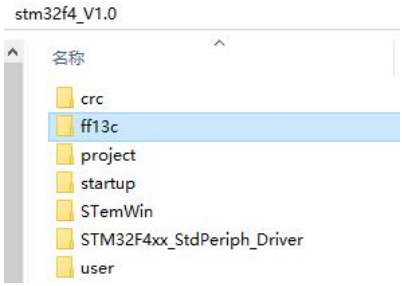
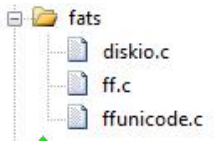
Fats 移植文档

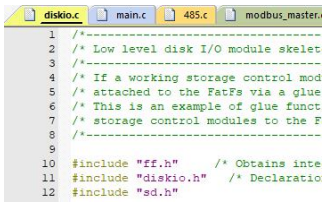
1 拷贝文件到工程



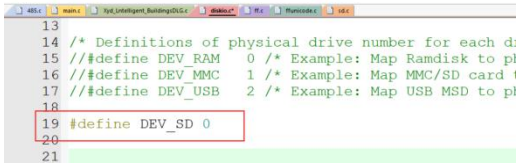
2 添加.c 文件和.h 文件



3 包含头文件　　将存储器的底层驱动文件.h，包含到底层接口文件里

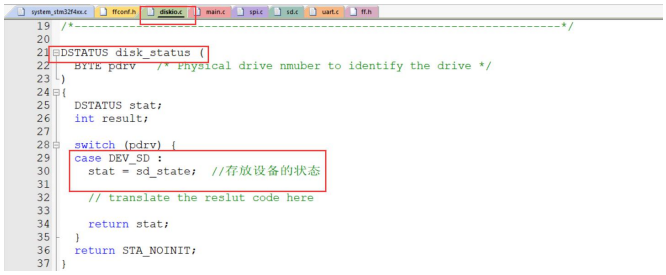


修改 diskio.c　　主要是做一些底层适配



#define DEV_SD　　0

获取 SD 卡状态



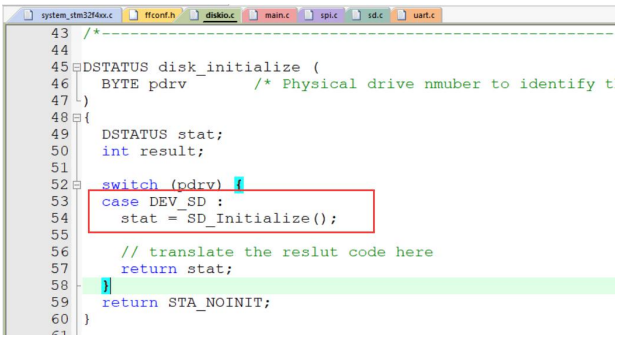
case DEV_SD :

stat = sd_state; //存放设备的状态

// translate the reslut code here

return stat;

SD 卡初始化

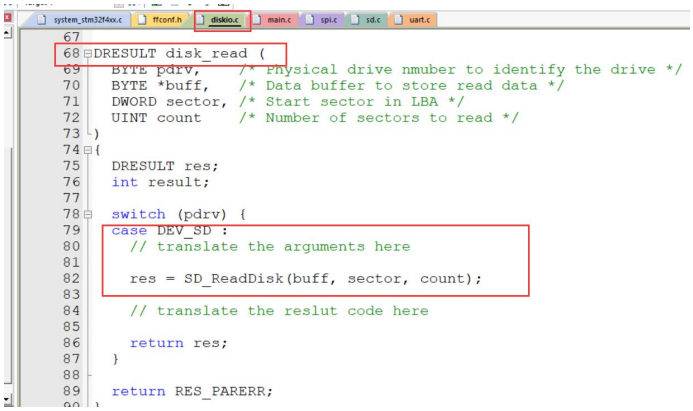


```
/*-----------------------------------------------------------------------*/

DSTATUS disk_initialize (
    BYTE pdrv       /* Physical drive nmuber to identify t
)
{
    DSTATUS stat;
    int result;

    switch (pdrv) {
    case DEV_SD :
      stat = SD_Initialize();

      // translate the reslut code here
      return stat;
    }
    return STA_NOINIT;
}
```

case DEV_SD :

stat = SD_Initialize();

文件系统底层读取函数

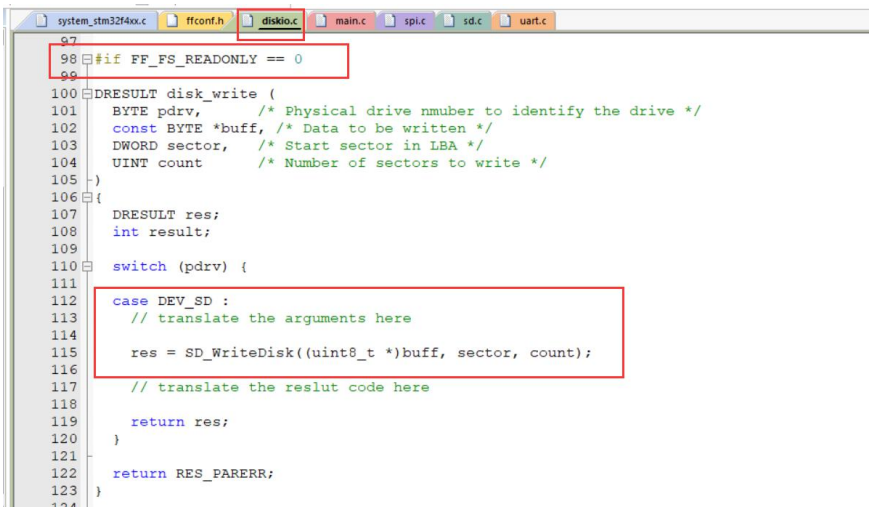


```
DRESULT disk_read (
    BYTE pdrv,      /* Physical drive nmuber to identify the drive */
    BYTE *buff,     /* Data buffer to store read data */
    DWORD sector,   /* Start sector in LBA */
    UINT count      /* Number of sectors to read */
)
{
    DRESULT res;
    int result;

    switch (pdrv) {
    case DEV_SD :
      // translate the arguments here

      res = SD_ReadDisk(buff, sector, count);

      // translate the reslut code here

      return res;
    }

    return RES_PARERR;
}
```

case DEV_SD :

res = SD_ReadDisk(buff, sector, count);

文件系统底层写函数



```
#if FF_FS_READONLY == 0

DRESULT disk_write (
    BYTE pdrv,      /* Physical drive nmuber to identify the drive */
    const BYTE *buff, /* Data to be written */
    DWORD sector,   /* Start sector in LBA */
    UINT count      /* Number of sectors to write */
)
{
    DRESULT res;
    int result;

    switch (pdrv) {

    case DEV_SD :
      // translate the arguments here

      res = SD_WriteDisk((uint8_t *)buff, sector, count);

      // translate the reslut code here

      return res;
    }

    return RES_PARERR;
}
```

case DEV_SD :

// translate the arguments here

res = SD_WriteDisk((uint8_t *)buff, sector, count);

文件系统底层获取存储器信息函数和获取时间的函数

```
131
132 DRESULT disk_ioctl (
133   BYTE pdrv,     /* Physical drive nmuber (0..) */
134   BYTE cmd,   /* Control code */
135   void *buff     /* Buffer to send/receive control data */
136 )
137 {
138   DRESULT res;
139   int result;
140
141   switch (pdrv) {
142     case GET_SECTOR_COUNT:*(uint8_t *)buff = sd_info.SD_csd.DeviceSize;break;
143     case GET_SECTOR_SIZE:*(uint8_t *)buff = sd_info.CardBlockSize;break;
144     case GET_BLOCK_SIZE:*(uint8_t *)buff = sd_info.CardBlockSize;break;
145   }
146
147   return RES_PARERR;
148 }
149 DWORD get_fattime (void)
150 {
151   return 0;
152 }
```

```
switch (pdrv) {
        case GET_SECTOR_COUNT:*(uint8_t *)buff = sd_info.SD_csd.DeviceSize;break;
        case GET_SECTOR_SIZE:*(uint8_t *)buff = sd_info.CardBlockSize;break;
        case GET_BLOCK_SIZE:*(uint8_t *)buff = sd_info.CardBlockSize;break;
    }
DWORD get_fattime (void)
{return 0;}
```

修改 ffconf.h

使用基础函数

```
25 /   3: f_lseek() function is removed in addition to 2.
26
27
28 #define FF_USE_STRFUNC  1
29 /* This option switches string functions, f_gets(), f_
30 /
```

定义可以使用什么类型的编码格式

```
70
71 #define FF_CODE_PAGE  936
72 /* This option specifies the OEM code page to be used on the target
73 /  Incorrect code page setting can cause a file open failure.
74 /
```