

# 一种简易数码相框的设计

丁鑫蕾,刘一清,余奔  
(华东师范大学,上海 200241)

**摘要:** 为应对数码相框成本较高的问题,介绍了一种简易数码相框的设计,实现从 SD 卡读取 bmp 图片文件并在 TFT 液晶屏上显示的功能。本设计成本低,具有很高的实用价值。

**关键词:** 数码相框;SD 卡;STM32; $\mu$ COS-II;FATFS

中图分类号: TP368

文献标识码: A

文章编号: 1674-7720(2011)03-0015-03

## Design of a simple digital photo frame

Ding Xinlei, Liu Yiqing, Yu Ben  
(East China Normal University, Shanghai 200241, China)

**Abstract:** To deal with the high cost of the digital photo frame, this article introduced a design of a simple digital photo frame which can read a bmp picture file from SD card and display it on the TFT LCD screen. The low-cost program has a very high practical value.

**Key words:** digital photo frame; SD card; STM32;  $\mu$ COS-II; FATFS

本文介绍一种简单、低成本的数码相框的设计,它高主频可以达到 72 MHz, 内置 512 KB 的闪存以及 64 KB 的 SRAM。同时其外设资源丰富, 包括并行 LCD 接口

### 1 硬件电路

该数码相框的硬件平台以 32 bit STM32 系列的处理器 STM32F103ZET6 为核心, 外围电路主要包括一个 2.8 英寸的 TFT 液晶模块、1 个 SD 卡插座、5 个 LED 灯和 2 个按键。硬件系统如图 1 所示。

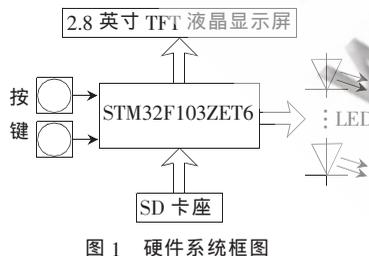


图 1 硬件系统框图

系统关键器件的电路原理图如图 2 所示。STM32 系列 MCU 基于专为要求高性能、低成本、低功耗的嵌入式应用而设计的 ARM Cortex-M3 内核, 包括了 STM32F103 增强型系列和 STM32F101 基本型系列。本设计使用的是增强型系列的高端型号 STM32F103ZET6, 其最

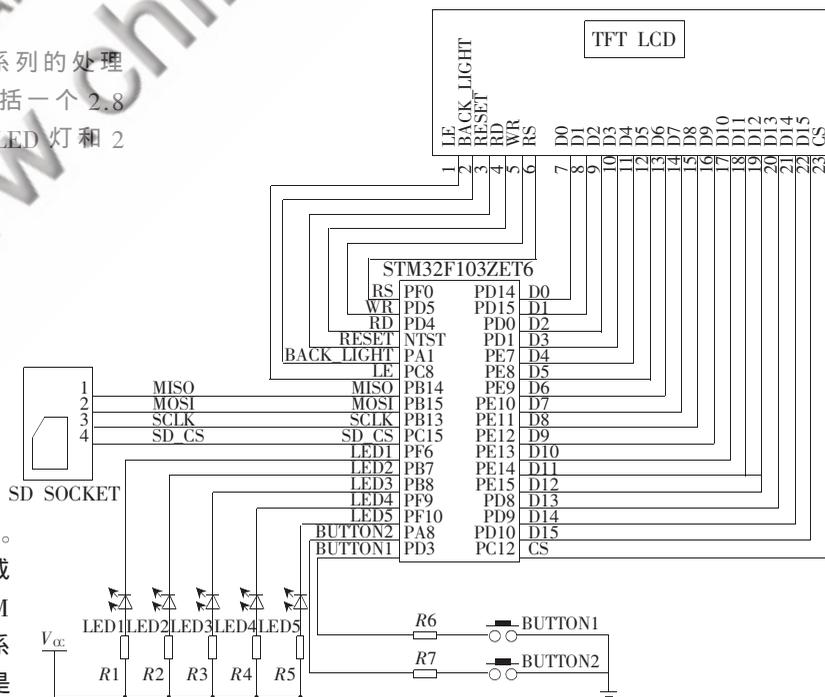


图 2 系统关键器件电路原理图

(FSMC)、12 通道 DMA 控制器、定时器、ADC、DAC、SDIO、I<sup>2</sup>C、SPI、USB、UART 等<sup>[1]</sup>。可以说,它完全超越了同价位的 8 bit 或者 16 bit 单片机的性能,而又不像一般 32 bit 处理器那样价格昂贵。

本设计使用了一个带有通用并行接口(FSMC)的 2.8 英寸 TFT 液晶模块,这种液晶模块很常见,而且价格便宜,另外还有 3.2 英寸甚至更大的尺寸可供选择。STM32F103ZET6 通过一个 SPI 接口与 SD 卡座连接,以便于控制和简化程序,而且对于静态图片,SPI 的传输速度已足够。另外,还有用于调试的 2 个按键和 5 个 LED。

## 2 软件设计

### 2.1 移植 FATFS 文件系统

FATFS 文件系统是一种完全免费开源的 FAT 文件系统模块,专为小型嵌入式系统而设计,用标准 C 语言编写,很方便移植<sup>[2]</sup>。本设计移植的 FATFS 版本为 R0.07e,它支持 FAT12、FAT16 和 FAT32,由于不涉及多个存储介质,因此,本设计仅移植了其简化版本 Tiny-FATFS,其占用内存更少,只需要大约 1 KB RAM。FATFS 具有三层结构,如图 3 所示。



图 3 FATFS 层次结构

应用层为用户提供了一系列接口函数,只需要简单的调用即可,而无须理会复杂的 FAT 协议。移植过程中,需要使用者修改的是底层接口(包括存储介质读写接口和供给文件创建修改时间的实时时钟)。

Tiny-FATFS 包含 5 个文件,分别是 tff.c、tff.h、diskio.c、diskio.h 和 integer.h,移植之前首先要编写初始化和读写 SD 卡的 SPI 接口代码。代码主要包括 SD\_Init()、SD\_ReadSingleBlock()、SD\_ReadMultiBlock()、SD\_WriteSingleBlock()、SD\_WriteMultiBlock() 函数,而这些代码将在 diskio.c 中被调用。移植 FATFS 文件过程:

(1) 准备好 SPI 读写 SD 卡的代码后就可以修改 diskio.c 了,实际上只需要编写 6 个接口函数:

① 存储媒介初始化函数: DSTATUS disk\_initialize (BYTE drv)。由于存储媒介是 SD 卡,所以实际上是对 SD 卡的初始化。drv 是存储媒介号码,由于 Tiny-FATFS 只支持一个存储媒介,所以 drv 应恒为 0。执行无误返回 0,错误返回非 0。

② 状态检测函数: DSTATUS disk\_status (BYTE drv)。检测是否支持当前的存储媒介,对 Tiny-FATFS 来说,只要 drv 为 0,就认为支持,然后返回 0。

③ 读扇区函数: DRESULT disk\_read (BYTE drv, BYTE\*buff, DWORD sector, BYTE count)。在 SD 卡读接口函数的基础上编写,\*buff 存储已经读取的数据,sector 是

开始读的起始扇区,count 是需要读的扇区数。1 个扇区为 512 B。执行无误返回 0,错误返回非 0。

④ 写扇区函数: DRESULT disk\_write (BYTE drv, const BYTE\*buff, DWORD sector, BYTE count)。在 SD 卡写接口函数的基础上编写,\*buff 存储要写入的数据,sector 是开始写的起始扇区,count 是需要写的扇区数。1 个扇区为 512 B。执行无误返回 0,错误返回非 0。

⑤ 存储媒介控制函数: DRESULT disk\_ioctl (BYTE drv, BYTE ctrl, Void\*buff)。ctrl 是控制代码,\*buff 存储或接收控制数据。可以在此函数里编写自己需要的功能代码,如获得存储媒介的大小、检测存储媒介的上电与否,存储媒介的扇区数等。如果是简单的应用,也可以不用编写,返回 0 即可。

⑥ DWORD get\_fatime (Void)。本设计未涉及到文件的创建和修改,因此实时时钟部分未编写,直接返回一个 32 bit 无符号整数即可。

(2) 对 Tiny-FATFS 进行配置,主要是修改 tff.h 和 diskio.h。在 diskio.h 中,只需要配置一项:使能或者使能只读属性。由于本设计只涉及到 SD 卡的读操作,因此定义使能只读属性:#define READONLY1。这样可以裁减掉关于写 SD 卡的那部分代码,从而使文件系统更精简。

在 tff.h 中,需要对整个文件系统做全面的配置,包括:

- ①#define MCU\_ENDIAN 1。STM32 是小端模式,选择 1。
  - ②#define FS\_READONLY 1。使能只读操作,去掉写操作,节省空间。
  - ③#define FS\_MINIMIZE 3。设为 3 表示使用最基本的 FATFS 函数,最大限度节省空间。
  - ④#define USE\_STRFUNC 0。禁用字符串函数功能。
  - ⑤#define USE\_FORWARD 0。禁用 forward 功能。
  - ⑥#define FAT321。支持 FAT32 文件系统。
  - ⑦#define USE\_FSINFO 1。支持 FAT32 磁盘信息获取。
  - ⑧#define USE\_SJIS 0。禁用 shift-JIS 码。
  - ⑨#define USE\_NTFLAG 1。对文件名大小写敏感。
- 到此为止,FATFS 的移植就全部完成了。

编写一段程序测试 FATFS 是否移植成功。以 1 s 的间隔循环读取 SD 卡中的 bmp 格式图片并显示,main 函数的流程如图 4 所示。



图 4 FATFS 测试程序流程图

需要说明的是,为了便于循环读图,SD 卡中存储的 bmp 格式图片被刻意用数字编号来命名(如:1.bmp~n.bmp),这样,在程序中只需要简单处理文件名就可以按顺序读取图片了。

通常 bmp 图片都是 24 位颜色深度的(即 RGB=8:8:8),而本设计使用的 TFT 液晶是 16 位颜色深度的(即 RGB=5:6:5),因此在程序中需要作转换处理才能正常显示。转换程序代码如下:

```
u16 RGB888toRGB565(u8 r,u8 g,u8 b)
{
    return(u16)(r & 0xF8) << 8|(g & 0xFC) << 3|(b &
        0xF8) >> 3;
}
```

由于 bmp 图片是最简单的图片格式,不涉及解码处理,因此大大简化了程序设计。

## 2.2 在 $\mu$ COS-II 平台上加载 FATFS 文件系统

FATFS 的加载比其他一些嵌入式文件系统要简单<sup>[4]</sup>,只需要在工程中加入 tff.c 和 diskio.c 这两个文件即可。需要注意的是,在 tff.h 中有以下 4 个函数:

```
int fputc(int, FILE*);
int fputs(const char*, FILE*);
int fprintf(FILE*, const char*, ...);
char* fgets(char*, int, FILE*);
```

这 4 个函数与 stdio.h 中定义的函数有冲突,需要将其删除。

此外,tff.h 中还定义了一个 EOF,也与 stdio.h 中的定义有冲突,只需把它重命名即可(如 EOFF)。

## 2.3 在 $\mu$ COS-II 平台上编写任务函数

定义三个任务以实现数码相框功能:

```
static void TaskStart(void *pdata);
static void Task1(void *pdata);
static void Task2(void *pdata);
```

其中,TaskStart 的优先级最高,在 main 函数中,初始化  $\mu$ COS-II 后即被创建:

```
/* Initialize "μC/OS-II, The Real-Time Kernel" */
OSInit();
/* Create the start task */
OSTaskCreate(TaskStart, (void*)0,
    &TaskStartStk[TASK_START_STK_SIZE-1], 0);
```

负责初始化目标板、SD 卡和 LCD,并创建任务 Task1,同时处理按键 USER\_BUTTON\_1。当按下 USER\_BUTTON\_1 时,删除任务 Task2,其流程如图 5 所示。

Task1 的任务是循环读取 SD 卡的 bmp 格式图片并送 LCD 显示,同时还要处理按键 USER\_BUTTON\_2。当按下 USER\_BUTTON\_2 时,创建任务 Task2,其示意性代码如下:

```
static void Task1(void *pdata)
{
```

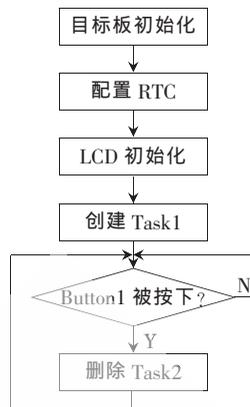


图 5 TaskStart 流程图

```
/* 设置变量; */
While(1)
{
    if(! USER_BUTTON_2)
        OSTaskCreate(Task2, (void *)0,
            &TaskStk2[TASK2_STK_SIZE-1], 4);
    Read_pic_display();
    OSTimeDly(OS_TICKS_PER_SEC/2);
}
Task2 任务则只负责执行流水灯程序,即循环依次
```

点亮 5 个 LED,其优先级最低。

## 3 测试结果

本设计的实验平台是 STM32 红牛开发板,实物效果图如图 6 所示。



图 6 实物效果图

实验结果表明, $\mu$ COS-II+FATFS 方案实现的数码相框工作稳定,开机后  $\mu$ COS-II 加载成功即开始循环读取 SD 卡中的 bmp 图片并在 TFT 屏上显示。图片间的切换是简单的拉幕式切换,间隔为 5 s(可调),经长时间观察,无读取失败、掉帧、花屏等问题,每张图片都是完美展示。根据测试程序的设计,按下 USER\_BUTTON\_2 按键则流水灯任务被创建,5 个 LED 灯开始以走马灯的形式循环点亮;与此同时,图片读取和显示未受到任何影响,

证明  $\mu\text{COS-II}$  的任务调度机制稳定可靠。再按下 USER\_BUTTON\_1, 则流水灯任务被删除, LED 停止动作。

本设计基本实现了数码相框的功能, 其设计的硬件极其简单, 相比于一些基于高端 32 bit MCU (如 ARM) 的方案<sup>[5]</sup>, 成本要低得多。相比于一些简单的读图显示程序, 本设计基于嵌入式实时操作系统  $\mu\text{COS-II}$ , 在稳定性和可扩展性上都具有优势。另外, STM32 性能优越, 可以很流畅地运行  $\mu\text{COS-II}$  和 FATFS 系统, 也能够驱动 QVGA 甚至更高分辨率的 TFT 液晶屏, 这是同价位 8 bit 或 16 bit 单片机所不能比拟的<sup>[6]</sup>。本设计还有需要改进的地方, 如增加 JPEG 解码、增强文件名支持、改进幻灯片切换方式以及添加其他功能等。

#### 参考文献

[1] 王永虹, 徐炜, 郝立平. STM32 系列 ARM Cortex-M3 微控制器原理与实践[M]. 北京: 北京航空航天大学出版社, 2008.

- [2] 洪岳炜, 王百鸣, 谢超英. 一种易于移植和使用的文件系统 FATFS Module[J]. 单片机与嵌入式系统应用, 2008 (1): 29-31.
- [3] 任哲. 嵌入式实时操作系统  $\mu\text{C/OS-II}$  原理与应用[M]. 北京: 北京航空航天大学出版社, 2005.
- [4] 王命延, 余爱军. 一种加载在  $\mu\text{COS-II}$  内核上的嵌入式文件系统[J]. 南昌大学学报, 2005, 29(2): 197-199.
- [5] 王伟能, 吴烁宇, 陈弟虎, 等. 基于 S3C2410 和 Qt/Embedded 的数码相框设计[J]. 微计算机信息, 2009, 25 (6-2): 81-83.
- [6] 郑友胜.  $\mu\text{COS-II}$  在 C51 中的应用设计[J]. 中小企业科技, 2007(1): 126-127.

(收稿日期: 2010-09-07)

#### 作者简介:

丁鑫蕾, 女, 1986 年生, 硕士研究生, 主要研究方向: 高速电路设计与视频图像处理。