

1.77 寸 TFT 使用说明

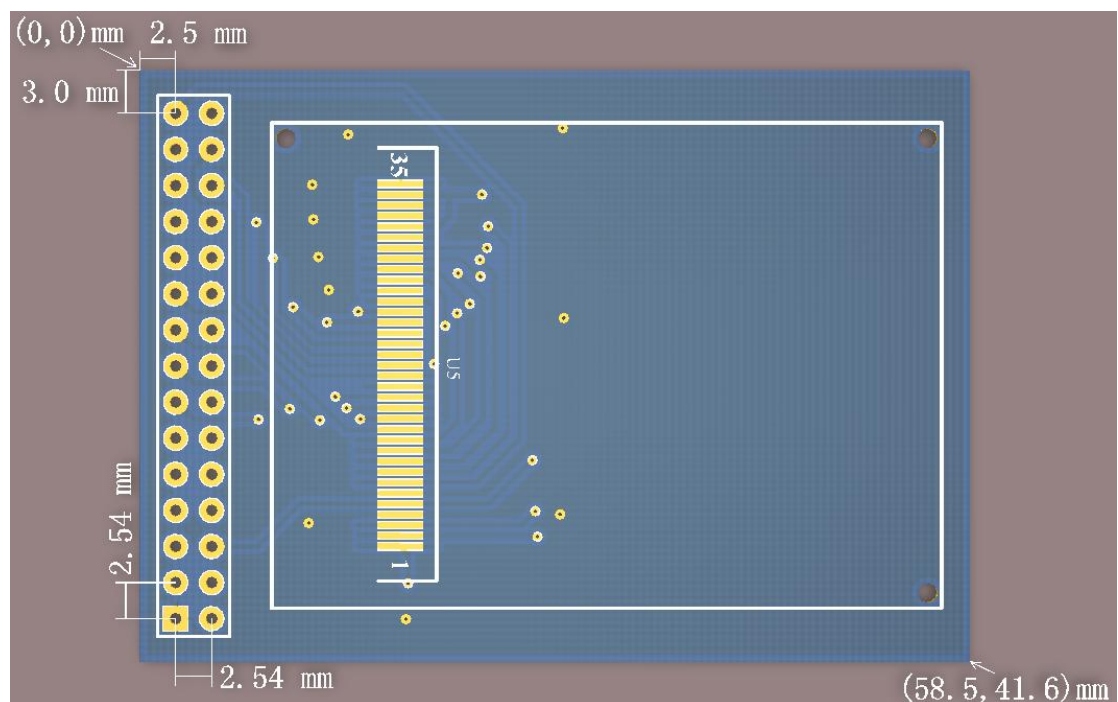
外观美观大方，质量可靠，采用品牌厂商生产的 TFT，提供多个例程，程序精炼、准确、实用。您的满意就是我们的成功！

TFT 尺寸：1.7”；

分辨率：128xBGRx160，可达 262K 色；

电压类型：支持 5V 或 3.3V 电压输入，5v 电压接入时，“VCC_SEL”（即 16、17pin）断开；3.3v 电压接入时，“VCC_SEL”短接；

接口类型：支持 8 位数据接口或 16 位数据接口，8 位数据接口可与 12864 液晶接口兼容（修改一下程序中的接口定义即可）。使用 16 位数据接口时，“MOD”（即 18pin）连接到 GND。



TFT 液晶各引脚功能说明如下：

GND: 接地；

VCC: 5V/3V 电压输入，**注意 VCC_SEL 的连接**；

CS: 片选信号；

RS: 命令/数据选择；RS=0 写数据，RS=1 写命令；

WR: 写控制信号；

RD: 读控制信号;

D8 ~ D15: 使用 8 位数据口时为 8 位数据/命令并行口; 使用 16 位数据/命令并行口时为其高 8 位;

RST: 液晶复位信号;

VCC_SEL: 5v VCC 接入时 “VCC_SEL” 断开; 3.3v VCC 接入时 “VCC_SEL” 短接;

MOD: 数据/命令并行口的模式选择, MOD=GND 时 TFT 为 16 位工作模式, MOD 不连接时 TFT 为 8 位工作模式;

D0 ~ D7: 仅在使用 16 位数据/命令并行口时使用, 为 16 位数据/命令并行口的低 8 位;

液晶控制 IC 说明请参考“S6D0144. pdf”, 液晶 FPC 接口定义请参考“GG1P0360 管脚定义. pdf”。

8 位接口模式, 程序中的接口定义:

D8 ~ D15 = P0;

CS = P2^1;

RST = P2^2;

RS = P2^5;

WR_ = P2^4;

RD_ = P2^3;

16 位接口模式, 程序中的接口定义:

D0 ~ D7 = P0;

D8 ~ D15 = P1;

CS = P2^1;

RST = P2^2;

RS = P2^5;

WR_ = P2^4;

RD_ = P2^3;

SD 卡使用方法

程序 4 的功能是将 SD 卡中存储的图片显示到 TFT，故此处 SD 卡主要用来存储图片。其使用方法说明如下：

1. 77 寸彩屏分辨率为 128x160，程序控制中用 2 个字节（16 位）表示一个点的颜色。

$128 \times 160 \times 2 = 40960$ 字节，即为 40k，所以如果完整写一幅图片，需要 40k 存储空间，51 系列单片机一般最大内部 ROM 为 64k，显然不能放多张图片，常常需要 SD 卡存储图片。SD 卡使用如下，SD 卡要格式化 FAT（即 FAT16 格式），然后把需要显示的大小为 128x160 像素的 bmp 或 JPG 格式图片通过 Image2LCD 软件转换成 bin 格式并存储到 SD 卡中。

注意：SD 卡中不能存储其他任何文件，使用之前一般需要先格式化，否则可能导致图片输出不完整。

图片存储到 SD 卡后，可以通过 winhex 软件查看各图片存储的地址。打开软件如图 2 所示，点击第一张图片“7. bin”（通过地址查看可知，存储器里面实际上是这张存放在前面），可以看到右端第 1 扇区地址是 520，这是数据区最小的地址。所有图片在 SD 卡中是依次存放的，读图片也是依次进行的。看上图中左下角圈起来的 2 个数字，上面的物理扇区编号，下面的是逻辑扇区编号，配套的程序中由于没有使用完整的 FAT 格式所以这里我们选择物理编号 769，那么对应的地址就是 $769 \times 512 = 393728$ ，这个是 1G 卡 FAT 格式化后的初始数据。不同容量的 SD 卡的初始地址不同，请使用 winhex 软件查看对应的物理扇区编号，并计算出对应的地址，然后在样例中图示标记位置进行更改。

Drive H: [unregistriert] 100% free File system: FAT16

Default Edit Mode State: original

Undo level: 0

Undo reverses: n/a

Alloc. of visible drive space: 2

Cluster No.: 7. bin \

Snapshot taken 0 min. ago

Physical sector No.: 769

Logical sector No.: 520

Used space: 336 KB

Free space: 344,064 bytes

Total capacity: 0.9 GB

Bytes per cluster: 16,384

Free clusters: 61,946

Total clusters: 61,967

Bytes per sector: 512

WinHex - [Drive H:]

Drive H: [unregistriert] 100% free File system: FAT16

Default Edit Mode State: original

Undo level: 0

Undo reverses: n/a

Alloc. of visible drive space: 5

Cluster No.: 1. bin \

Snapshot taken 4 min. ago

Physical sector No.: 865

Logical sector No.: 616

Used space: 336 KB

Free space: 344,064 bytes

Total capacity: 0.9 GB

Bytes per cluster: 16,384

图 2

```

initial();
SdInit();                                     //SD卡初始化
while(1)
{
    for(j=0;j<80;j++)                          //80表示一幅图片含有80x512字节的信息
    {
        SdReadBlock(DATA,AddTemp+(j*512),512); //每次读出512字节放到缓冲区
        for(i=0;i<256;i++)                     //然后写到液晶屏，可以显示256个像素，每个像素16位即
        {
            LCD_SetPos(x-4,x-4,y,y);
            write_data_2byte(DATA[2*i+1],DATA[2*i]);
            x++;

            if(x==128)                          //检测是否写到屏的边缘 128x160
            {
                y++;
                x=0;
                if(y==160)
                y=0;
            }
        }
        AddTemp = AddTemp+((j+16)*512);        //写完一幅图片后把SD地址加80x512到下一个图片地址
                                                //通过winhex 可以看出前图与后图间有16个扇区的空白区
                                                //等待按键按下继续执行循环显示下一幅图片，如果没有
    }
    while(KEY);
}

```

图 3

图 3 中所示的数据简略说明如下：

按键可以定义到单片机任意 I/O 口，这里表示写完一幅图片后如果按下按键，则显示下一张图片，否则停留在当前图片。

“j+16”的解释如下：一张完整的图片占用 40kbyte 存储空间，每个扇区大小为 512，也就是说存储一张图片需要 80 个扇区，据此可推，第一张图片的逻辑地址为 520，下一张图片的逻辑地址则应该为 520+80=600，但通过 winhex 可以看出实际的逻辑地址是 616，所以要通过在程序中加 16 来跳过这个空白区。这里的程序仅作从 SD 卡中读取图片测试用，实际 SD 卡读写需要读出引导程序，然后确定下一组数据的位置，这需要复杂的写 FAT 文件系统，有兴趣的可另去研究。