

2.8 寸 TFT 液晶模块使用手册

NBCTFT2.8_V2.2-----2012.3.29



合肥炜煌电子有限公司

www.hfwhdz.com

目录

一、产品简介3

 1、功能简介3

 2、主要参数3

二、使用说明4

 2.1、硬件介绍4

 2.1.1 FPC 接口电路4

 2.1.2 触摸控制电路5

 2.1.3 IO 扩展电路6

 2.1.4 电源电路7

 2.1.5 SD 接口电路7

 2.1.6 TFT 控制接口电路10

 2.2、程序控制实例（以 51 单片机控制为例介绍）12

 2.2.1 实例一、字符、汉字、画线显示实验12

 2.2.2 实例二、触摸屏画图演示实验14

 2.2.3 实例三、触摸屏控制直流电机实验16

 2.2.4 实例四、电子相册-SD 卡图片读取显示实验18

一、产品简介

1、功能简介

Thin Film Transistor (薄膜场效应晶体管),是指液晶显示器上的每一液晶象素点都是由集成在其后的薄膜晶体管来驱动。从而可以做到高速度高亮度高对比度显示屏幕信息。目前在手机上TFT使用最为广泛,中高端彩屏手机中普遍采用的屏幕,分65536色及26万色,1600万色三种,其显示效果非常出色。随着技术的进步,TFT不仅应用在手机上,许多智能仪表,工控人机界面也都在使用TFT取代之前的黑白屏。



2、主要参数

TFT2.8_V2.2主要有如下参数特点:

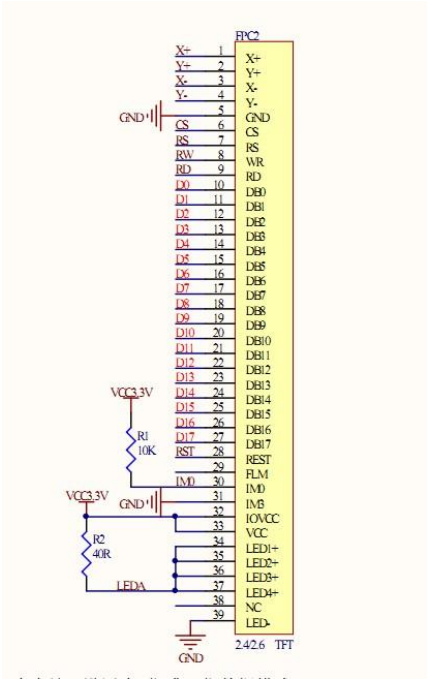
| 类目 | 参数 |
|-----------|-----------|
| 模块尺寸(长X宽) | 80mmx54mm |
| 像素 | 320x240 |
| 颜色 | 26万色 |
| 驱动IC | ILI9320 |
| 触摸类型 | 电阻式 |
| 背光类型 | LED |

二、使用说明

2.1、硬件介绍

2.8寸TFT模块电路共分为FPC接口电路、触摸控制电路、IO扩展电路、电源电路、TFT控制接口电路、SD接口电路等6部分。

2.1.1 FPC接口电路

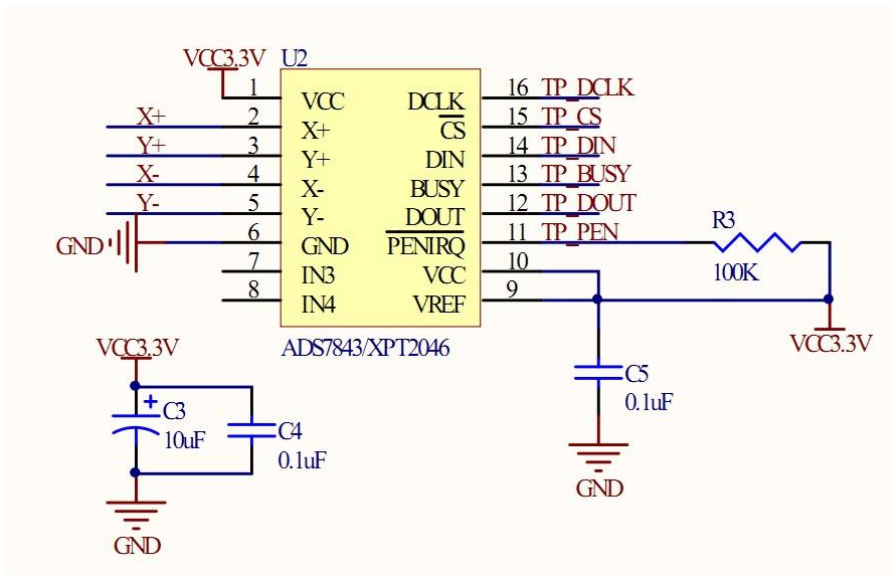


FPC接口

| Pin No. | Symbol | Level | Description |
|---------|-------------|----------------|---|
| 1 | X+ | - | Touch panel coordinate in the right side of envisage drawing. |
| 2 | Y+ | - | Touch panel coordinate in the down side of envisage drawing. |
| 3 | X- | - | Touch panel coordinate in the left side of envisage drawing. |
| 4 | Y- | - | Touch panel coordinate in the up side of envisage drawing. |
| 5 | GND | 0V | Ground. |
| 6 | /CS_MAIN | H/L | Chip selection. |
| 7 | RS | H/L | Command / data select pin. |
| 8 | /WR | H/L | Write signal. |
| 9 | /RD | H/L | Read signal. |
| 10~27 | DB00~DB17 | H/L | Data bus. |
| 28 | /RESET | H/L | Reset signal. |
| 29 | FLM | H/L | Tearing effect output pad to synchronize MCU to frame writing, activated by S/W command. When this is not activated, this pad should be low. If not in use, leave this unconnected. |
| 30 | IM0 | H/L | IM0 IM3 Interface Mode 0 0 80-system 16-bit DB17-10, DB8-1 |
| | | | 1 0 80-system 8-bit DB17-10 |
| 31 | IM3 | | 0 1 80-system 18-bit DB17-0 |
| | | | 1 1 80-system 9-bit DB17-9 |
| 32 | IOVCC | 1.8V/2.8V(typ) | I/O power supply for LCD driver. |
| 33 | VCC | 2.8V(typ) | Logic power supply. |
| 34~37 | LED1+~LED4+ | - | LED light anode. |
| 38 | NC | - | No connection. |
| 39 | LED- | - | LED light cathode. |

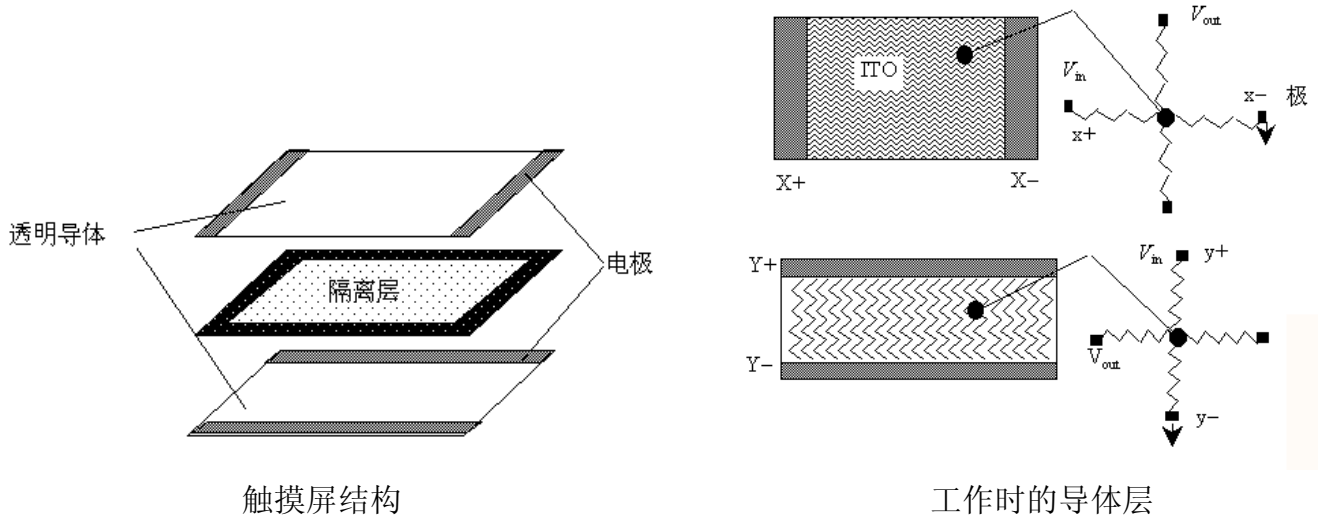
FPC排线接口定义

2.1.2 触摸控制电路



触摸控制电路

对于ADS7843/XPT2046控制IC的使用，可以阅读IC的使用手册，下面介绍一下触摸屏的工作原理。



典型触摸屏的结构一般由三部分组成：两层透明的阻性导体层、两层导体之间的隔离层、电极。触摸屏工作时，上下导体层相当于电阻网络。当某一层电极加上电压时，会在该网络上形成电压梯度。如有外力使得上下两层在某一点接触，则在电极未加电压的另一层可以测得接触点处的电压，从而知道接触点处的坐标。比如，在顶层的电极(X+,X-)上加上驱动电压，则在顶层导体层上形成电压梯度，当有外力使得上下两层在某一点接触，在底层就可以测得接触点处的电压，再根据该电压与电极(X+)之间的距离关系，知道该处的X坐标。然后，将驱动电压切换到底层电极（Y+,Y-）上，并在顶层测量接触点处的电压，从而知道Y坐标。

计算触点的X，Y坐标具体分为如下两步：

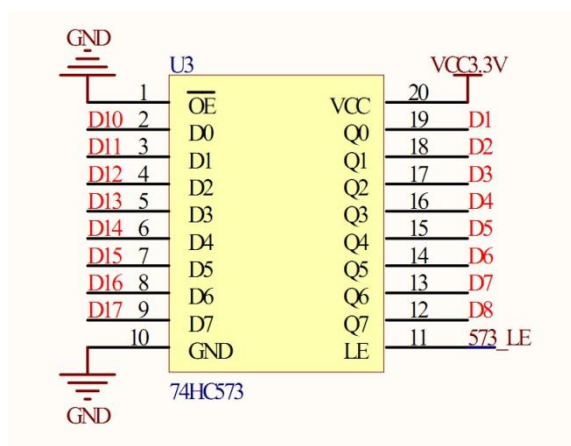
1. 通过触摸屏控制芯片ADS7843/XPT2046分别采集触点在X轴和Y轴上产生的电压数字量，如下面公式中的X值和Y值。
2. 计算坐标

$$X_{LCD} = \frac{X - X_{min}}{X_{max} - X_{min}} \times W$$
$$Y_{LCD} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \times H$$

式中，X和Y分别为触点在X工作面和Y工作面上产生的电压的数字量的测量值；（X，Y）反映了触点在触摸屏上的坐标。Xmin，Ymin，Xmax和Ymax分别为触摸屏上最小和最大坐标点在X轴和Y轴上产生的电压的数字量的实际测量值，它们是常量，可通过测量得到；（Xmin，Ymin）和（Xmax，Ymax）反映了触摸屏上最小、最大坐标点的坐标；W和H分别是LCD显示屏X轴和Y轴上的像素点总数，例如本2.8寸TFT，W为240，H为320；（XLCD，YLCD）为触点映射到LCD显示屏上的像素点坐标。

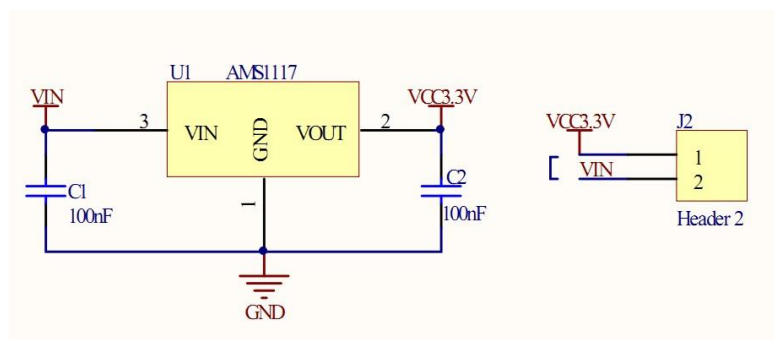
2.1.3 IO扩展电路

为了节省IO资源，通过IO扩展仅需要使用8位控制器的IO口即可驱动16位数据模式，电路如下图所示。



控制器的8位IO与锁存器的D0~D7连接，锁存器的输出与TFT的低8位数据口连接，当写16位数据时，先将16位数据的低8位送到D10~D17，此时通过控制LE管脚将数据锁存到TFT的低8位数据口，然后再将16位数据的高8位送到D10~D17，即完成16位数据送至TFT数据口的工作。

2.1.4 电源电路

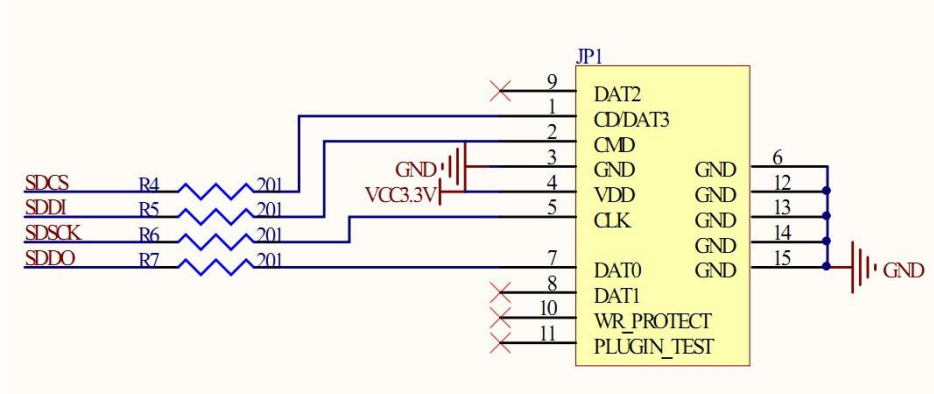


由于TFT只能在3.3V电压下工作，所以当输入电压VIN为5V时，需要通过3.3V稳压IC降到3.3V电压，当输入电压为3.3V时，需要使用0欧电阻将J2短接，相当于不通过稳压IC直接给模块供电。

注意：出厂的TFT模块，均为默认5V供电。

2.1.5 SD接口电路

SD卡控制电路如下图所示，接口为SPI接口，读写实现较为简单。



后面实例四程序的功能是读取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：








TFT彩屏分辨率为320x240，程序控制中用16bit表示一个点的颜色，即为2个字节，所以如果完整写一幅图片，大小为320x240x2=153600字节，即需要150k存储空间，单片机一般ROM不够大，所以需要SD卡存储图片。

SD卡使用如下：

1、SD卡要格式化成FAT(即FAT16格式)，然后把需要显示的大小为320x240像素的bmp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（打开winhex.exe，点击Tools->Open Disk找到SD卡打开）查看各图片存储的地址。打开软件如图所示，点击第一张图片“车.bin”，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，那么对应的地址就是769x512=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。

| | | | | | | | | | | | | | | |
|--|-----|---------|------------|-------|------------|-------|------------|---|--|--|--|-----|--|-------|
| (Root directory) | | 16.0 KB | | | | | | | | | | 488 | | |
|  车.bin | bin | 150 KB | 2011-08-19 | 18... | 2009-12-16 | 22... | 2011-08-19 | A | | | | | | 520 |
|  花.bin | bin | 150 KB | 2011-08-19 | 18... | 2009-12-16 | 23... | 2011-08-19 | A | | | | | | 840 |
|  美女1.bin | bin | 150 KB | 2011-08-19 | 18... | 2009-12-16 | 22... | 2011-08-19 | A | | | | | | 1,160 |
|  美女2.bin | bin | 150 KB | 2011-08-19 | 18... | 2009-12-16 | 22... | 2011-08-19 | A | | | | | | 1,480 |
|  美女3.bin | bin | 150 KB | 2011-08-19 | 18... | 2009-12-16 | 23... | 2011-08-19 | A | | | | | | 1,800 |
|  美女4.bin | bin | 150 KB | 2011-08-19 | 18... | 2009-12-16 | 23... | 2011-08-19 | A | | | | | | 2,120 |
|  香车美女.bin | bin | 150 KB | 2011-08-19 | 18... | 2009-12-16 | 22... | 2011-08-19 | A | | | | | | 2,440 |

| | | | | | | | | | | | | | | | | | | |
|--------------------------------|---------------------|----------|-------|----|----|----|----|----|----|----|-----|----|--------|----|-----|----|----|----|
| [unregistriert] | | Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Drive M: | 100% free | 00041000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| File system: | FAT16 | 00041010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Default Edit Mode | | 00041020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| State: | original | 00041030 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Undo level: | 0 | 00041040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Undo reverses: | n/a | 00041050 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | | 00041060 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Alloc. of visible drive space: | | 00041070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Cluster No.: | 2 | 00041080 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | 车.bin | 00041090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | \ | 000410A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Snapshot taken | 1 min. ago | 000410B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Physical sector No.: | 769 | 000410C0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Logical sector No.: | 520 | 000410D0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | | 000410E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| Used space: | 1.1 MB | 000410F0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| | 1,146,880 bytes | 00041100 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| Free space: | 0.9 GB | 00041110 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| | 1,014,120,448 bytes | 00041120 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| Total capacity: | 0.9 GB | 00041130 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| | 1,015,549,440 bytes | 00041140 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| | | 00041150 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| Bytes per cluster: | 16,384 | 00041160 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| Free clusters: | 61,897 | 00041170 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| Total clusters: | 61,967 | 00041180 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| | | 00041190 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| Bytes per sector: | 512 | 000411A0 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| Usable sectors: | 1,982,944 | 000411B0 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 | 01 | 00 |
| First data sector: | 520 | | | | | | | | | | | | | | | | | |
| Physical disk: | 1 | | | | | | | | | | | | | | | | | |
| Sector 520 of 1983495 | | Offset: | 41000 | | | | | | | | = 0 | | Block: | | n/s | | | |


```

unsigned int x,y; //定义液晶屏坐标
unsigned long j; //执行循环需要的临时变量
unsigned int i;
unsigned long AddTemp=393728; //SD卡地址第一个数据物理地址初始值，可以用winhex查看，
//这里是512扇区，769x512=393728，根据实际SD卡内容更改

CS=1;
delayms(5);
RES=0;
delayms(5);
RES=1;
delayms(5);
ILI9325_Initial(); //液晶屏初始化
SdInit(); //SD卡初始化
while(1)
{
for(j=0;j<300;j++) //300表示一幅图片含有300x512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2个字节
{
LCD_SetPos(x,x,y,y);
Write_Data(DATA[2*i+1],DATA[2*i]);
x++;
if(x==240) //检测是否写到屏的边缘 240x320
{
y++;
x=0;
if(y==320)
y=0;
}
}
}
AddTemp = AddTemp+((j+20)*512); //写完一幅图片后把SD地址加300x512到下一个图片地址
while(KEY); //等待按键按下继续执行循环显示下一幅图片，如果没有按下则等待
// while (AddTemp>2621440)
// j=0,AddTemp=721408;
}
}

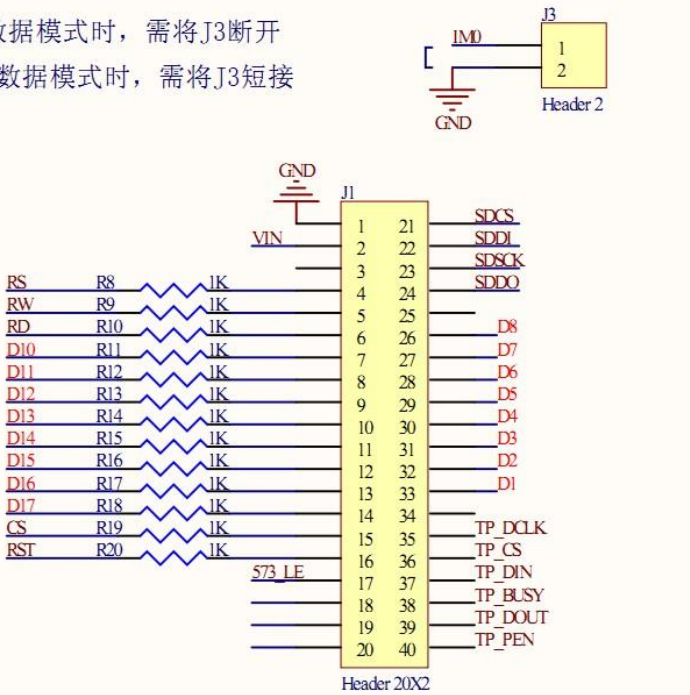
```

图中所示的数据简略说明如下：

“j+20”的解释如下：一张完整的图片占用150k存储空间，每个扇区大小为512，也就是说存储一张图片需要300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+300=820，但通过winhex可以看出实际的逻辑地址是840，物理地址也是等差增加，所以要通过在程序中加20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。

2.1.6 TFT控制接口电路

8位数据模式时，需将J3断开
16位数据模式时，需将J3短接



图中第1脚到第20脚是为兼容12864接口而设计：

- 第1脚为电源地；
- 第2脚为电源输入；
- 第4脚到第16脚与控制器间增加了一个限流电阻，作用是在使用5V控制器驱动TFT时，保护TFT以免被过流损坏；
- 第17脚为IO扩展控制位；

第21脚到第24脚为SD卡控制接口；

第26脚到第33脚为TFT数据口的低8位，由于模块上增加了IO扩展，建议用户不作使用。

第35脚到第40脚为触摸控制接口；

| 引脚号 | 引脚名称 | 功能说明 |
|-----|------|---------------------------|
| 1 | GND | 模块的电源地 |
| 2 | VIN | 模块的电源正端（3.3V 供电时需将 J2 短接） |
| 3 | NC | 空脚 |
| 4 | RS | 并行的指令/数据选择信号； |
| 5 | R/W | 并行的读写选择信号； |
| 6 | RD | 读控制信号 |
| 7 | D10 | 数据口高 8 位 |
| 8 | D11 | |
| 9 | D12 | |
| 10 | D13 | |
| 11 | D14 | |
| 12 | D15 | |
| 13 | D16 | |
| 14 | D17 | |
| 15 | CS | 片选 |
| 16 | RST | 复位 |
| 17 | LE10 | 74HC573 锁存控制 |

| | | |
|----|------|----------------------|
| 18 | NC | 空脚 |
| 19 | NC | 空脚 |
| 20 | NC | 空脚 |
| | | |
| 21 | SDCS | SD 卡片选信号 |
| 22 | SDDI | SD 卡串行数据输入 |
| 23 | SCK | SD 卡时钟信号 |
| 24 | SDDO | SD 卡串行数据输出 |
| 25 | NC | 数据口低 8 位 |
| 26 | D8 | |
| 27 | D7 | |
| 28 | D6 | |
| 29 | D5 | |
| 30 | D4 | |
| 31 | D3 | |
| 32 | D2 | |
| 33 | D1 | |
| 34 | NC | |
| 35 | CLK | 触摸屏外部时钟输入 |
| 36 | TPCS | 触摸屏片选信号 |
| 37 | TPDI | 触摸屏串行数据输入，在时钟上升沿数据移进 |
| 38 | BUSY | 触摸屏忙指示，低电平有效 |
| 39 | TPDO | 触摸屏串行数据输出，在时钟下降沿数据移出 |
| 40 | PEN | 触摸屏中断输出 |

2.2、程序控制实例（以51单片机控制为例介绍）

2.2.1 实例一、字符、汉字、画线显示实验

主程序：

```
main()
{
    unsigned int i;
    Device_code=0x9320;          //TFT控制IC型号
    TFT_Initial();               //初始化LCD

    while(1)                     //循环
    {
        //5种颜色将屏分为5个区域
        Show_RGB(0,240,0,64,Blue);
        Show_RGB(0,240,64,128,Green);
        Show_RGB(0,240,128,192,Magenta);
        Show_RGB(0,240,192,256,Red);
        Show_RGB(0,240,256,320,Yellow);

        //第1个区域显示16x16汉字
        LCD_PutString(24,16,"北方蓝芯科技开发有限公司",White,Blue);
        LCD_PutString(24,40,"  蓝芯科技  与您同行",White,Blue);

        //第2个区域显示ascii字符中的数字和符号
        LCD_PutString(80,72,"0123456789",Black,Green);
        LCD_PutString(16,96,"( ` , ./ < > ' : [ ] { } \ | ? - = + * & ^ % $ " , Black,Green);

        //第3个区域显示ascii字符中字母
        LCD_PutString(16,136,"abcdefghijklmnopqrstuvwxyz",Blue,Magenta);
        LCD_PutString(16,156,"ABCDEFGHIJKLMNOPQRSTUVWXYZ",Blue,Magenta);

        //第4个区域显示TFT控制器IC型号
        LCD_PutString(16,200,"The Device IC Of TFT Is:",Black,Red);
        LCD_PutString(96,224,"ILI",Black,Red);
        LCD_PutChar(120, 224, 0x30+(Device_code>>12), Black, Red);
        LCD_PutChar(128, 224, 0x30+((Device_code>>8)&0x000f), Black, Red);
        LCD_PutChar(136, 224, 0x30+((Device_code>>4)&0x000f), Black, Red);
        LCD_PutChar(144, 224, 0x30+(Device_code&0x000f), Black, Red);

        //第5个区域打点实现画线
        for(i=312;i>264;i--){ Put_pixel(68,i,Blue);}
        for(i=68;i<172;i++){ Put_pixel(i,264,Blue);}
        for(i=264;i<312;i++){ Put_pixel(172,i,Blue);}
        for(i=172;i>68;i--){ Put_pixel(i,312,Blue);}
```

```
delayms(20000); //显示一段时间
CLR_Screen(Red); //用背景色清屏
}
}
```

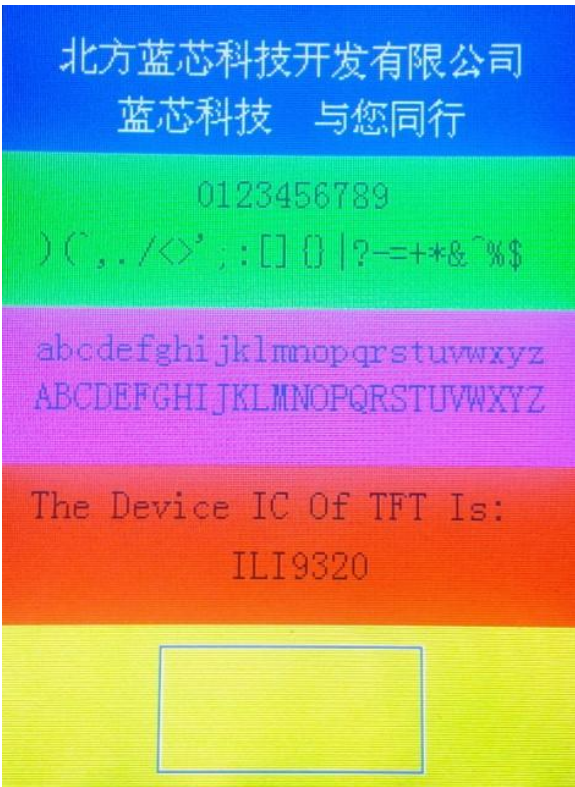
硬件接线连接:

| 51单片机IO | TFT接口 | 说明 |
|----------|---------|-------------|
| TFT控制接线 | | |
| P0 | D10~D17 | TFT数据口 |
| P2^5 | RS | TFT指令/数据选择 |
| P2^4 | RW | TFT写指令/数据选择 |
| P2^3 | RD | TFT读控制信号 |
| P2^2 | CS | TFT片选 |
| P2^1 | RES | TFT复位 |
| P2^0 | LE | 74HC573锁存信号 |
| 电源供电接线 | | |
| 电源地 | GND | |
| 电源正极(5V) | VIN | |

注意：如果本模块在NBC开发板上使用时，只需将TFT对应插接在12864接口上即可（TFT的1脚对应12864接口1脚，请勿接错!!!），本实验无需使用杜邦线连接。

程序运行效果图:

运行程序，TFT显示内容如下图，由于拍摄问题，下面图片和实际效果相比，色彩有较大失真。



2.2.2 实例二、触摸屏画图演示实验

主程序：

```
main()
{
    Device_code=0x9320;
    TFT_Initial();    //TFT初始化
    start_7843();

    CLR_Screen(Blue); //清屏

    LCD_PutString(16,5,"Please write on the board!",Magenta,Blue);
    while(1)
    {

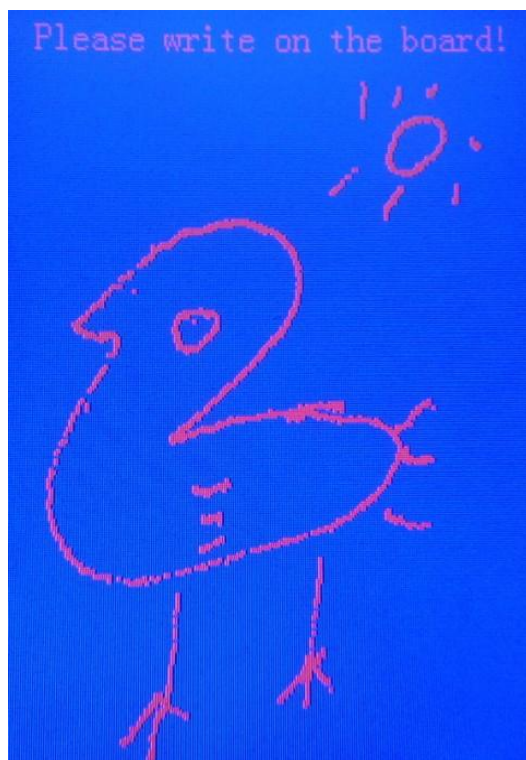
        if(Getpix()==1)    //采集触点坐标
            drawpoint(lx,ly,Red);    //在触点写红色
    }
}
```

硬件接线连接：

| 51单片机IO | TFT接口 | 说明 |
|----------|---------|--------------|
| TFT控制接线 | | |
| P0 | D10~D17 | TFT数据口 |
| P2^5 | RS | TFT指令/数据选择 |
| P2^4 | RW | TFT写指令/数据选择 |
| P2^3 | RD | TFT读控制信号 |
| P2^2 | CS | TFT片选 |
| P2^1 | RES | TFT复位 |
| P2^0 | LE | 74HC573锁存信号 |
| 触摸控制接线 | | |
| P3^2 | PEN | 触摸屏中断输出 |
| P3^3 | TPDO | 触摸屏串行数据输出 |
| P3^4 | BUSY | 触摸屏忙指示，低电平有效 |
| P3^5 | TPDI | 触摸屏串行数据输入 |
| P3^6 | TPCS | 触摸屏片选信号 |
| P3^7 | CLK | 触摸屏外部时钟输入 |
| 电源供电接线 | | |
| 电源地 | GND | |
| 电源正极(5V) | VIN | |

注意：如果本模块在NBC开发板上使用时，将TFT对应插接在12864接口上（TFT的1脚对应12864接口1脚，请勿接错!!!），TFT控制接线与电源供电接线部分无需使用杜邦线连接，触摸控制接线部分需使用杜邦线连接。

程序运行效果图：



2.2.3 实例三、触摸屏控制直流电机实验

主程序：

```
main()
{
    Device_code=0x9320;
    speed=5;

    TFT_Initial();           //tft初始化
    start_7843();           //初始化触摸控制IC
    TO_init();              //定时器初始化
    init_interface();       //初始化人机控制界面

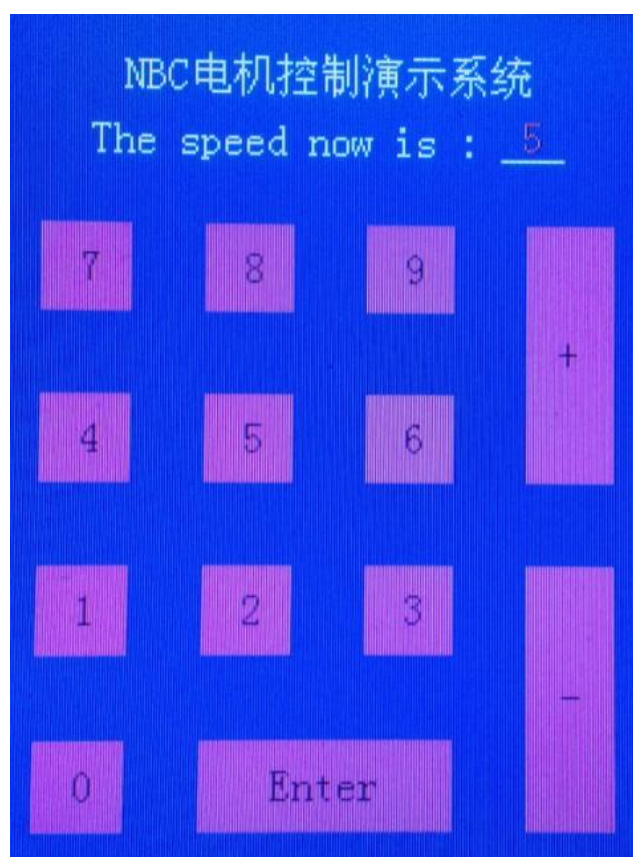
    while(1)
    {
        if(Getpix()==1)key_scan();    //扫描触摸按键实现电机调速
    }
}
```

硬件接线连接：

| 51单片机IO | TFT接口 | 说明 |
|-------------------|----------|--------------|
| TFT控制接线 | | |
| P0 | D10~D17 | TFT数据口 |
| P2^5 | RS | TFT指令/数据选择 |
| P2^4 | RW | TFT写指令/数据选择 |
| P2^3 | RD | TFT读控制信号 |
| P2^2 | CS | TFT片选 |
| P2^1 | RES | TFT复位 |
| P2^0 | LE | 74HC573锁存信号 |
| 触摸控制接线 | | |
| P3^2 | PEN | 触摸屏中断输出 |
| P3^3 | TPDO | 触摸屏串行数据输出 |
| P3^4 | BUSY | 触摸屏忙指示，低电平有效 |
| P3^5 | TPDI | 触摸屏串行数据输入 |
| P3^6 | TPCS | 触摸屏片选信号 |
| P3^7 | CLK | 触摸屏外部时钟输入 |
| 直流电机控制接线 | | |
| P1^0 | J18_DCMT | PWM驱动直流电机 |
| 将直流电机连接到主板上的JP7接口 | | |
| 电源供电接线 | | |
| 电源地 | GND | |
| 电源正极(5V) | VIN | |

注意：如果本模块在NBC开发板上使用时，将TFT对应插接在12864接口上（TFT的1脚对应12864接口1脚，请勿接错!!!），TFT控制接线与电源供电接线部分无需使用杜邦线连接，触摸控制接线与直流电机控制接线部分需使用杜邦线连接。

程序运行效果图：



控制说明：

- 1、0~9代表电机转速等级，按下某一等级值，例如按下数字“5”，然后按一下“Enter”键，此时，电机转速变为第8级，数字越大，转速越快，输入0时，电机速度为0；
- 2、触摸按键“+”和“-”直接实现电机加减速，无需按下“Enter”键。

2.2.4 实例四、电子相册-SD卡图片读取显示实验

主程序:

```
main()
{
    unsigned int x,y; //定义液晶屏坐标
    unsigned long j;  //执行循环需要的临时变量
    unsigned int i;
    unsigned long AddTemp=288256;//SD卡地址第一个数据物理地址初始值，可以用winhex查看，
    //这里是第563扇区，每个扇区512字节，563x512=288256，根据实际SD卡内容更改
    Device_code=0x9320;
    TFT_Initial();           //初始化LCD
    CLR_Screen(Blue);       //用背景色清屏
    SdInit();               //SD卡初始化

    while(1)
    {
        for(j=0;j<300;j++)    //300表示一幅图片含有300x512(320x240x2)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2个字节
            {
                LCD_SetPos(x,x,y,y);
                Write_Data(DATA[2*i+1],DATA[2*i]);
                x++;
                if(x==240)        //检测是否写到屏的边缘 240x320
                {
                    y++;
                    x=0;
                    if(y==320)
                    y=0;
                }
            }
        }
        AddTemp = AddTemp+(j*512); //使用32M的SD卡增加为300,其他SD卡实际增加值一般为320
        // AddTemp = AddTemp+((j+20)*512); //写完一幅图片后把SD地址加320x512到下一个图片地址
        while(PEN);    //等待按键按下继续执行循环显示下一幅图片，如果没有按下则等待
    }
}
```

注意：以上程序中两行加下划线的语句，用户需根据实际SD卡容量修改。

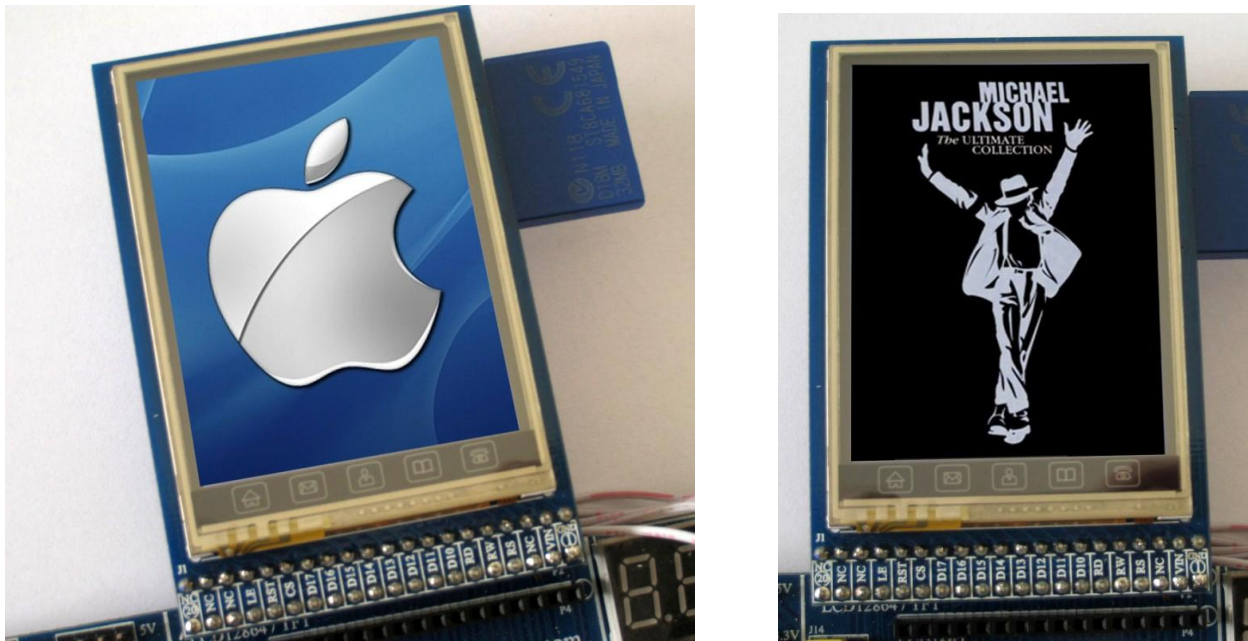
硬件接线连接:

| | | |
|---------|---------|--------|
| 51单片机IO | TFT接口 | 说明 |
| TFT控制接线 | | |
| P0 | D10~D17 | TFT数据口 |

| | | |
|----------|------|--------------|
| P2^5 | RS | TFT指令/数据选择 |
| P2^4 | RW | TFT写指令/数据选择 |
| P2^3 | RD | TFT读控制信号 |
| P2^2 | CS | TFT片选 |
| P2^1 | RES | TFT复位 |
| P2^0 | LE | 74HC573锁存信号 |
| SD卡控制接线 | | |
| P1^3 | SDCS | 触摸屏中断输出 |
| P1^2 | SDDI | 触摸屏串行数据输出 |
| P1^1 | SCK | 触摸屏忙指示，低电平有效 |
| P1^0 | SDDO | 触摸屏串行数据输入 |
| 触摸控制接线 | | |
| P3^7 | PEN | 触摸控制图片翻页 |
| 电源供电接线 | | |
| 电源地 | GND | |
| 电源正极(5V) | VIN | |

注意：如果本模块在NBC开发板上使用时，将TFT对应插接在12864接口上（TFT的1脚对应12864接口1脚，请勿接错!!!），**TFT控制接线**与**电源供电接线**部分无需使用杜邦线连接，**SD卡控制接线**与**触摸控制接线**部分需使用杜邦线连接。

程序运行效果图：



控制说明：触碰TFT屏，实现图片翻页功能，显示下一张。