

My Controller

Изучаем микроконтроллеры STM32

- [Home](#)
- [О сайте](#)
- [СОФТ](#)

« [STM32 Внешние устройства. Введение](#)
[STM32 SPI Введение](#) »

STM32 SD Card. Инициализация и использование карты памяти

Общение с SD-картой производится путем отправки специальных команд, которые включают в себя:

- код команды;
- 32-х разрядный аргумент;
- код CRC, подтверждающий правильность данных (в режиме SPI этот код нужен только для двух команд)

Команд много, но для решения поставленной задачи можно ограничиться перечисленными ниже:

```
1 #define GO_IDLE_STATE          0           //Программная перезагрузка
2 #define SEND_IF_COND          8           //Для SDC V2 - проверка диапазона напряжений
3 #define READ_SINGLE_BLOCK     17          //Чтение указанного блока данных
4 #define WRITE_SINGLE_BLOCK    24          //Запись указанного блока данных
5 #define SD_SEND_OP_COND       41          //Начало процесса инициализации
6 #define APP_CMD                55         //Главная команда из ACMD <n> команд
7 #define READ_OCR               58         //Чтение регистра OCR
```

Командам даны осмысленные имена, которые будем использовать при обращении к функции, посылающей их карте памяти. Текст этой функции приведен ниже:

[Показать код](#)

Дальше следуют функции, посредством которых основная программа будет общаться с картой памяти

Функция инициализации **uint8_t SD_init(void)**

[Показать код](#)

При вызове этой функции выполняется инициализация карты памяти, определяется ее тип (SD или SDHC).

В случае успешной инициализации, возвращается ноль, иначе — единица.

Если карта успешно инициализирована, можно выполнять чтение/запись секторов карты. Для этого предназначены две следующие функции.

Функция чтения выбранного сектора **uint8_t SD_ReadSector(uint32_t BlockNumb, uint8_t *buff)**

[Показать код](#)

Функция записи выбранного сектора **uint8_t SD_WriteSector(uint32_t BlockNumb, uint8_t *buff)**

[Показать код](#)

Приведенный ниже пример иллюстрирует порядок использования данных функций.

В нем выполняется копирование данных из сектора №2 в сектор №8:

```
1 uint8_t Buff[512];           //буфер для чтения/записи сектора карты           //
2
3 SD_init();                   //выполнить инициализацию
4 SD_ReadSector(2, &Buff);     //копировать сектор №2 в буфер
5 SD_WriteSector(8, &Buff);    //записать данные из буфера в сектор №8
```

Как видите, если использовать приведенные выше функции, то работать с картой памяти очень просто. Единственное, что отсутствует в этом примере — это анализ результата, возвращаемого функциями. Если функция вернет единицу, значит нет связи с картой. Это нужно учитывать.

Если требуется увеличить скорость обмена, необходимо при инициализации модуля SPI задать другое значение скорости.

Небольшое примечание. Я работал с картами памяти «Transcend» с объемом 2 гигабайта.

Другие тоже должны работать, но я не проверял.

Рубрика: [Карта памяти SD](#)

Комментарии (53) на “STM32 SD Card. Инициализация и использование карты памяти”

- Sky:
[17.02.2012 в 13:12](#)

А для чего задана переменная wait в функции чтения сектора, ошибочно?

[Ответить](#)

- kontroller:
[17.02.2012 в 13:48](#)

Осталась от последней модернизации функции. Сейчас уберу. Спасибо.

Получилось с SPI2 ?

[Ответить](#)

- Sky:
[17.02.2012 в 13:53](#)

К сожалению не получается, щупаю ножку PB14(MISO) осциллографом — она притянута вверх но никаких сигналов не пробегает.. по остальным что то бежит :(

[Ответить](#)

- kontroller:
[17.02.2012 в 14:00](#)

А если проверить работу модуля как я раньше описал?

Также некоторую проверку модуля можно выполнить, если соединив между собой MISO и MOSI послать байт данных. Если приняли посланный байт, значит порт настроен правильно, модуль работает. Под вопросом останется правильность настройки фазы и полярности тактового сигнала и порядок выдачи данных.

[Ответить](#)

- Sky:
[17.02.2012 в 14:09](#)

Если я замыкаю эти выводы — процессор вообще не запускается

[Ответить](#)

- Sky:
[17.02.2012 в 14:13](#)

А нет, запустился. Тока вот как правильно послать байт? Я делаю так:

```
SD_WriteSector(8, «f»);  
SD_ReadSector(8, & *Buff);
```

[Ответить](#)

- kontroller:
[17.02.2012 в 14:18](#)

Примерно так:

```
uint8_tmp;  
tmp = spi_send (0xA5);  
Отправили 0xA5 и его же должны получить.
```

[Ответить](#)

- Sky:
[17.02.2012 в 14:31](#)