

FATFs SD 卡底层程序之图片显示

硬件环境：DAXIA32 学习板 STM32F103VC 72MHz

编程环境：MDK 3.80a

主程序：

```
//*****  
*****  
/* 函数名称：MAIN  
/* 功          能：从 SD 卡读 BMP 图片并显示  
/* 编          写：Bunagi  
/* 修          改：2010/9/4  
//*****  
*****  
  
int main(void)  
{  
    FATFS fs;  
    DIR dir;  
    FIL fil;  
    FILINFO fno;  
    FRESULT res;  
    char Buf[512];  
    u8 i;  
    char FilSearchCount[10][13]; // 文件名检索存放  
  
    ChipHalInit();  
    ChipOutHalInit();  
  
    Lcd_Init();  
    TFT_X = 0;  
    TFT_Y = 0;  
    BG_COLOR = WHITE;  
    TFTFill(WHITE);  
  
    SysTickDelay(1000);  
    TFTShowString16(TFT_X, TFT_Y, "图片显示测试", RED);  
  
    while(!GET_DOWN());  
  
    TFT_X = 0;  
    TFT_Y = 0;
```

```

TFTFill(BLACK);
//*****
f_mount(0,&fs);

if(!FilSearch(&fs,&dir,"","BMP",FilSearchCount))
{
    TFTShowString16(TFT_X,TFT_Y,"检索目录失败!",RED);
    while(1);
}
else
{
    for(i = 0;i < 10;i++)
    {
        if(FilSearchCount[i][0] != 0)
        {
            TFTShowString16(TFT_X,TFT_Y,&FilSearchCount[i][0],RED);
            TFT_Y += 16;
        }
    }
}

for(;;)
{
    for(i =0;i < 10;i++)
    {
        if(FilSearchCount[i][0] != 0)
            if(!BMPDisp(&fil,&dir,&fno,&FilSearchCount[i][0],&Buf,239,31
9))
                TFTShowString16(TFT_X,TFT_Y,"打开图片失败!",RED);
        while(!GET_DOWN());
    }

    TFT_X = 0;
    TFT_Y = 0;
    TFTShowString16(TFT_X,TFT_Y,"图片已播放结束!",RED);
    while(!GET_UP());
}
while(1);
}

```

```

//*****
*****
/** 函数名称: NameCompare
/** 功    能: 字符串比较 比较 0x2E 前的字符(文件名比较)
/** 输入参数: str1, str2
/** 输出参数: TRUE 两字符串相同          FALSE 两字符串不同
/** 编      写: Bunagi
/** 修      改: 2010/9/4
//*****
*****
bool NameCompare(u8 *str1, u8 *str2)
{
    u8 i=13;
    u8 flag;
    while(--i)
    {
        if(str1[i] == 0x2e) flag = i;
    }
    for(i=0; i < flag+3; i++)
    {
        if(str1[i] != str2[i]) return FALSE;
    }
    return TRUE;
}

//*****
*****
/** 函数名称: BMPInfoLoad
/** 功    能: 读 SD 卡里 BMP 文件的头 512 字节数据, 并填充进 BMP 信息数据
结构里
/** 输入参数: bfinfo
/**          binfo
/**          Buf
/** 输出参数: 无
/** 编      写: Bunagi
/** 修      改: 2010/9/4
//*****
*****
void BMPInfoLoad(BMPFILINFO *bfinfo, BMPINFO *binfo, u8 *Buf)
{
    // 填充 BMPFIL 信息
    bfinfo->wBFIType = (Buf[0]<<8) + Buf[1];
    bfinfo->dwBFISize = (Buf[5]<<24) + (Buf[4]<<16) + (Buf[3]<<8) + Buf[2];
}

```

```

bfinfo->dwBFIReserved = 0x00;
bfinfo->dwBFIStartAdd = (Buf[13]<<24) + (Buf[12]<<16) + (Buf[11]<<8) +
Buf[10];
// 填充 BMP 信息
binfo->dwBISize      = (Buf[17]<<24) + (Buf[16]<<16) + (Buf[15]<<8) +
Buf[14];
binfo->dwBIWidth      = (Buf[21]<<24) + (Buf[20]<<16) + (Buf[19]<<8) +
Buf[18];
binfo->dwBIHeight = (Buf[25]<<24) + (Buf[24]<<16) + (Buf[23]<<8) +
Buf[22];
binfo->wBIPlanes      = 0x01;
binfo->wBIBitCount = (Buf[29]<<8) + Buf[28];
binfo->dwBICompression = (Buf[33]<<24) + (Buf[32]<<16) + (Buf[31]<<8) +
Buf[30];
binfo->dwBISizeImage = (Buf[37]<<24) + (Buf[36]<<16) + (Buf[35]<<8) +
Buf[34];
binfo->dwBIXpelsPerMeter = (Buf[41]<<24) + (Buf[40]<<16) + (Buf[39]<<8)
+ Buf[38];
binfo->dwBIYpelsPerMeter = (Buf[45]<<24) + (Buf[44]<<16) + (Buf[43]<<8)
+ Buf[42];
binfo->dwBIClrUsed = (Buf[49]<<24) + (Buf[48]<<16) + (Buf[47]<<8) +
Buf[46];
binfo->dwBIClrImportant = (Buf[53]<<24) + (Buf[52]<<16) + (Buf[51]<<8)
+ Buf[50];
}
//*****
*****
/** 函数名称: BMPDisp
/** 功      能: 在指定坐标点位置显示图片 (图片大小依据 LCD 屏开始及结
束坐标确定
/** 输入参数:
/** 输出参数:
/** 编      写: Bunagi
/** 修      改: 2010/9/4
//*****
*****

bool BMPDisp(FIL *fil, DIR *dir, FILINFO *fno, u8 *FilSearch, u8 *Buf, LOC_X
tft_end_x, LOC_Y tft_end_y)
{
    u8 color_byte, rgb;
    int tft_x, tft_y;
    ul6 count, color, tft_start_x, tft_start_y;
    UINT len;

```

```

BMPFILINFO bfinfo;
BMPINFO binfo;
FRESULT res;

    tft_x = tft_end_x;
    tft_y = tft_end_y;
rgb = 0;

res = f_opendir(dir, "");
// 查找指定文件名的文件
while(1)
{
    res = f_readdir(dir, fno);
    if(res != FR_OK) return FALSE;

    if(NameCompare(fno->fname, FilSearch)) break;
    else if(fno->fname[0] == 0) return FALSE;
}

res = f_open(fil, fno->fname, FA_OPEN_EXISTING | FA_READ);
    if(res != FR_OK) return FALSE;
res = f_read(fil, Buf, BMPBufSize, &len);
    if(res != FR_OK) return FALSE;


    BMPInfoLoad(&bfinfo, &binfo, Buf);

tft_start_x = tft_end_x - (u16)(binfo.dwBIWidth-1);
tft_start_y = tft_end_y - (u16)(binfo.dwBIHeight-1);

rgb = 0;
    count = bfinfo.dwBFISStartAdd;                                // 当前像素数
据地址指针
    color_byte = binfo.wBIBitCount >> 3; // binfo.wBIBitCount / 8 判
断是几位的位图

    do
    {
        while(count < BMPBufSize)
        {
            //处理 24 位位图
            if(color_byte == 3)
            {
                switch(rgb)

```

```

{
    case
0:          // 蓝色像素数据转换成 R5G6B5 格式
            {
                color =
Buf[count] >> 3;

                rgb++;
                break;
            }

    case
1:          // 绿色像素数据转换成 R5G6B5 格式
            {
                color |=
(Buf[count] >> 2) << 5;

                rgb++;
                break;
            }

    case
2:          // 红色像素数据转换成 R5G6B5 格式
            {
                color |=
(Buf[count] >> 3) << 11;

                rgb++;
                break;
            }

    default: break;
}
if(rgb == color_byte)          // 象
素转换完成
{
    TFTPoint(tft_x, tft_y, color);
    color = 0;
    rgb = 0;
    if(--tft_x < tft_start_x)
    {
        tft_x = tft_end_x;
        if(--tft_y <
tft_start_y) return TRUE;

        // 处理每行像素对齐
        if(((binfo.dwBIWidth*3)%4) != 0) count +=4 -
((binfo.dwBIWidth*3)%4);
    }
}
count ++;

```

```

        }
        else
        {
            // 添加处理其它比如 16 位图的程序
        }
    }
    // 读下一个扇区数据
    if(f_read(fil, Buf, BMPBufSize, &len)) return FALSE;
    count = 0;
}while(!res && len);
return TRUE;
}

//*****
//*****
/** 函数名称: FilSearch
/** 功      能: 在指定路径下查找指定扩展名的文件，并记录在(*p)[13]数组中，注意最大记录条
/**      勿溢出
/** 输入参数: fs
/**      dir
/**      path 要检索的文件路径
/**      name 要检索的文件扩展名
/**      (*p)[13] 检索结果存放地址
/** 输出参数: TRUE 成功      FALSE 出错
/** 编      写: Bunagi
/** 修      改: 2010/9/4
//*****
//*****
bool FilSearch(FATFS *fs, DIR *dir, XCHAR *path, u8 *name, char (*p)[13])
{
    FILINFO fno;
    FRESULT res;
    u8 i, j, k;

    j = k = 0;

    res = f_opendir(dir, path);
    if(res != FR_OK) return FALSE;

    do
    {

```

```

    res = f_readdir(dir,&fno);
    if(res != FR_OK) return FALSE;

    if(fno.fname[0] == 0) return TRUE;
    i = 13;
    while(--i)
    {
        if((fno.fname[i] == 0x2E) && (fno.fname[i+1] == name[0] &&
fno.fname[i+2] == name[1] && fno.fname[i+3] == name[2])) break;
    }
    if(i)          //检索到一个符合扩展名条件的文件
    {
        for(k =0;k < 13;k++) *((p+j)+k) = fno.fname[k];
        j++;
    }
}while(!res && fno.fname[0] != 0);
}

```