

# 第一章 EMMC 及 SD 卡测试实验

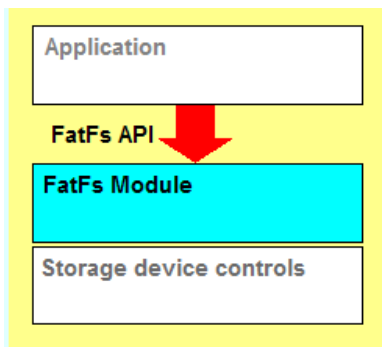
实验工程为“ps\_emmc”

本章介绍使用 FatFs 文件系统模块向 EMMC 和 SD 卡写入一个文件，并读出其内容。

## 1.1 FatFs 简介

FatFs 是一个通用的文件系统模块，用于在小型嵌入式系统中实现 FAT 文件系统。FatFs 的编写遵循 ANSI C，因此不依赖于硬件平台。它可以嵌入到便宜的微控制器中，如 8051, PIC, AVR, SH, Z80, H8, ARM 等等，不需要做任何修改。

应用程序通过 API 函数来调用 FatFs 系统模块，从而来控制 SD 卡这些存储设备。



FatFs 系统提供了很多 API 函数，我们在下面列举了以下我们例程中会用的 API 函数。

- f\_mount - 注册/注销一个工作区域 (Work Area)
- f\_open - 打开/创建一个文件
- f\_close - 关闭一个文件
- f\_read - 读文件
- f\_write - 写文件

关于 API 函数的介绍和说明，大家可以参考以下的网站进行更深一步的了解，这个网站上

给出来了每个 API 函数的使用说明和例子。

[http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)

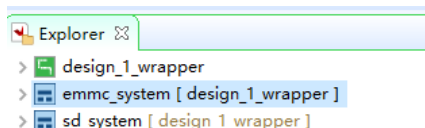
## 1.2 硬件工程的建立

利用 “ps\_hello”工程另存一个工程，工程中已经配置了 SD 卡和 EMMC，这里不再配置。

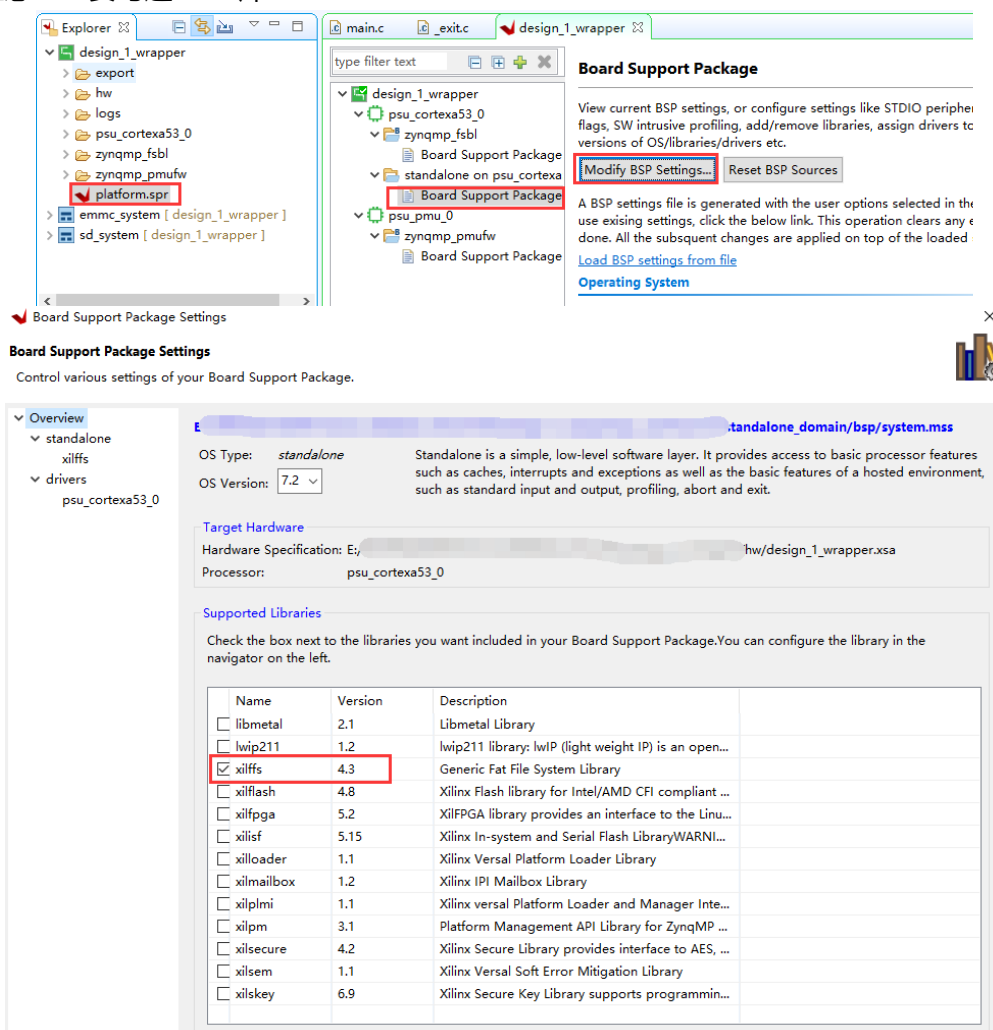
导出硬件信息，不包含 bitstream

## 1.3 Vitis 程序编写

1. 例程中提供了 EMMC 和 SD 卡的测试程序，程序流程为注册工作空间→检查状态→检查是否驱动准备好，文件系统是否存在，如果不存在，则创建→检查目录文件是否存在，如果有，则删除→写入一个文件→读出文件内容。



2. 注意 BSP 要勾选 xilffs 库



3. 在 main 函数中, 进行工作空间注册, 文件系统检查, 如果没有文件系统的时候需要创建文件系统, 利用 f\_mkfs 函数, 如果是 SD0, 路径就是"0:", 如果是 SD1, 路径就是"1:", 创建文件系统需要较长时间, 要耐心等待!

```
int main()
{
    FRESULT rc;
    FILINFO fno;
    int i;
    BYTE work[FF_MAX_SS];

    /*
     * Register the work area of the volume
     */
    rc = f_mount(&fatfs, FILE_PATH, 0);
    if (rc != FR_OK)
    {
        xil_printf("mount failed!\r\n");
        return 0;
    }
    /*
     * Check existence of a file or sub-directory
     */
    rc = f_stat(FILE_PATH, &fno);

    if (rc != FR_OK)
    {
        /*
         * Check if filesystem exist
         */
        if (rc == FR_NO_FILESYSTEM)
        {
            xil_printf("Creating a file system...\r\n");
            /*
             * Create an FAT volume on the logical drive
             */
            rc = f_mkfs("0:", FM_FAT32, 0, work, sizeof(work));
            if (rc != FR_OK)
            {
                xil_printf("error:Create a file system fail %d\r\n", rc);
                return 0;
            }

            rc = f_stat(FILE_PATH, &fno);
            xil_printf("Successfully Create a file system!\r\n");
        }
    }
}
```

4. 再判断是否文件存在, 存在的话, 先删除掉, 然后向其中写入一个文件, 并再读出来。

```
if (rc == FR_OK)
{
    /*
     * if file existed,remove it
     */
    rc = f_unlink(FILE_PATH);
    if (rc != FR_OK)
    {
        xil_printf("delete file failed!\r\n");
        return 0;
    }
    rc = f_stat(FILE_PATH, &fno);
    xil_printf("Successfully delete existed file!\r\n");
}

/*
 * if file does not exist, create a new one and write data
 */
if (rc == FR_NO_FILE)
{
    xil_printf("Object file is not exist!\r\n");
    file_write(&fil, FILE_PATH, writebuf, sizeof(writebuf), FA_CREATE_ALWAYS | FA_WRITE);
    xil_printf("Successfully create file!\r\n");
}

/*
 * Read data from file
 */
rc = f_stat(FILE_PATH, &fno);
if (rc == FR_OK)
{
    file_read(&fil, FILE_PATH, readbuf, FA_OPEN_EXISTING | FA_READ);
    xil_printf("File Contents: %s\r\n", readbuf);
    for (i = 0; i < sizeof(writebuf); i++)
    {
        if (readbuf[i] != writebuf[i])
        {
            xil_printf("error:File Contents is not correct!\r\nWrite data is %s\r\nRead data is %s\r\n", writebuf, readbuf);
            return 0;
        }
    }
}

return 1;
}
```

5. 定义文件路径, 文件内容, 文件路径要注意, 如果 SD0 就是 "0:/", 如果是 SD1 就

是"1:/", 对于本工程, SD0 就是 EMMC, SD1 就是 SD 卡。

```
/*  
 * define file path  
 */  
#define FILE_PATH "1:/1.txt" 文件路径  
static FIL fil; /* File object */  
static FATFS fatfs;  
/*  
 * write and read buffer  
 */  
char writebuf[] = "Hello Alinx!"; 文件内容  
char readbuf[200];
```

## 1.4 下载调试

### 1.4.1 EMMC 测试

1. 下载 EMMC 的程序到开发板, 则会打印以下信息, 如果 EMMC 中已经存在目录文件, 则会打印第一条语句, 最后一条语句则是打印文件中的内容。

```
Successfully delete existed file!  ←  
Object file is not exist!  
Successfully create file!  
File Contents: Hello Alinx!  ←
```

### 1.4.2 SD 卡测试

1. 同样的方法将 SD 卡程序下载到开发板, 也会打印与 EMMC 相同的内容。
2. 可以将 SD 卡取出, 用读卡器在电脑上读出

| 名称    | 修改日期          | 类型   | 大小   |
|-------|---------------|------|------|
| 1.TXT | 2010/1/1 0:00 | 文本文档 | 1 KB |

```
1.TXT - 记事本  
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)  
Hello Alinx!
```