зеленин

27 января 2025 г.

## 1. Постановка задачи

**Условия задачи.** Реализуйте передачу запрошенной области адресного пространства по USB в формате Intel HEX. Например, на запрос `get_hex 8000000 2` ответ должен быть в виде

0200000040800F2

0200000000000FE

00000001FF

## 2. Описание работы

В принципе, в этом задании нет каких-то особых подключений, поэтому можно оставить лишь небольшую описательную часть и алгоритм. Подключение к плате `mcuinside` осуществляется через MicroUSB разъем. Для работы выбран режим CDC и подключен соответствующий программный стек в разделе Middleware. Устройство принимает команды и передает ответ через виртуальный COM-port. В случае введения некорректных команд, в консоль выводится справка с указанием списка команд и формата принимаемых данных.

```
Usage: get_hex [address] [bytes]          get specified count of [bytes] from memory at [address]
Example: get_hex 8000000 64
```

Рис. 1: Вывод справки

В случае корректно введенной команды, функция `print_to_ihex()` выводит в виртуальный COM порт указанное количество байт начиная с заданного адреса.

```
:0200000040800F2
:10000000000000120390B00089D090008A309000821
:10001000A9090008AF090008B50900080000000A0
:10002000000000000000000000000000BB09000804
:10003000C9090008000000000D7090008E509000808
:10004000890B0008890B0008890B0008890B000840
:10005000890B0008890B0008890B0008890B000830
:10006000890B0008890B0008890B0008890B000820
:10007000890B0008890B0008890B0008890B000810
:10008000890B0008890B0008890B000800000009C
:10009000000000000000000000000000890B0008C4
:1000A000890B0008890B0008890B0008890B0008E0
:1000B000890B0008890B0008890B0008890B0008D0
:1000C000890B0008890B0008890B0008890B0008C0
:1000D000890B0008890B0008890B000800000004C
:00000001FF
```

Рис. 2: Пример ответа

**Принцип построения ответа следующий:**

Первой формируется строка вида :02 0000 09 0800 F2, где 02 говорит о том, что в строке содержится 2 байта данных, 0000 - младшие два байта адреса 02 - функция (передача старших двух байт адреса), 0800 старшие два байта введенного адреса F2 - контрольная сумма.

Далее, формируются строки с байтами данных начиная с указанного адреса, в заданном количестве байт :10 0010 00 A90900...000 A0- младшие два байта адреса (старшие указаны в первой строке), 00 - код функции (передача данных из памяти), затем следует строка с данными, далее - контрольная сумма (A0).

В конце ответа всегда пишется стандартная строка :00 0000 01 FF (01) - конец файла.

### 3. Исходный код проекта

Ниже можно ознакомиться с сходным кодом функции *void print\_to\_ihex(void\*addr, uint32\_t len)* и переопределенную *int write(int file, uint8\_t \*data, uint16\_t len)* (используется для вывода в printf через CDC устройство).

```
void print_to_ihex(void *addr, uint32_t len)

{
    //printing Extended Segment Address Record to ASCII codes
    uint16_t tmp = (uint16_t) (((uint32_t) addr) >> 16);
    uint8_t checksum = 0;

    char my_buf[45] = "\n:02000004";
    u16_to_hex(&my_buf[10], tmp);
    checksum = 0x1u + ~(6u + (tmp >> 8 & 0xff) + (tmp & 0xff));
    u8_to_hex(&my_buf[14], checksum);
    my_buf[16] = '\n';

    CDC_Transmit_FS((uint8_t*) my_buf, 17);

    //printing Data record to ASCII
    int i = 0;
    do {
        checksum = 0;
        my_buf[0] = ':';
        //printing a segment address in ASCII at positions my_buf[3..8]
        //my_buf[1] and my_buf[2] stores byte count in ASCII
        tmp = (uint16_t)((uint32_t)(addr + i)) ;
        checksum += tmp;
        u16_to_hex(&my_buf[3], tmp);

        //TYPE 00
        tmp = 0x0u;
        u8_to_hex(&my_buf[7], tmp);

        //printing HEX bytes in ASCII (positions my_buf[9...])
        do {
            tmp = *(((uint8_t*) addr) + i);
            checksum += tmp;
            u8_to_hex(&my_buf[9 + 2 * (i % 16)], tmp);
            //      1 + 2 + 4 + 2
            // 9 = ":" + "LL" + "AAAA" + "TT"
        } while (++i % 16 && i < len);
        //i % 16 - because we use maximum 16 data bytes count for the one string

        //if i == 16, i % 16 = 0, but tmp should be 16.
        tmp = i % 16 ? : 16; //tmp here store a byte count.
```

```

    u8_to_hex(&my_buf[1], tmp);

    checksum += tmp;
    checksum = 0x1u + ~checksum;
    //checksum calculation: summ = 1 + NOT(Val_1 + Val_2 + ... + Val_N)
    u8_to_hex(&my_buf[9 + 2 * tmp], checksum);
    my_buf[11 + 2 * tmp] = '\n';
    CDC_Transmit_FS((uint8_t*) my_buf, tmp * 2 + 12); //transmit buffer via CDC Device

} while (i < len);
CDC_Transmit_FS((uint8_t*) ":00000001FF\n", 12);
}

void u8_to_hex(char *buff, uint8_t byte) {
    const char hex[] = "0123456789ABCDEF";
    buff[0] = hex[(byte >> 4) & 0xF];
    buff[1] = hex[byte & 0xF];
}

void u16_to_hex(char *buff, uint16_t word) {
    u8_to_hex(buff, (word >> 8) & 0xFF);
    u8_to_hex(buff + 2, word & 0xFF);
}

int __write(int file, uint8_t *data, uint16_t len)

{
    CDC_Transmit_FS(data, len);
    return len;
}

```

Исходные коды проекта приложены к дополнительным материалам отчета.

#### 4. Заключение

В ходе выполнения практического задания получилось осуществить коммуникацию платы mcuinside с ПК через USB-интерфейс в режиме CDC. Что интересно, в случае переопределения printf через \_\_write или \_\_iowritechar иногда возникали потери символов. При отправке через CDC\_Transmit\_FS все работало корректно.

#### 5. Дополнительные материалы

Материалы к отчету расположены в папке на google диск по следующей ссылке: Материалы к ДЗ №05