

Современные 32-разрядные ARM-микроконтроллеры серии STM32: использование карт памяти SD и MMC в режиме SPI

Олег Вальпа (г. Миасс, Челябинская обл.)

В статье приведён способ подключения карт памяти SD и MMC к микроконтроллерам серии STM32 от компании STMicroelectronics по интерфейсу SPI и примеры программ обращения к карте памяти в режиме SPI.

ВВЕДЕНИЕ

Несмотря на то, что микроконтроллеры серии STM32 от компании STMicroelectronics [1] обладают солидными объёмами памяти в сотни мегабайт, для некоторых проектов этого оказывается недостаточно. Примерами таких проектов являются устройства для записи и воспроизведения звука и изображений. Использование микроконтроллеров в устройствах с графическими дисплеями также требует объёмных хранилищ данных. Для упомянутых устройств требуется память объёмом в десятки гигабайт.

К счастью, в настоящее время в мире уже производятся элементы памяти большого объёма, который измеряется в гигабайтах. При этом они доступны по цене даже для массового потребителя. Наиболее подходящими элементами для проектов с микроконтроллерами являются карты памяти SD (Secure Digital).

ОБЗОР КАРТ ПАМЯТИ

Карты памяти SD представляют собой миниатюрные прямоугольные плоские

устройства с контактными площадками для подключения к интерфейсу аппаратуры. Они предназначены для многократной записи и хранения информации. Эти карты памяти были разработаны для использования в портативных устройствах. В настоящее время они активно используются в цифровых фотоаппаратах, видеокамерах, мобильных телефонах, смартфонах, планшетах, электронных книгах, GPS-навигаторах, в некоторых игровых приставках и т.п.

В пластиковом корпусе этих карт памяти размещается флэш-память и специализированный контроллер (см. рис. 1).

Первые карты памяти SD появились в 1999 г. В настоящее время существуют три разновидности карт памяти SD, отличающиеся по своим габаритным размерам и назначению контактов. Это карты SD, mini SD и micro SD. На рисунке 2 представлены все три типа данных карт памяти, с указанием их размеров.

Прототипом карт памяти SD послужили карты памяти MMC (Multi Media Card), появившиеся на свет в 1996 г. Раз-

меры такой карты составляют $24 \times 32 \times 1,5$ мм, напряжение питания – 3,3 В, а ёмкость памяти – от 64 КБ до 4 ГБ. Карта памяти MMC совместима с разработанной позднее SD-картой и может использоваться вместо неё. Обратная замена не всегда возможна, так как SD-карты толще карт MMC и механически не входят в слот для MMC-карт.

Существуют четыре поколения карт памяти SD-формата, различающиеся объёмом данных:

- SD 1.0 – от 8 МБ до 2 ГБ;
- SD 1.1 – до 4 ГБ;
- SDHC – до 32 ГБ;
- SDXC – до 2 ТБ.

Карты памяти SD могут поддерживать несколько режимов обмена данными. Одним из таких режимов является относительно простой и открытый интерфейс SPI (Serial Peripheral Interface). Данный режим обмена данными поддерживают карты памяти SD и MMC. Назначение контактов для всех карт памяти приведено на рисунке 3.

Интерфейс карт памяти включает в себя от 7 до 11 контактов. Некоторые из этих контактов не используются и зарезервированы. Назначение контактов карт памяти приведено в таблице 1.

Питание карт памяти осуществляется от источника постоянного тока с выходным напряжением 3,3 В. В состоянии покоя карта памяти потребляет

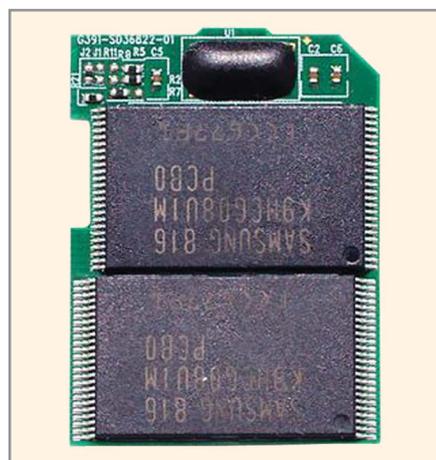


Рис. 1. Устройство карты памяти

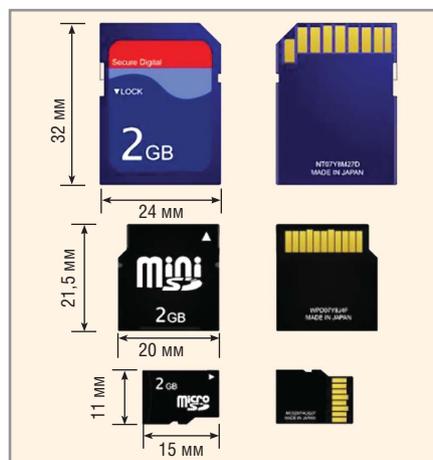


Рис. 2. Типы карт памяти

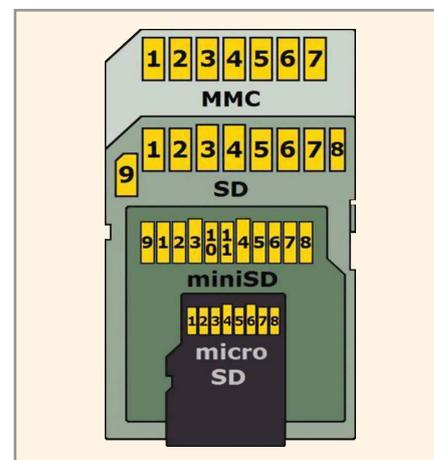


Рис. 3. Назначение контактов карт памяти

ток не более 1 мА, а при передаче данных – от 20 до 100 мА. Скорость передачи данных по интерфейсу SPI может достигать 12 Мбод.

На рисунке 4 приведена структурная схема карты памяти SD. На этой схеме показано устройство карты памяти и состав внутренних регистров, доступных для чтения по интерфейсу карты.

Подключение карты памяти к микроконтроллеру

Подключить SD-карту к микроконтроллеру можно несколькими способами, но наиболее предпочтительным является подключение карты к выводам SPI, поскольку данный способ позволяет задействовать аппаратные средства микроконтроллера и готовые библиотеки для работы с SPI-интерфейсом.

Режим SPI является альтернативным режимом работы с картами памяти MMC и SD и более простым, по сравнению с использованием протокола SD. Карту памяти можно подключить к большинству микроконтроллеров через встроенный в них универсальный порт SPI или порт GPIO (универсальный порт ввода/вывода). Поэтому режим SPI прекрасно подходит для недорогих встраиваемых приложений.

Схема подключения карты памяти SD к микроконтроллеру приведена на рисунке 5.

Протокол обмена данными

Рассмотрим протокол обмена данными с картой памяти SD в кратком варианте, включающем в себя основную и необходимую для понимания процессов обмена информацию. Подробное описание протокола изложено в [2].

В режиме SPI данные передаются последовательно по байтам. Наиболее подходящим режимом SPI для работы с картами памяти MMC и SD является режим SPI 0, формирующий положительный синхроимпульс, защёлкивание по переднему фронту и сдвиг по заднему фронту.

Кадр команды, передаваемый от микроконтроллера к карте, имеет фиксированную длину в 6 байт. Формат обмена данными между микроконтроллером и картой памяти по интерфейсу SPI приведён на рисунке 6.

В ответ на полученную команду карта памяти передаёт ответ с условным названием R1, R2 или R3. Поскольку процедура обмена данными тактируется синхроимпульсами, генерируемыми микроконтроллером, последний должен

формировать синхронизацию и продолжать чтение байтов, пока не получит действительный ответ. Время ответа на команду обозначено на диаграмме (см. рис. 6) символами N*8 и составляет от 0 до 8 байт для карт памяти SD, и от 1 до 8 байт для карт MMC. Сигнал CS должен удерживаться в состоянии низкого уровня в течение всей транзакции, включающей в свой состав команду, ответ и передачу данных, если таковые имеются. Код CRC в режиме SPI не используется, но его присутствие с произвольным значением в кадре команды необходимо для полного заполнения пакета посылки. При необходимости достоверный код CRC генерируется с помощью полинома $x^{16}+x^{12}+x^5+1$ по стандарту CCITT.

Команды режима SPI

В режиме SPI для всех команд отведено 6 бит, поэтому количество команд

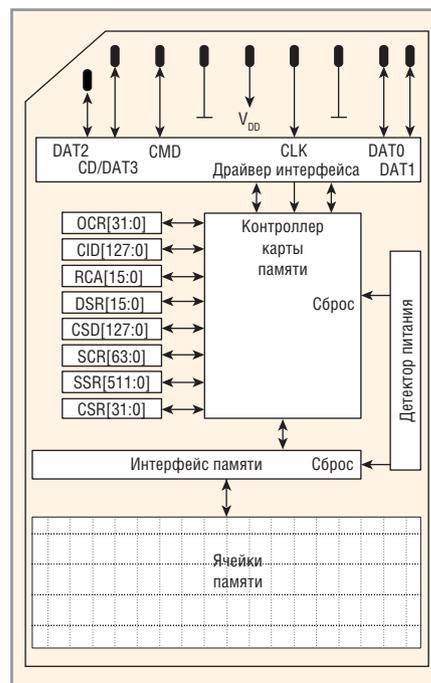


Рис. 4. Структурная схема карты памяти SD

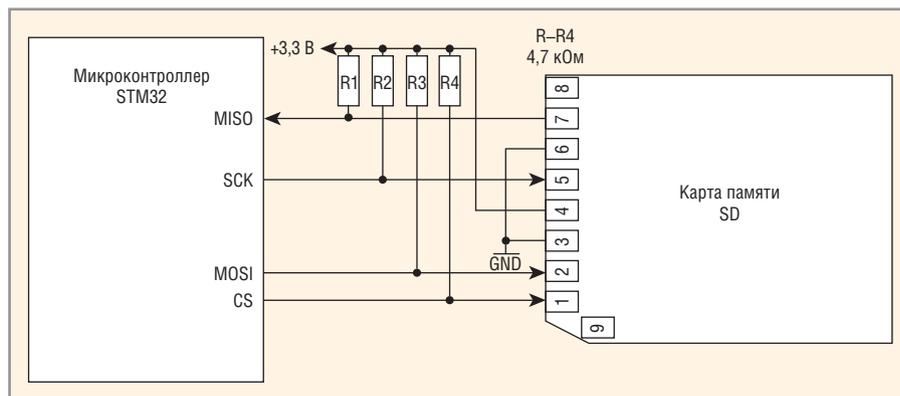


Рис. 5. Схема подключения карты памяти SD к микроконтроллеру

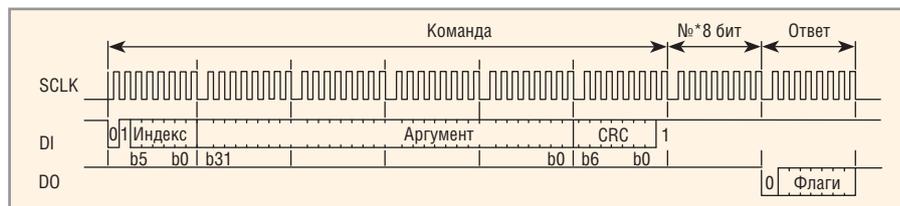


Рис. 6. Формат обмена данными по SPI

Таблица 1. Назначение контактов карт памяти

Вывод				Название вывода	Тип вывода	Назначение
MMC	SD	Mini SD	Micro SD			
1	1	1	2	CS	Вход	Выбор режима SPI (инверсный сигнал)
2	2	2	3	DI(MOSI)	Вход	Ввод данных SPI в последовательном режиме
3	3	3	-	GND	-	Общий
4	4	4	4	VDD	-	Питание от 2,7 до 3,6 В
5	5	5	5	SCK	Вход	Синхронизация SPI
6	6	6	6	GND	-	Общий
7	7	7	7	DO(MISO)	Выход	Вывод данных SPI в последовательном режиме
-	8	8	8	NC	-	Не используется
-	9	9	1	NC	-	Не используется
-	-	10	-	NC	-	Зарезервировано
-	-	11	-	NC	-	Зарезервировано

не может превышать 64. Все команды имеют условное обозначение CMD с номером от 0 до 63, который имеют индексом команды. Кроме того, каждой команде назначено индивидуальное имя в виде аббревиатуры, например GO_IDLE_STATE. Первый байт пакета команды состоит из комбинации двух бит 0 и 1 начала команды и шести бит индекса команды. Например, для команды CMD0 первый байт будет иметь значение 0x40, а для команды CMD59 – 0x7B. В таблице 2 приведён краткий перечень команд, используемых для чтения, записи и инициализации карты памяти. Более детальную информацию по всем командам можно найти в спецификации [2].

ОТВЕТ КАРТЫ ПАМЯТИ В РЕЖИМЕ SPI

Существует три типа ответа карты памяти на команду. Эти ответы имеют следующие условные обозначения: R1, R2 и R3. Тип ответа зависит от принятой команды. Для большинства команд возвращается байт ответа R1. Назначение бит в байте ответа R1 приведено в таблице 3.

Ответ 0x00 означает успешное выполнение команды. Если произойдёт какая-нибудь ошибка, будет установлен соответствующий бит ответа R1.

Ответ R2 возвращается только для команды запроса регистра статуса CMD13. Он состоит из двух байт. Его первый байт представляет собой ответ R1. Назначение всех бит второго байта ответа R2 приведено в таблице 4.

Ответ R3 возвращается только для команды CMD58. Его первый байт – это ответ R1, за которым следует содержимое регистра OCR карты памяти, состоящее из четырёх байт.

Некоторые команды требуют для ответа больше времени, чем N*8. Такие команды возвращают ответ R1b, который представляет собой ответ R1 и следующий за ним флаг занятости, т.е. на линии DO интерфейса удерживается низкий уровень до тех пор, пока выполняется внутренний процесс. Микроконтроллер должен ждать окончания процесса, пока не получит байт с содержимым, равным 0xFF.

АКТИВАЦИЯ РЕЖИМА SPI

При подаче питания на карту памяти происходит её аппаратный сброс и она переключается в режим обмена информацией SD. Чтобы переключить карту в режим SPI, требуется выполнить ряд процедур.

После подачи питания на карту памяти необходимо сделать паузу в несколько миллисекунд. Затем требуется установить на линиях DI и CS высокий уровень и подать не менее 74 импульсов на вход SCK. После этого карта памяти будет готова принять команду в режиме SPI.

ПРОГРАММНЫЙ СБРОС

Чтобы осуществить программный сброс карты памяти, нужно передать команду CMD0 при низком уровне на линии CS. После обнаружения команды CMD0 карта опрашивает сигнал CS. Если уровень сигнала CS низкий, т.е. активный, то карта входит в режим SPI. Команда CMD0 должна иметь корректный код CRC, значение которого должно быть равным 0x95 (является правильным только для команды CMD0). Т.е. следует отправить карте памяти по интерфейсу SPI следующий пакет данных: 0x40, 0x0, 0x0, 0x0, 0x0, 0x95. После того как карта памяти переключится в режим SPI, проверка CRC отключается и значение CRC не анализируется картой памяти в дальнейшем. Если команда CMD0 принята успешно, карта войдёт в состояние ожидания (Idle) и пошлёт ответ R1 с установленным в единицу битом In Idle State, т.е. R1 примет значение 0x01.

ИНИЦИАЛИЗАЦИЯ КАРТЫ ПАМЯТИ

В состоянии ожидания карта принимает только команды CMD0, CMD1

Таблица 2. Команды карты памяти SD в режиме SPI

Индекс команды	Аргумент	Ответ	Аббревиатура	Описание
CMD0	Нет	R1	GO_IDLE_STATE	Программный сброс карты памяти
CMD1	Нет	R1	SEND_OP_COND	Активировать процесс инициализации
ACMD41*	Нет	R1	APP_SEND_OP_COND	Начать процесс инициализации (только для карт SD)
CMD9	Нет	R1	SEND_CSD	Считать регистр CSD
CMD10	Нет	R1	SEND_CID	Считать регистр CID
CMD12	Нет	R1b	STOP_TRANSMISSION	Прекратить чтение данных
CMD13	Нет	R2	SEND_STATUS	Запрос регистра статуса
CMD16	Данные [31:0]	R1	SET_BLOCKLEN	Задать длину блока в байтах
CMD17	Адрес [31:0]	R1	READ_SINGLE_BLOCK	Считать блок данных
CMD18	Адрес [31:0]	R1	READ_MULTIPLE_BLOCK	Считать множество блоков
CMD23	Данные[15:0]	R1	SET_BLOCK_COUNT	Задать количество блоков для обмена (только для карт MMC)
ACMD23*	Данные [22:0]	R1	SET_WR_BLOCK_ERASE_COUNT	Задать количество блоков для предварительного стирания (только для карт SD)
CMD24	Адрес [31:0]	R1	WRITE_BLOCK	Записать блок
CMD25	Адрес [31:0]	R1	WRITE_MULTIPLE_BLOCK	Записать множество блоков
CMD55*	Нет	R1	APP_CMD	Команда расширения
CMD58	Нет	R3	READ_OCR	Считать регистр OCR
CMD59	Данные [0]	R1	CRC_ON_OFF	1 – включить, или 0 – выключить анализ CRC

* Команда ACMDn состоит из последовательности команд CMD55 и CMDn

Таблица 3. Назначение бит в байте ответа R1

Бит	Имя или значение	Описание
0	In Idle State	Карта находится в состоянии ожидания и запущен процесс инициализации
1	Erase Reset	Последовательность команд стирания была очищена
2	Illegal Command	Был обнаружен недопустимый код команды
3	Communication CRC Error	Проверка CRC последней команды ошибочна
4	Erase Sequence Error	Произошла ошибка в последовательности команд стирания
5	Address Error	Ошибка выравнивания адреса, который не совпадает с длиной блока, использованного в команде
6	Parameter Error	Аргумент команды (например, адрес, длина блока) находится за пределами допустимого диапазона для этой карты
7	0	Всегда нулевой бит

Таблица 4. Назначение бит во втором байте ответа R2

Бит	Имя или значение	Описание
0	Card is Locked	Карта памяти заблокирована
1	WP Erase Skip Lock/Unlock CMD Failed	Попытка стирания защищённого от записи блока или команда блокировки/разблокировки не удалась
2	Error	Во время выполнения команды произошла неизвестная ошибка
3	CC Error	Внутренняя ошибка контроллера карты памяти
4	Card ECC Failed	Неудачная попытка исправления данных
5	WP Violation	Нарушение защиты записи. Команда пыталась записать блок, защищённый от записи
6	Erase Parameter	Неправильный выбор области для стирания блоков данных
7	Out of Range CSD Overwrite	Выход за диапазон или карта переполнена

и CMD58. Любые другие команды будут отклонены. Когда карта обнаруживает команду CMD1, она начинает инициализацию. Чтобы определить, завершена ли инициализация, микроконтроллер должен повторять посылку команды CMD1 и проверять ответ. Как только карта успешно инициализировалась, бит In Idle State в ответе R1 будет сброшен. Процесс инициализации может занять несколько сотен миллисекунд. Чем больше объём карты памяти, тем больше времени требуется на её инициализацию. Команды чтения и записи начнут восприниматься картой памяти после окончания инициализации. В это время можно считать содержимое регистров OCR и CID для подтверждения диапазона рабочего напряжения, объёма карты или другого необходимого параметра карты памяти.

Команда CMD1 воспринимается не всеми типами карт памяти SD. Поэтому, если команда CMD1 была отклонена картой, или если вышло время ожидания при опросе состояния покоя, необходимо выполнить повторную передачу команды, но уже в формате ACMD41.

Таблица 5. Формат пакета данных

Маркер данных Data Token	Блок данных Data Block	Контрольная сумма CRC
1 байт	От 1 до 2048 байт	2 байта

ОБМЕН ДАННЫМИ С КАРТОЙ ПАМЯТИ

В процессе обмена данными с картой памяти, после ответа на команду может быть получен один или несколько пакетов данных. Пакет данных состоит из маркера данных (DataToken), блока данных (Data Block) и контрольной суммы CRC. Формат пакета данных приведён в таблице 5.

Существует три значения маркера данных, означающих следующее:

- 0xFE – маркер данных для команд CMD17, CMD18 и CMD24;
- 0xFC – маркер данных для команды CMD25;
- 0xFD – маркер останова транзакции Stop Tran для команды CMD25.

Последний маркер Stop Tran в данном списке означает конец многоблочной записи и используется в виде одиночного байта без блока данных и CRC.

ОДНОБЛОЧНОЕ ЧТЕНИЕ

Одноблочное чтение инициируется командой CMD17. Обмен данными между картой памяти и микроконтроллером будет производиться стандартными блоками по 512 байт. Аргументом команды является адрес начала блока, содержимое которого следует читать. Адресация карты побайтная, начиная с нуля, но считывать данные можно только блоками. Адресом блока служит первый его байт. Для блоков размером в 512 байт нулевой блок имеет адрес 0x0000, первый блок – 0x0200, второй блок – 0x400 и т.д. В картах памяти типа SDHC адресация поблочная, и адресом блока служит его номер.

В ответ на команду CMD17 карта выдаёт ведущему контроллеру пакет данных. После обнаружения правильного маркера данных микроконтроллер принимает следующий за ним блок данных и два байта CRC, которые



Разъёмы Bulgin для промышленной автоматизации

Данные, сигналы, питание

- M5, M8, M12, M16, M23 серии
- A, B и D кодирование для всех серий
- Степень защиты IP67 от влаги и пыли
- Винтовой запирающий механизм промышленного стандарта
- Число контактов от 3 до 12

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР ПРОДУКЦИИ BULGIN

ProCHIP
POWERED BY ProSOFT

Активный компонент вашего бизнеса

ТЕЛ.: (495) 232-2522 / ФАКС: (495) 234-0640 / INFO@PROCHIP.RU / WWW.PROCHIP.RU



Реклама

Листинг 1

```

// Подключение библиотек
#include "stm32f10x.h"
// Определение макрокоманд
#define GO_IDLE_STATE 0 // Программный сброс карты
#define SEND_IF_COND 8 // Проверка напряжений питания
#define READ_SINGLE_BLOCK 17 // Чтение блока данных
#define WRITE_SINGLE_BLOCK 24 // Запись блока данных
#define SD_SEND_OP_COND 41 // Инициализация карты
#define APP_CMD 55 // Главная команда из ACMD команд
#define READ_OCR 58 // Чтение регистра OCR
#define CS_ENABLE GPIOA->BSRR = GPIO_BSRR_BR4; // Сигнал CS=0
#define CS_DISABLE GPIOA->BSRR = GPIO_BSRR_BS4; // Сигнал CS=1
// Определение структур
GPIO_InitTypeDef GPIO_InitStructure; // Структура для настройки пор-
тов ввода-вывода
USART_InitTypeDef USART_InitStructure; // Структура для настройки
порта UART
// Определение переменных
uint8_t SDHC; // Глобальная переменная для типа карты
uint8_t Buff[512]; // Буфер данных для SD карты
// Главный модуль программы
int main(void)
{
    unsigned long address,i;
    unsigned char c;
    init(); // Инициализация портов микроконтроллера
    // Инициализация карты памяти SD
    delay(1000000); // Пауза
    if(SD_init()==0) send_Uart_str(USART1,"Карта SD подключена\n");
    else send_Uart_str(USART1,"Карта SD не обнаружена\n");
    while(1) // Бесконечный цикл
    {
        send_Uart_str(USART1,"Введите команду(w-запись r-чтение):");
        c=getch_Uart(USART1); // Читать байт из порта UART
        while(getch_Uart(USART1)!=13) {} // Ждать ввода <enter>
        send_Uart_str(USART1,"\nВведите адрес блока SD карты:");
        buff_clear(); // Очистка буфера
        address=read_int_uart(USART1); // Принять адрес блока SD карты
        send_Uart_str(USART1,"\nВведен адрес:");
        send_int_uart(USART1,address); // Вывод введенного адреса для про-
        верки
        send_Uart(USART1,'\n');
        if(c=='w') // Если принята команда записи карты памяти
        {
            send_Uart_str(USART1,"Введите не более 512 символов завершив ввод
            клавишей Enter\n");
            read_str_uart(USART1,Buff); // Принять строку с порта UART
            i=0;
            while((i<512)&&(Buff[i]!=0)) i++; // Найти конец текстовой строки
            send_Uart_str(USART1,"Длина текста=");
            send_int_uart(USART1,i); // Вывести длину текстовой строки
            send_Uart(USART1,'\n');
            // Запись данных из буфера на SD карту
            if(SD_WriteSector(address, Buff)==0) send_Uart_str(USART1,"Запись
            завершена\n");
            else send_Uart_str(USART1,"Запись не удалась\n");
        }
        else // Чтение блока SD карты в буфер
        {
            if(SD_ReadSector(address, Buff)==0) send_Uart_str(USART1,"Чтение
            завершено\n");
            else send_Uart_str(USART1,"Чтение не удалось\n");
            // Вывод содержимого буфера в порт UART
            for(i=0;i<512;i++) send_Uart(USART1,Buff[i]);
        }
    }
}

```

необходимо принять, даже если CRC не используется. По умолчанию размер блока равен 512 байт. Размер блока можно изменить командой CMD16, но это применимо не для всех типов карт памяти. Если во время операции чтения произошла какая-нибудь ошибка, вместо пакета данных будет возвращён маркер ошибки, состоящий из одного байта. Формат данных маркера ошибки приведён в таблице 6.

Многоблочное чтение

Существует возможность прочитать из карты памяти несколько блоков данных подряд, начиная с заданного адреса. Для данной процедуры используется команда многоблочного чтения CMD18. Для карт MMC, перед командой чтения CMD18, можно с помощью команды CMD23 задать количество передаваемых блоков. Если количество блоков не задано, то будет инициировано неограниченное многоблочное чтение. Т.е. операция чтения будет продолжаться до тех пор, пока микроконтроллер не прервёт её командой CMD12. Получаемый сразу же после передачи команды CMD12 байт является дополняющим, поэтому его не нужно учитывать. После этого байта следует ответ на команду.

Одноблочная запись

Для одноблочной записи данных в карту памяти используется команда CMD24. После её получения карта памяти передаёт пакет данных после небольшой задержки, составляющей время передачи одного или нескольких байт. Формат пакета для одноблочной записи такой же, как у команды блочного чтения. После передачи пакета карта сразу же выдаёт ответ на данные (Data Response), за которым следует флаг занятости. Большинство карт памяти не могут изменять размер записываемого блока, поэтому он является фиксированным и составляет 512 байт.

Многоблочная запись

С помощью команды многоблочной записи CMD25 можно записать последовательность из нескольких блоков, начиная с заданного адреса. Если перед этой командой число передаваемых блоков не было задано с помощью команды CMD23 (только для карт MMC, или ACMD23 – для карт SD), транзакция будет инициирована как неограниченная многоблочная запись, т.е. операция записи будет продолжаться, пока

Таблица 6. Формат данных маркера ошибки

Бит	Имя	Назначение
0	Error	Во время операции произошла ошибка
1	CC Error	Внутренняя ошибка контроллера карты памяти
2	Card ECC Failed	Неудачная попытка исправления данных
3	Out of Range	Выход за диапазон
4	Card is Locked	Карта памяти заблокирована
5	0	Всегда нулевой бит
6	0	Всегда нулевой бит
7	0	Всегда нулевой бит

микроконтроллер не прервёт её передачей маркера Stop Tran.

ЧТЕНИЕ РЕГИСТРОВ CSD И CID КАРТЫ ПАМЯТИ

Команда CMD9 чтения регистра CSD (Card Specific Data) и команда CMD10 чтения регистра CID (Card Identification Data) подобны одноблочному чтению, за исключением длины блока – этот параметр отсутствует. Содержимое регистров CSD и CID передаётся ведущему контроллеру в виде 16-байтных блоков данных.

Регистр CID содержит данные, по которым можно идентифицировать карту памяти. К этим данным относятся: серийный номер, ID производителя, дата изготовления и т.д.

Регистр CSD содержит информацию о таких характеристиках карты памяти, как размера блока карты памяти, потребление в режиме чтения и записи и т.п.

Регистр OCR (Operation Conditions Register) содержит информацию о напряжении питания карты памяти, типе питания карты памяти, статусе процесса инициализации карты и т.п.

Более детальную информацию о регистрах CSD и CID можно найти в спецификации [2].

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

В качестве демонстрационного примера рассмотрим небольшую программу (см. Листинг 1). С помощью этой программы можно организовать ручную запись и чтение строк текста в карту памяти по заданному адресу блока в режиме диалога. Данная программа организует простой интерфейс с человеком через порт UART. Подключив этот порт к компьютеру и используя любую подходящую терминальную программу, можно управлять процессом записи и чтения данных в карту памяти. Данная программа снабжена подробными комментариями, позволяющими понять её работу.

В приведённой программе использован набор функций для настройки

портов и обращения с картой памяти. Данный набор функций приведён в листинге 2 (см. дополнительные

материалы к статье на сайте журнала www.soel.ru).

С помощью приведённого набора функций можно создавать программы, использующие карты памяти в качестве энергонезависимых носителей произвольных данных. Такие задачи очень часто возникают при разработке современных микропроцессорных устройств.

ЛИТЕРАТУРА

1. www.st.com.
2. www.sdcard.org/downloads/pls/pdf/part1_500.pdf.



Магнитодиэлектрики MICROMETALS

Применение сердечников Micrometals гарантирует:

- снижение стоимости индуктивных компонентов
- повышение надёжности аппаратуры
- снижение потерь на 30...50% по сравнению с ферритами
- оптимизацию конструкции и уменьшение габаритов индуктивных компонентов



PROCHIP
POWERED BY PROSOFT

Активный компонент вашего бизнеса
ТЕЛ.: (495) 232-2522 / INFO@PROCHIP.RU / WWW.PROCHIP.RU



Реклама