

零死角玩转STM32



串行FLASH文件系统 FatFs

淘宝：firestm32.taobao.com

论坛：www.firebbs.cn



扫描进入淘宝店铺

01

文件系统简介

02

FatFs文件系统简介

03

FatFs文件系统移植实验

04

FatFs功能使用实验

参考资料:《零死角玩转STM32》

“SPI—串行FLASH文件系统FatFs” 章节

串行FLASH文件系统FatFs



使用SPI FLASH直接存储数据

当需要记录字符“STM32 SPI FLASH”时。可以把这些文字转化成ASCII码，存储在数组中，然后调用SPI_FLASH_BufferWrite函数，把数组内容写入到SPI Flash芯片的指定地址上，在需要的时候从该地址把数据读取出来，再对读出来的数据以ASCII码的格式进行解读。

	A	B	C	D	E	F	G
1	S	T	M	3	2	-	F
2	4	2	9	S	P	I	
3	F	L	A	S	H		
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							

这样做会出现以下问题：

- 难以记录有效数据的位置
- 难以确定存储介质的剩余空间
- 不明确应以何种格式来解读数据

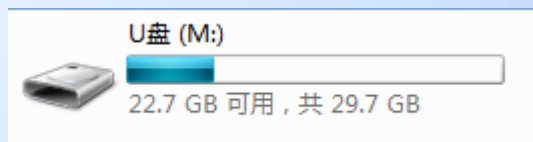
串行FLASH文件系统FatFs



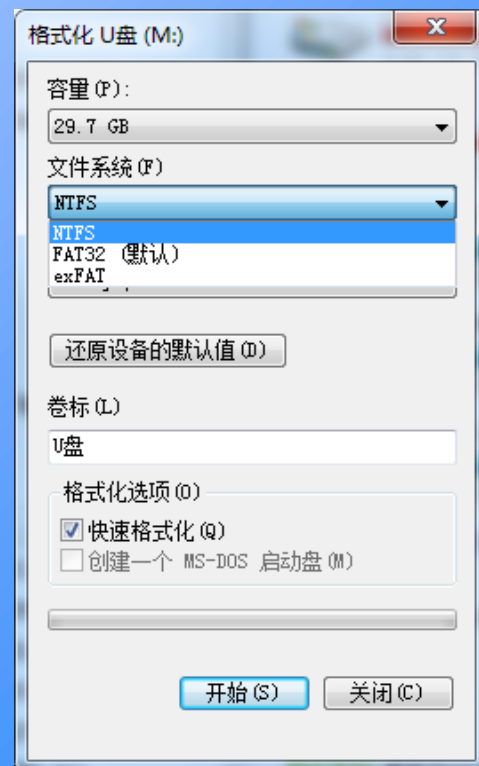
Windows上的文件系统

文件系统，就是对数据进行管理的方式。使用文件系统可有效地管理存储介质。

文件系统在计算机中的表现形式：



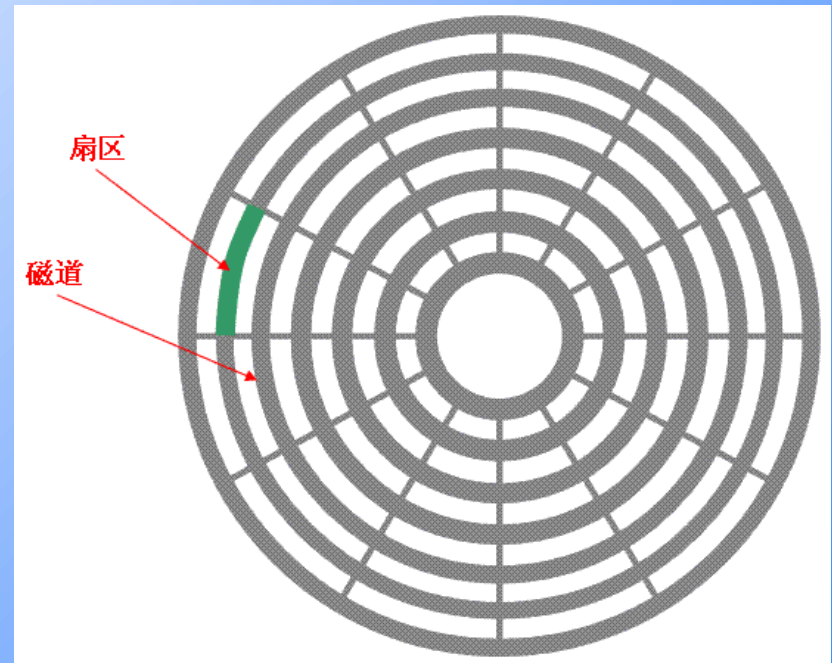
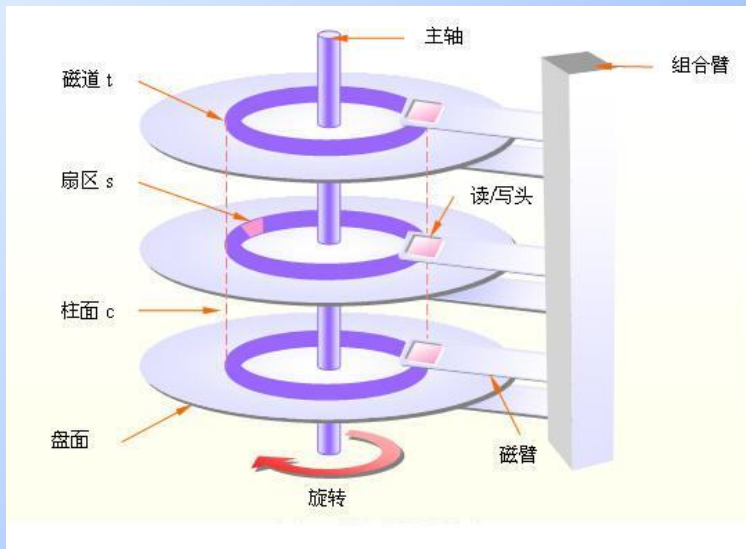
名称	修改日期
.idea	2015/9/15
.settings	2015/9/15
app	2015/9/15
device	2015/9/15
IoT_server	2015/9/15
static	2015/9/15
templates	2015/9/15
.project	2015/9/15
.pydevproject	2015/9/15
db.sqlite_new	2015/9/15
db.sqlite3	2015/9/15
db_fill.py	2015/9/15
db_new.sqlite3	2015/9/15
db_search.py	2015/9/15
db_search.pyc	2015/9/15
django_wsgi.py	2015/9/15
IoT_server.xml	2015/9/15
manage.py	2015/9/15
requirements.txt	2015/9/15
runtest.py	2015/9/15
uwsgi_test.py	2015/9/15



串行FLASH文件系统FatFs



磁盘的物理结构



串行FLASH文件系统FatFs



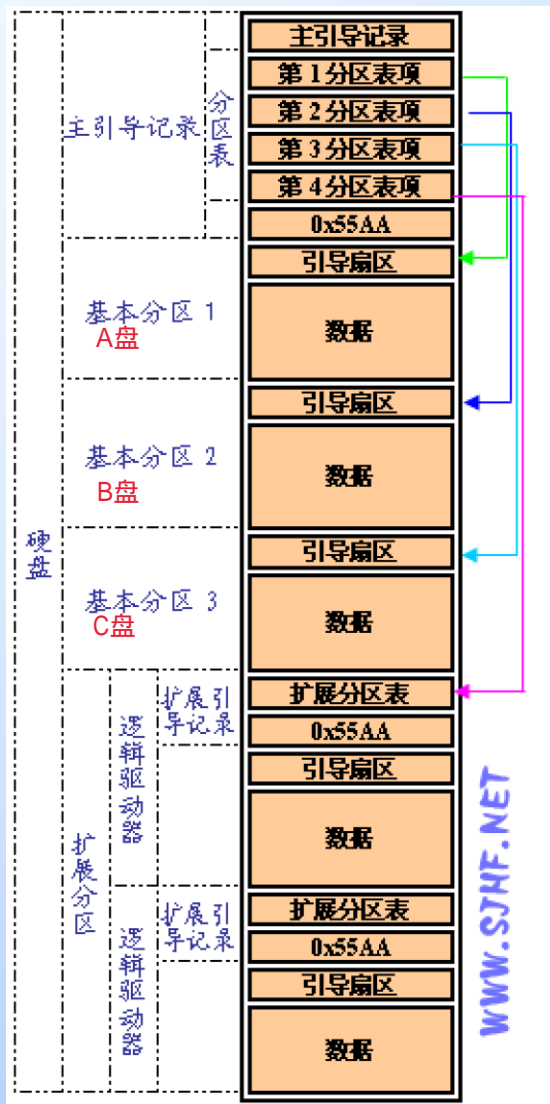
使用文件系统时，它为了存储和管理数据，在存储介质建立了一些组织结构，这些结构包括操作系统引导区、目录和文件。

常见的windows下的文件系统格式包括FAT32、NTFS、exFAT。在使用文件系统前，要先对存储介质进行格式化。格式化时会在存储介质上新建一个文件分配表和目录。这样，文件系统就可以记录数据存放的物理地址，剩余空间。

串行FLASH文件系统FatFs



磁盘分区表



Windows操作系统为了便于用户对磁盘的管理。加入了磁盘分区的概念，即将一块磁盘逻辑划分为几块，它会把磁盘的分区信息记录到硬盘分区表中。

在硬盘分区表中，描述了各个逻辑分区的属性，如分区开始和结束位置所在的物理地址 (柱面号、扇区号)，空间大小等信息。

串行FLASH文件系统FatFs



文件系统的结构与特性

使用文件系统时，数据都以文件的形式存储。写入新文件时，先在目录中创建一个文件索引，它指示了文件存放的物理地址，再把数据存储在地址中。当需要读取数据时，可以从目录中找到该文件的索引，进而在相应的地址中读取数据。具体还涉及到逻辑地址、簇大小、不连续存储等一系列辅助结构或处理过程。

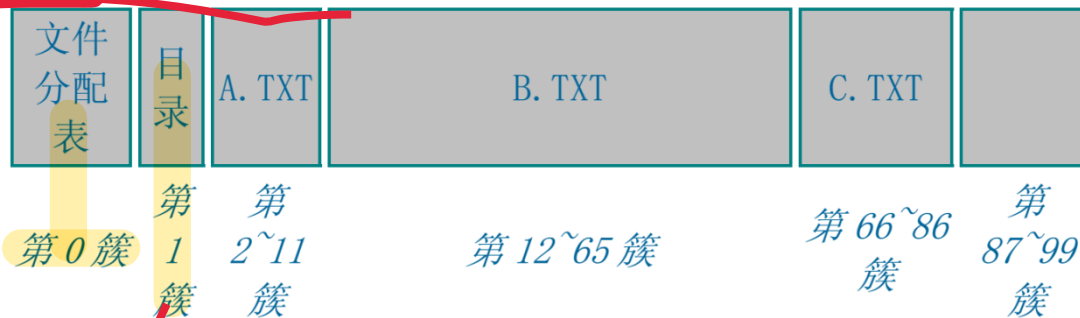
文件系统的存在使存取数据时，不再是简单地向某物理地址直接读写，而是要遵循它的读写格式，如经过逻辑转换，一个完整的文件可能被分开成多段存储到不连续的物理地址，使用目录或链表的方式来获知下一段的位置。

关键词：不再是按物理地址直接写，而是分成多段存储到不连续的物理地址
目录，链表

串行FLASH文件系统FatFs



文件系统的空间示意图



存储了A.TXT，
B.TXT，C.TXT文件

目录示意图

共 10 行 记 录	A.TXT	2	10	2004.3.22 10:41	2004.3.22 10:41	只读	
	B.TXT	12	53.6	1949:10:1 12:0	2003.8.22 20:40	隐藏	
	C.TXT	66	20.5	2000:3:8 21:11	2005:3:8 9:11	系统	
	⋮						

文件名(占50个字节) 开始簇(占4个字节) 文件大小(占10个字节) 创建日期、时间(占10字节) 修改日期、时间(占10字节) 读写属性(占4字节) 保留(12字节)

记录了文件的开始簇
位置、大小等信息

串行FLASH文件系统FatFs



文件系统的空间示意图



存储了A.TXT，
B.TXT， C.TXT文件

文件分配表

存储的是文件的存储信息，即而是分成多段存储到不连续的物理地址

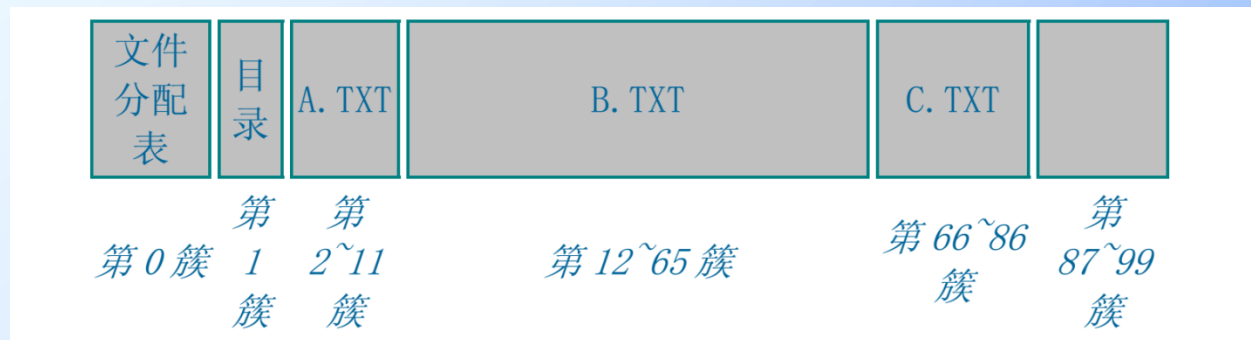
簇号	1	2	3	...	11	12	13	...	65	66	67	...	86	87	...	99
对应数据	目录	3	4	...	FF	13	14	...	FF	67	68	...	FF	00	...	00

文件 a.txt 我们根据目录项中指定的 a.txt 的首簇为 2，然后找到文件分配表的第 2 簇记录，上面登记的是 3，就能确定下一簇是 3。找到文件分配表的第 3 簇记录，上面登记的是 4，就能确定下一簇是 4.....直到指到第 11 簇，发现下一个指向是 FF，就是结束。文件便读取完毕。

串行FLASH文件系统FatFs

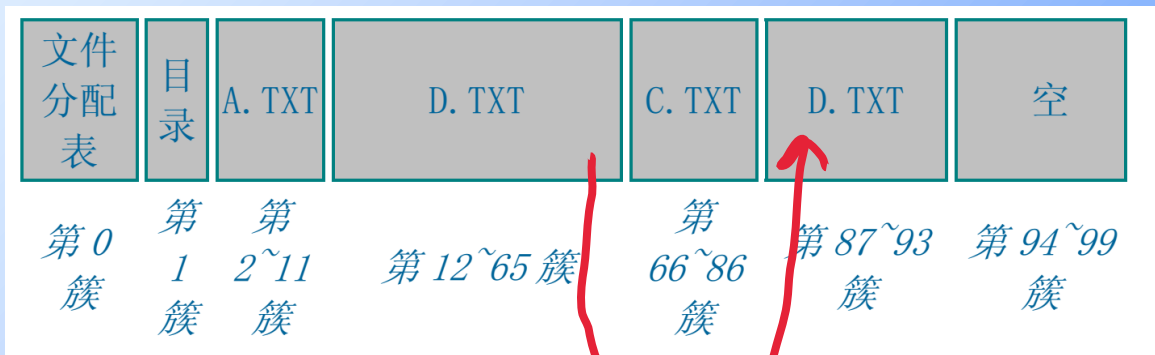


文件系统的空间示意图



删除B.TXT文件，创建D.TXT文件后的空间示意图

文件分成多段存储到不连续的物理地址，通过第0簇的文件分配表，能按照链表的方式读取整个文件



共
10
行
记
录

文件名(占 50个字节)	开始簇(占4个 字节)	文件大小 (占10个字 节)	创建日期、时间 (占10字节)	修改日期、时间 (占10字节)	读写属性 (占4字 节)	保留(12 字节)
-----------------	----------------	----------------------	--------------------	--------------------	--------------------	--------------

共 10 行记录

D. TXT 12

先将B文件的存储簇部分占慢，然后再用链表的形式，继续从C文件之后继续开始存储

串行FLASH文件系统FatFs



原文件分配表示意图

创建新的D后，开始簇在12，但大小有61byte。如何存？

先将B文件的存储簇部分占慢，然后再用链表的形式，继续从C文件之后继续开始存储

簇号	1	2	3	...	11	12	13	...	65	66	67	...	86	87	...	99
对应数据	目录	3	4	...	FF	13	14	...	FF	67	68	...	FF	00	...	00

删除B.TXT文件，创建D.TXT文件后的文件分配表示意图

这就是文件分配表的意义，用于查询文件的存储的不连续地址

簇号	1	2	3	...	11	12	13	...	65	66	67	...	86	87	88	...	93	94	...	99
对应数据	目录	3	4	...	FF	13	14	...	87	67	68	...	FF	88	89	...	FF	00	...	00

先将B文件的存储簇部分占慢，然后再用链表的形式，继续从C文件之后继续开始存储

零死角玩转STM32



THANKS

论坛：www.firebbs.cn

淘宝：firestm32.taobao.com



扫描进入淘宝店铺