

青风手把手教你学 stm32f030 系列教程

----- 库函数操作版本

出品论坛: www.qfv8.com 青风电子社区



作者：青风

出品论坛: www.qfv8.com

淘宝店: <http://qfv5.taobao.com>

QQ 技术群: 241364123

硬件平台：QF-STM32F030 开发板

2.19 TFT LCD 显示图片

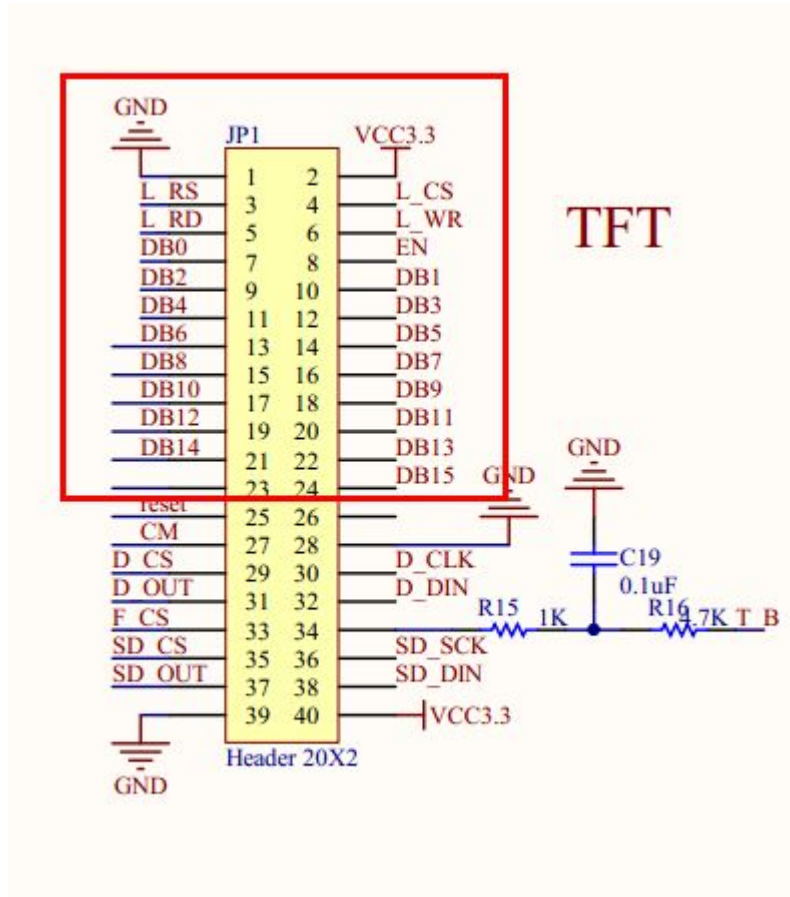
2.19.1 原理分析

图片显示的原理和前面 18 节讲的汉子显示一样的，也是需要把图片按照文件格式进行扫描成二进制数据，常用的图片格式有很多，一般最常用的有三种：JPEG（或 JPG）、BMP 和 GIF。我们这里为了简单只接受 BMP 图片的显示，其他类型的图片显示会在后面更新。我们可以通过 WinHex 软件打开图片，可读取到该文件的原始数据。如下图所示：

1234.bmp		Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1234.bmp		00000000	42	4D	B6	01	00	00	00	00	00	00	36	00	00	00	28	00	BM.....6....(
D:\Administrator\桌面		00000016	00	00	0F	00	00	00	08	00	00	00	01	00	18	00	00	00
文件大小:	438 B	00000032	00	00	00	00	00	00	C4	0E	00	00	C4	0E	00	00	00	00Ä...Ä...
	438 字节	00000048	00	00	00	00	00	00	31	31	31	31	31	31	31	31	31	311111111111
节省编辑模式		00000064	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
状态:	原始的	00000080	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
取消磁头:	0	00000096	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	111...1111111111
反向取消:	no	00000112	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
		00000128	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
创建时间:	2011-09-10 20:52:08	00000144	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	111...1111111111
		00000160	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
最后写入时间:	2011-09-10 20:52:08	00000176	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
		00000192	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	111...1111111111
属性:	A	00000208	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
图标:	1	00000224	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
		00000240	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	111...1111111111
模式:	十六进制	00000256	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
字符集:	ANSI ASCII	00000272	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
偏移地址:	decimal	00000288	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
每页字节数:	37x16=592	00000304	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	111...1111111111
当前窗口:	1	00000320	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
窗口总数:	1	00000336	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	1111111111111111
剪贴板:	可用	00000352	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	31	111...1111111111
		00000368	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	1111111111111111
暂存文件夹:		00000384	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	0000111111111111
	S:\TEMP	00000396	31	31	31	00	00	00	31	31	31	31	31	31	31	31	31	31	111...1111111111

但是这样只能每次都进行图片转换比较麻烦，因此在后面我们引入文件系统批量的读取文件信息，能够方便的实现数码相框的功能。

2.19.2 硬件准备:



端口配置:

硬件连接: PB0-PB7---DB0-DB7 低 8 位

PC0-PC7---DB8-DB7 高 8 位

PD2---WR PC12---RD PC11---CS PC10---RS PB11---CM

其中 WR:写信号输入引脚, 低电平时有效。

RD:读信号输入引脚, 低电平时有效。

CS: TFT 液晶屏的片选信号, 低电平时有效。

RS: 写数据和写命令选择引脚。取高写数据, 取低写命令。

CM: 选择 8BIT 和 16BIT 选择引脚。

2.19.3 软件准备:

由于 f030 的内存非常小, 我们采用 SD 卡存储一个图片文件进行显示, 这个工程的树建立如下:



我们需要编写一个读取图片的函数，为了简便，只写了读取 BMP 文件的读取函数。BMP 文件格式，又称为 Bitmap（位图）或是 DIB(Device-Independent Device，设备无关位图)，是 Windows 系统中广泛使用的图像文件格式。

BMP 文件可以保存单色位图、16 色或 256 色索引模式像素图、24 位真彩色图象，每种模式种单一像素的大小分别为 1/8 字节，1/2 字节，1 字节和 3 字节。

我们设计显示 BMP。这种文件格式还定义了像素在开始处有一个文件头，大小为 54 字节。保存了包括文件格式标识、颜色数、图象大小、压缩方式等信息，图象的宽度和高度都是一个 32 位整数，在文件中的地址分别为 0x0012 和 0x0016。54 个字节以后，如果是 16 色和 256 位色 BMP，则还有一个颜色表，但 24 位色 BMP 没有这个。因此我们大体需要编写两个函数，首先是读取 BMP 头文件信息，代码如下，这个代码是直接从网上移植过来的：

```
01. BMP_HEADER TFTBmpGetHeadInfo(uint8_t *buf)
02. {
03.     BMP_HEADER bmpHead;
04.
05.     bmpHead.bfType = (buf[0] << 8) + buf[1];           // BM
06.     bmpHead.bfSize = (buf[5] << 24) + (buf[4] << 16) + (buf[3] << 8) + buf[2]; // 文件大小
07.     bmpHead.biWidth = (buf[21] << 24) + (buf[20] << 16) + (buf[19] << 8) + buf[18]; // 图像宽度
08.     bmpHead.biHeight = (buf[25] << 24) + (buf[24] << 16) + (buf[23] << 8) + buf[22]; // 图像高度
09.     bmpHead.biBitCount = (buf[29] << 8) + buf[28]; // 每个像素的位数，单色位图为 1，256
        色为 8，16bit 为 16，24bit 为 24
10.     return bmpHead;
11. }
```

然后对图像文件进行扫描:

```

12. uint8_t TFTBmpDisplay(uint8_t *bmpName,uint16_t x,uint16_t y)
13. {
14.     FATFS fs;           // 建立文件系统
15.     FIL file;           // 建立文件
16.     UINT  br;           // 字节计数器
17.     FRESULT res;        // 返回值信息
18.     BMP_HEADER bmpHead; // 头信息
19.     uint16_t i;
20.
21.     f_mount(0, &fs); // 挂载文件系统
22.     res = f_open(&file, (const TCHAR *)bmpName, FA_OPEN_EXISTING|FA_READ); // 打开
    BMP 文件并读取到 file 中
23.     if(res != FR_OK)
24.     {
25.         return res;
26.     }
27.     else
28.     {
29.         res = f_read(&file, Buffer, 54, &br); // 读取头文件信息
30.         if(res != FR_OK)
31.         {
32.             return res; // 返回错误表示
33.         }
34.         else
35.         {
36.             bmpHead = TFTBmpGetHeadInfo(Buffer); // 获取头信息
37.
38.             if(bmpHead.bfType == 0x424D) // 判断是否为 BMP 图像
39.             {
40.                 LCD_WR_REG_DATA(0x0003, 0x1010); // 由下而上显示
41.                 LCD_XYRAM(x, y, x+bmpHead.biWidth-1, y+bmpHead.biHeight-1);
42.                 LCD_WR_REG_DATA(0x0020,x); // 设置 X 坐标位置
43.                 LCD_WR_REG_DATA(0x0021,y+bmpHead.biHeight-1); // 设置 Y 坐标位置(注
    意: 在由下而上显示的时候, 这里 y 坐标应该是最下边的值)
44.                 LCD_WR_REG(0x0022); // 指向 RAM 寄存器, 准备写数据到 RAM
45.                 while(1)
46.                 {
47.                     res = f_read(&file, Buffer, 240, &br); // 读取 240 个数据
48.                     if(res||br==0)
49.                         // 错误跳出
50.                         break;
51.                     for(i=0;i<80;i++)
52.                     {

```

```

52.                // 在 TFT 上显示一个像素点的颜色
53.                LCD_WR_DATA(((Buffer[i*3+2]/8)<<11 | (Buffer[i*3+1]/4)<<5
    |(Buffer[i*3]/8)));
54.                }
55.            }
56.            LCD_WR_REG_DATA(0x0003, 0x1030); // 恢复正常显示
57.            LCD_XYRAM(0, 0, 239, 319);        // 恢复 GRAM
58.        }
59.    }
60. }
61.
62.     f_close(&file);        // 关闭文件, 必须和 f_open 函数成对出现
63.     f_mount(0, 0);         // 卸载文件系统
64.     return FR_OK;          // 返回成功标志
65. }
66.

```

以上代码来自网络, 移植到 STM32F0 上可行, 特此声明。

因此对图像扫描显示后就可以在主函数中调用显示了:

```

67.     num = FileScan("picture"); // 扫描 picture 文件
68.     if(num>50)num = 50;        // 最多 50 个文件
69.     strcpy((char *)tempPath, "picture/"); // 把文件名路径给了 tempPath 暂存
70.
71.
72.     while(1)
73.     {
74.         for(i=0; i<num; i++)    // 循环扫描文件得到哪个是 BMP 文件
75.         {
76.             if(flag[i] == 1)    // 检测如果是 BMP 图片
77.             {
78.                 strcpy((char *)filePath, (char *)tempPath);
79.                 strcat((char *)filePath, (char *)FileN[i]);
80.                 TFTBmpDisplay((uint8_t*)filePath, 0, 0); // 显示出来图片
81.                 delay(10);
82.                 delay(10);
83.                 delay(10);
84.                 delay(10);
85.                 delay(10);
86.             }
87.         }
88.     }
89.
90.

```

实验下载现象:

显示照片如下:



