



MicroSD 卡 (SDIO 模式) 实验

| | |
|--------|------------------------------|
| 作者 | fire |
| E-Mail | firestm32@foxmail.com |
| QQ | 313303034 |
| 博客 | firestm32.blog.chinaunix.net |
| 硬件平台 | 野火 STM32 开发板 |
| 库版本 | ST3.0.0 |

实验描述: MicroSD 卡(SDIO 模式)测试实验, 没有跑文件系统, 只是单纯地读 block 并将测试信息通过串口 1 在电脑的超级终端上 打印出来。

硬件连接: PC8-SDIO-D0 : DATA0
PC9-SDIO-D1 : DATA1
PC10-SDIO-D2 : DATA2
PC11-SDIO-D3 : CD/DATA3
PC12-SDIO-CLK: CLK
PD2-SDIO-CMD : CMD

库文件 : startup/start_stm32f10x_hd.c
CMSIS/core_cm3.c
CMSIS/system_stm32f10x.c
FWlib/stm32f10x_gpio.c
FWlib/stm32f10x_rcc.c
FWlib/stm32f10x_usart.c
FWlib/stm32f10x_sdio.c
FWlib/stm32f10x_dma.c
FWlib/misc.c



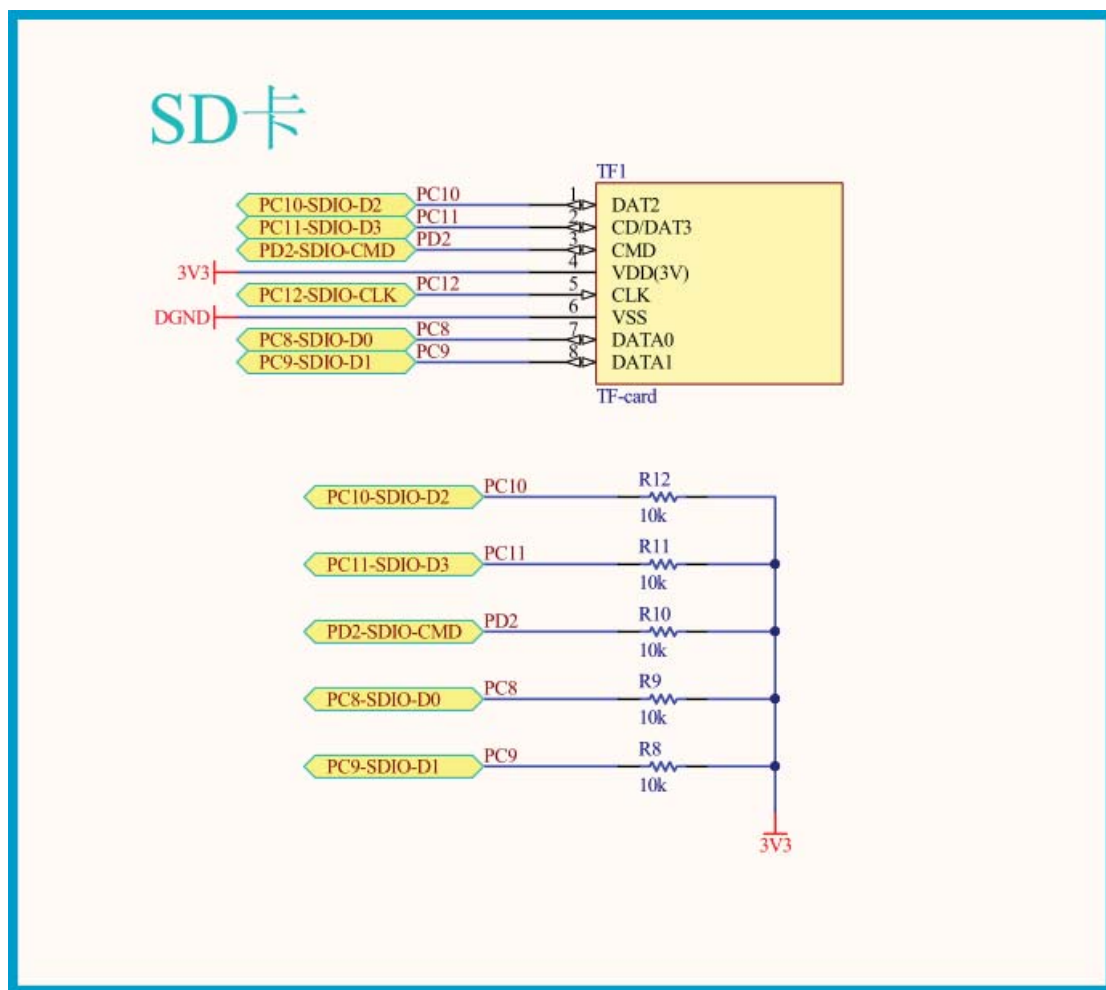
用户文件: USER/main.c

USER/stm32f10x_it.c

USER/usart.c

USER/sdcard.c

野火 STM32 开发板 MicroSD 卡硬件原理图:



SDIO 简介->

野火 STM32 开发板的 CPU (STM32F103VET6)具有一个 SDIO 接口。

SD/SDIO/MMC 主机接口可以支持 MMC 卡系统规范 4.2 版中的 3 个不同的数据总线模式: 1 位(默认)、4 位和 8 位。在 8 位模式下, 该接口可以使数据传输速率达到



48MHz, 该接口兼容 SD 存储卡规范 2.0 版。SDIO 存储卡规范 2.0 版支持两种数据总线模式: 1 位(默认)和 4 位。

目前的芯片版本只能一次支持一个 SD/SDIO/MMC 4.2 版的卡, 但可以同时支持多个 MMC 4.1 版或之前版本的卡。除了 SD/SDIO/MMC, 这个接口完全与 CE-ATA 数字协议版本 1.1 兼容。

本实验用的是 MicroSD 卡 1 位(默认)模式, 速率为 1M/S, MicroSD 的容量为 1G。

实验讲解->

首先在工程中添加我们所需要的库文件:

```
FWlib/stm32f10x_gpio.c
FWlib/stm32f10x_rcc.c
FWlib/stm32f10x_usart.c
FWlib/stm32f10x_sdio.c
FWlib/stm32f10x_dma.c
FWlib/misc.c
```

然后在 stm32f10x_conf.h 这个头文件中将用到的库的头文件的注释去掉:

```
1. /* Includes -----
   */
2. /* Uncomment the line below to enable peripheral header file inclusion */
3. /* #include "stm32f10x_adc.h" */
4. /* #include "stm32f10x_bkp.h" */
5. /* #include "stm32f10x_can.h" */
6. /* #include "stm32f10x_crc.h" */
7. /* #include "stm32f10x_dac.h" */
8. /* #include "stm32f10x_dbgmcu.h" */
9. #include "stm32f10x_dma.h"
10. /* #include "stm32f10x_exti.h" */
11. /*#include "stm32f10x_flash.h"*/
12. /* #include "stm32f10x_fsmc.h" */
13. #include "stm32f10x_gpio.h"
14. /* #include "stm32f10x_i2c.h" */
15. /* #include "stm32f10x_iwdg.h" */
16. /* #include "stm32f10x_pwr.h" */
17. #include "stm32f10x_rcc.h"
18. /* #include "stm32f10x_rtc.h" */
19. #include "stm32f10x_sdio.h"
20. /* #include "stm32f10x_spi.h" */
21. /* #include "stm32f10x_tim.h" */
22. #include "stm32f10x_usart.h"
23. /* #include "stm32f10x_wwdg.h" */
24. #include "misc.h" /* High level functions for NVIC and SysTick (add-
25.                   on to CMSIS functions) */
```

配置完需要用到的头文件之后, 我们将从 main 函数开始分析:



```
1.  /*
2.  * 函数名: main
3.  * 描述   : 无
4.  * 输入   : 无
5.  * 输出   : 无
6.  */
7.  int main(void)
8.  {
9.      /* config the sysclock to 72M */
10.     SystemInit();
11.
12.     /* USART1 config */
13.     USART1 Config();
14.
15.     /* Interrupt Config */
16.     NVIC_Configuration();
17.
18.     printf( "\r\n 这是一个MicroSD卡实验(没有跑文件系统)..... " );
19.
20.     /*----- SD Init ----- */
21.     Status = SD_Init();
22.
23.     if (Status == SD_OK)
24.     {
25.         /*----- Read CSD/CID MSD registers -----*/
26.         printf( " \r\n SD_Init is ok " );
27.         Status = SD_GetCardInfo(&SDCardInfo);
28.     }
29.
30.     //printf( " \r\n SD_GetCardInfo Status is: %d ", Status );
31.     printf( " \r\n CardType is : %d ", SDCardInfo.CardType );
32.     printf( " \r\n CardCapacity is : %d ", SDCardInfo.CardCapacity );
33.     printf( " \r\n CardBlockSize is : %d ", SDCardInfo.CardBlockSize );
34.     printf( " \r\n RCA is : %d ", SDCardInfo.RCA);
35.     printf( " \r\n ManufacturerID is : %d ", SDCardInfo.SD_cid.ManufacturerID )
36. ;
37.     if (Status == SD_OK)
38.     {
39.         printf("\r\n SD_GetCardInfo is ok ");
40.         /*----- Select Card -----*/
41.         Status = SD_SelectDeselect((uint32_t) (SDCardInfo.RCA << 16));
42.     }
43.
44.     if (Status == SD_OK)
45.     {
46.         printf(" \r\n SD_SelectDeselect is ok ");
47.         // SDIO_BusWide_4b 会进入死循环,至今还未解决
48.         Status = SD_EnableWideBusOperation(SDIO_BusWide_1b);
49.     }
50.
51.     /*----- Block Erase -----*/
52.     if (Status == SD_OK)
53.     {
54.         /* Erase NumberOfBlocks Blocks of WRITE_BL_LEN(512 Bytes) */
55.         Status = SD_Erase(0x00, (BlockSize * NumberOfBlocks));
56.     }
57.
58.     if (Status == SD_OK)
59.     {
60.         printf(" \r\n SD_Erase is ok ");
61.         Status = SD_SetDeviceMode(SD_DMA_MODE);
62.     }
63.
64.     if (Status == SD_OK)
65.     {
66.         printf(" \r\n SD_SetDeviceMode is ok ");
67.         // 4bit 模式时会停在这里了,进入死循环,1bit 模式时则可以读取成功
68.         Status = SD_ReadMultiBlocks(0x00, Buffer_MultiBlock_Rx, BlockSize, NumberOfBlocks);
69.     }
70.
71.     if (Status == SD_OK)
72.     {
```



```
73.     printf(" \r\n SD ReadMultiBlocks is ok ");
74.     EraseStatus = eBuffercmp(Buffer_MultiBlock_Rx, MultiBufferWordsSize);
75.     printf( " \r\n eBuffercmp EraseStatus is %d ", EraseStatus );
76. }
77.
78. /*----- Block Read/Write -----*/
79. /* Fill the buffer to send */
80. Fill_Buffer(Buffer_Block_Tx, BufferWordsSize, 0x12345678);
81.
82.
83. if (Status == SD_OK)
84. {
85.     /* Write block of 512 bytes on address 0 */
86.     Status = SD_WriteBlock(0x00, Buffer_Block_Tx, BlockSize);
87. }
88.
89. if (Status == SD_OK)
90. {
91.     printf( " \r\n SD WriteBlock is ok " );
92.     /* Read block of 512 bytes from address 0 */
93.     Status = SD_ReadBlock(0x00, Buffer_Block_Rx, BlockSize);
94. }
95.
96. if (Status == SD_OK)
97. {
98.     printf(" \r\n SD ReadBlock is ok ");
99.     /* Check the corectness of written dada */
100.    TransferStatus1 = Buffercmp(Buffer_Block_Tx, Buffer_Block_Rx, Buffer
    rWordsSize);
101.    }
102.
103.    if (TransferStatus1 == PASSED)
104.    {
105.        /* 写进去的值和读出来的值相同 */
106.        printf( "\r\n Single Block Read/Write is ok " );
107.    }
108.
109.    /*----- Multiple Block Read/Write -----*/
110.    /* Fill the buffer to send */
111.    Fill Buffer(Buffer MultiBlock Tx, MultiBufferWordsSize, 0x0);
112.
113.    if (Status == SD_OK)
114.    {
115.        /* Write multiple block of many bytes on address 0 */
116.        Status = SD_WriteMultiBlocks(0x00, Buffer_MultiBlock_Tx, BlockSize,
    NumberOfBlocks);
117.    }
118.
119.    if (Status == SD_OK)
120.    {
121.        /* Read block of many bytes from address 0 */
122.        Status = SD ReadMultiBlocks(0x00, Buffer MultiBlock Rx, BlockSize,
    NumberOfBlocks);
123.    }
124.
125.    if (Status == SD_OK)
126.    {
127.        /* Check the corectness of written dada */
128.        /* 写进去的值和读出来的值相同 */
129.        TransferStatus2 = Buffercmp(Buffer MultiBlock Tx, Buffer MultiBlock
    _Rx, MultiBufferWordsSize);
130.    }
131.
132.    if ( TransferStatus2 == PASSED )
133.    {
134.        printf( " \r\n Multiple Block Read/Write is ok " );
135.    }
136.
137.    printf(" \r\n 好消息: MicroSD 卡读写测试成功 \r\n ");
138.
139.    while (1)
140.    {
141.    }
142.    } /* end of main */
```



`main` 函数有点长, 但认真分析下还是很好理解的, 大伙给点耐心, 用到的很多都是库操作, 都是 ST 的工程师为用户写好的。

首先, 我们还是一如既往地先调用我们的库函数 `SystemInit()`; 将我们的系统时钟设置为 72MHZ。函数 `USART1_Config()`; 用来初始化我们要用到的串口。关于这两个函数的详细讲解可参考前面的教程, 这里不在详述。

`NVIC_Configuration()`; 用来设置 SDIO 的中断优先级, 因为 SDIO 有三种工作模式: `SD_DMA_MODE`、`SD_POLLING_MODE`、`SD_INTERRUPT_MODE`, 其中工作中断模式时就需要设置中断优先级, 本实验用的是 SDIO 的 DMA 模式, 所以本实验不用调用这个函数也可以, 调用了也无所谓。

接下来的代码是与 MicroSD 卡的操作有关的, 与 MicroSD 卡有关的函数都在 `sdcard.c` 这个文件中实现, `sdcard.c` 不是库文件, 而是需要用户编写的。

1-> 首先调用 `SD_Init()`; 用于初始化 MicroSD 卡, 使卡处于就绪状态(准备传输数据)。

2-> 初始化 MicroSD 卡成功之后, 调用函数 `SD_GetCardInfo(&SDCardInfo)`; 来获取与卡的属性有关的信息。紧接着通过串口将这些信息在电脑的超级终端中打印出来。

3-> 获取到卡的正确信息后, 调用函数

`SD_SelectDeselect((uint32_t) (SDCardInfo.RCA << 16))`; 来选中这张卡。

4-> 确定好目标卡之后, 调用函数 `SD_EnableWideBusOperation(SDIO_BusWide_1b)`;

设定卡的数据位的宽度, 这里设置为 1 位, 之所以设置为 1 位而不是 4 位, 是因为 4 位模式我还没有调试成功, 仍需努力呀!

5-> 紧接着就调用函数 `SD_SetDeviceMode(SD_DMA_MODE)`; 设置 SDIO 的工作模式为 DMA 方式。



6->到了这里与卡有关的初始化都已完成, 假如程序能运行到这里的话, 卡的操作基本上已经成功了一半。

7->Block Read/Write 和 Multiple Block Read/Write 测试, 将数据写到卡里面去, 再将数据读出来, 如果写进去的和读出来的数据相同, 即为成功。

8->最后打印调试信息。

注意: 这个例程是没有跑文件系统的, 而是直接就去读卡的 **block**, 这样的话就会破坏卡的分区, 在实验完成之后, 你再把卡插到电脑上时, 电脑会提示你要重新初始化卡, 这是正常想象, 并不是本实验把你的卡弄坏了。但跑文件系统时就不会出现这种问题, 有关文件系统的操作将在下一讲的教程中讲解。



实验现象->

将野火 STM32 开发板供电(DC5V), 插上 JLINK, 插上串口线(两头都是母的交叉线), 插上 MicroSD 卡(我用的是 1G), 打开超级终端, 配置超级终端为 115200 8-N-1, 将编译好的程序下载到开发板, 即可看到超级终端打印出如下信息:

```
123 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(I) 帮助(H)
这是一个MicroSD卡实验(没有跑文件系统).....
SD_Init is ok
CardType is : 1
CardCapacity is : 966787072
CardBlockSize is : 512
RCA is : 60268
ManufacturerID is : 2
SD_GetCardInfo is ok
SD_SelectDeselect is ok
SD_Erase is ok
SD_SetDeviceMode is ok
SD_ReadMultiBlocks is ok
eBuffercmp EraseStatus is 0
SD_WriteBlock is ok
SD_ReadBlock is ok
Single Block Read/Write is ok
Multiple Block Read/Write is ok
好消息: MicroSD卡读写测试成功
-
已连接 1:30:24 ANSIW 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

实验讲解完毕, 野火祝大家学习愉快^_^。