

OpenJumper

# TFT 液晶屏 使用说明书

OpenJumper

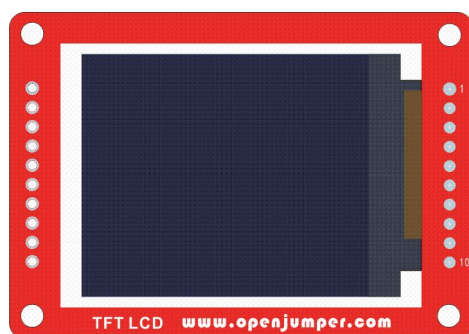
## 一、概述

OJ TFT 液晶屏是一款由 SPI 协议驱动的 128 x 160 彩色液晶模块，含有背光及插针。使用 TFT library 库文件，你就可以显示文本、图片等等。液晶背板载有 micro-SD 卡槽，可以直接调用存储于 TF 卡内的 24bit BMP 图片，并让液晶屏显示出来。

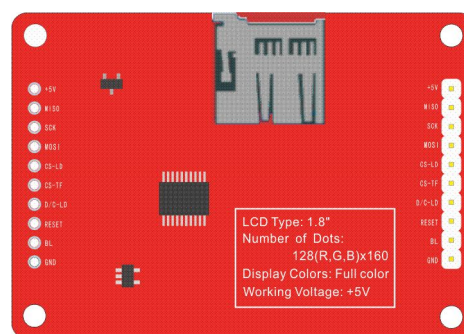
OJ TFT 液晶屏的引脚接口完全兼容 Arduino Esplora，可以直接插在 Arduino Esplora 的扩展口上。OJ TFT 液晶屏也可以用 Arduino UNO，Arduino Leonardo，Arduino DUE 等主控器进行驱动显示。

## 二、特点

- 1、液晶屏尺寸为 1.8”，点阵格式为 160\*128。
- 2、液晶屏工作电压为 DC 5V。
- 3、板载 Micro-SD 卡槽，配合 SD 库文件可以进行 TF 卡操作。
- 4、可用 PWM 调节 LED 背光。
- 5、屏幕引脚接口可方便连接至 Arduino Esplora 或 Ardiuno Robot。
- 6、通过使用 TFT 库文件，连接 TFT 屏至 Arduino 的 SPI 通信接口就可以很方便的进行显示。
- 7、液晶屏功耗：背光不亮 10mA      背光亮      60mA



a.正面



b.反面

引脚号	标识	功能
1	+5V	电源 DC 5V+
2	MISO	SPI 接口
3	SCK	
4	MOSI	
5	CS-LD	液晶屏使能
6	CS-TF	TF 卡使能
7	D/C-LD	液晶屏数据/指令控制
8	RESET	液晶屏复位

9	BL	液晶屏背光控制： 背光亮：置 High，并可通过 PWM 控制亮度 背光灭：置 Low 或悬空
10	GND	电源 DC 地

因本 TFT 液晶屏的驱动芯片的寄存器与官方的少许差异，需单独下载并使用 TFT 库文件。

将下载的库文件放入 arduino IDE 的 libraries 文件夹中后，打开 arduino IDE，你会看到在 IDE 的【示例】里多出了一个 TFT 库，选中它会出现两个选项，一个是【Esplora】，一个是【arduino】。如果你使用的是一般的 Arduino 控制器来驱动，请选择【arduino】文件夹下的例程，如果你使用的是 Arduino Esplora 控制器，那么请选择【esplora】下的例程，这是因为 Arduino Esplora 的集成度更高，并且有扩展 I/O IC，所以是不兼容的，这一点一定要注意。

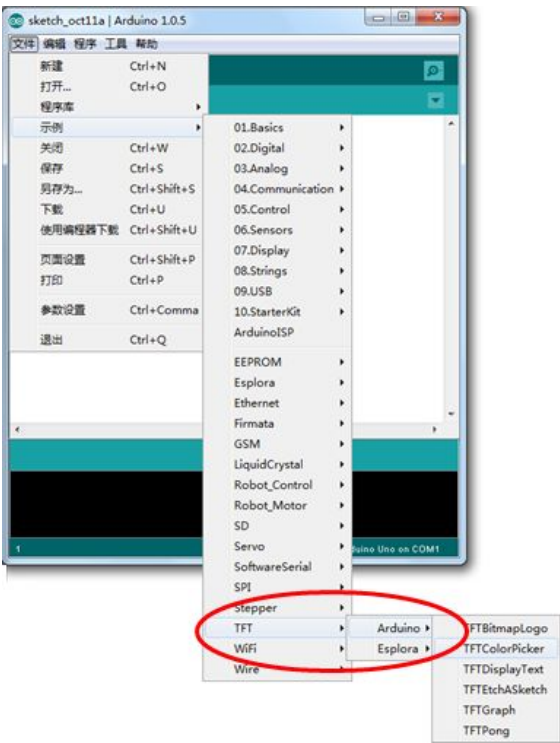


图 2:

## 1、显示文字

我们以连接 Arduino UNO 为例。

连接示意图

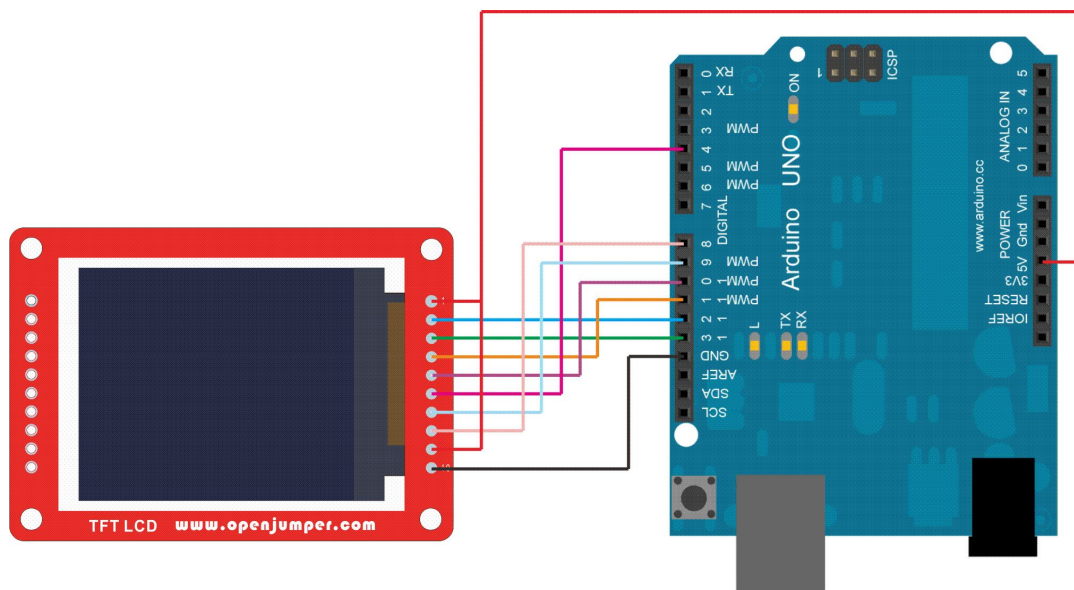


图 3：连接示意图

表 1：引脚连接表

TFT LCD 引脚编号		连接	Arduino UNO 引脚编号
1	+5V	↔	5V
2	MISO	↔	12
3	SCK	↔	13
4	MOSI	↔	11
5	CS-LD	↔	10
6	CS-TF	↔	4
7	D/C-LD	↔	9
8	RESET	↔	8
9	BL	↔	5V
10	GND	↔	GND

打开【TFTDisplayText】这个例子，写入 Arduino UNO，你就可以观察到屏幕在不断的显示模拟口 A0 读取到的信号数值：

/\*

Arduino TFT text example

This example demonstrates how to draw text on the TFT with an Arduino. The Arduino reads the value of an analog sensor attached to pin A0, and writes the value to the LCD screen, updating every quarter second.

This example code is in the public domain

Created 15 April 2013 by Scott Fitzgerald

```

http://arduino.cc/en/Tutorial/TFTDisplayText
*/

#include <TFT.h> // Arduino LCD library
#include <SPI.h>

//pin definition for the Uno 采用 Arduino Uno 时的连接定义
#define lcd_cs 10 // TFT 板定义 CS-LD 接到 UNO 数字 I/O 口 10 号端口
#define dc 9 // TFT 板定义 D/C-LD 接到 UNO 数字 I/O 口 9 号端口
#define rst 8 // TFT 板定义 RESET 接到 UNO 数字 I/O 口 8 号端口

// pin definition for the Leonardo 采用 Arduino Leonardo 时的连接定义
// #define cs 7
// #define dc 0
// #define rst 1

// create an instance of the library
TFT TFTscreen = TFT(cs, dc, rst);

// char array to print to the screen
char sensorPrintout[4];

void setup() {
    // Put this line at the beginning of every sketch that uses the GLCD:
    TFTscreen.begin(); //清除屏幕并初始化屏幕

    // clear the screen with a black background
    TFTscreen.background(0, 0, 0); //设置屏幕背景颜色为黑色

    // write the static text to the screen
    // set the font color to white
    TFTscreen.stroke(255,255,255); //设置文字颜色为白色
    // set the font size
    TFTscreen.setTextSize(2); //设置显示文字大小
    // write the text to the top left corner of the screen
    TFTscreen.text("Sensor Value :\n",0,0); //从屏幕左上角开始输出字符“Sensor Value
    (传感器数值)”
    // ste the font size very large for the loop
    TFTscreen.setTextSize(5); //再次设置文字大小，较之前更大，让传感器数值显示更加
    清楚
}

void loop() {
    // Read the value of the sensor on A0

```

```
String sensorVal = String(analogRead(A0)); //读取 A0 模拟量值

// convert the reading to a char array 转换为 char 数组
sensorVal.toCharArray(sensorPrintout, 4);

// set the font color
TFTscreen.stroke(255,255,255); //设置文字颜色为白色
// print the sensor value 指定位置输出传感器数值
TFTscreen.text(sensorPrintout, 0, 20);
// wait for a moment
delay(250);
// erase the text you just wrote 擦除输出的数值
TFTscreen.stroke(0,0,0); //设置文字颜色为黑色
TFTscreen.text(sensorPrintout, 0, 20);
}
```

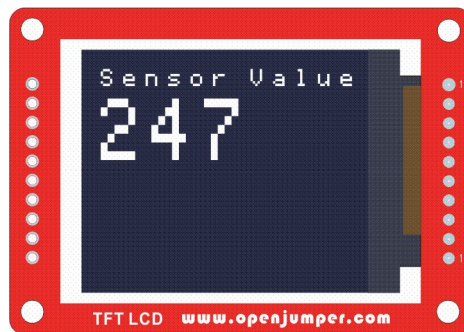


图 4：液晶显示输出示意图

## 2、显示图片

到 [arduino](http://arduino.cc/en/Tutorial/TFTBitmapLogo) 官网下载例程测试用 bmp 图片：

<http://arduino.cc/en/Tutorial/TFTBitmapLogo>

也可以自行设计一张像素在 24bit 以下的 BMP 图片，彩色黑白均可。图片像素大小需控制在液晶点阵范围内，即 128\*160。

**注意：暂不支持 32bit 的 BMP 图片及 JPG 格式文件。**

下面的例程就是调用我们自行设计的图片。



图片名称: openjumper.bmp

图片大小: 155\*103

像素 Bit: 24 位

例程代码:

```
#include <SPI.h>
#include <SD.h>
#include <TFT.h> // Arduino LCD library

// pin definition for the Uno 采用 Arduino Uno 时的连接定义
#define sd_cs 4 //TFT 板定义 CS-TF 接到 UNO 数字 I/O 口 4 号端口
#define lcd_cs 10 // TFT 板定义 CS-LD 接到 UNO 数字 I/O 口 10 号端口
#define dc 9 // TFT 板定义 D/C-LD 接到 UNO 数字 I/O 口 9 号端口
#define rst 8 // TFT 板定义 RESET 接到 UNO 数字 I/O 口 8 号端口

// pin definition for the Leonard 采用 Arduino Leonardo 时的连接定义
// #define sd_cs 8
// #define lcd_cs 7
// #define dc 0
// #define rst 1

TFT TFTscreen = TFT(lcd_cs, dc, rst);

// this variable represents the image to be drawn on screen
PImage logo;

void setup() {
  // initialize the GLCD and show a message
  // asking the user to open the serial line
  TFTscreen.begin(); //清除屏幕并初始化屏幕
  SPI.setClockDivider(SPI_CLOCK_DIV16); //针对个别 SD 卡初始化成功率较低, 可调整 DIV 参数降低 SPI 速率。参数值 128, 64, 32, 16, 8, 4, 2, 数值越大速率越低。
  TFTscreen.background(255, 255, 255); //设置屏幕背景颜色为白色

  TFTscreen.stroke(0, 0, 255); //设置文字颜色为红色
  TFTscreen.println();
```

```

TFTscreen.println("Arduino TFT Bitmap Example");
TFTscreen.stroke(0, 0, 0);  //设置文字颜色为黑色
TFTscreen.println("Open serial monitor");
TFTscreen.println("to run the sketch");

// initialize the serial port: it will be used to
// print some diagnostic info
Serial.begin(9600);
while (!Serial) {
    // wait for serial line to be ready
}

// clear the GLCD screen before starting
TFTscreen.background(0, 0, 0);  //设置屏幕背景颜色为黑色

// try to access the SD card. If that fails (e.g.
// no card present), the setup process will stop.
Serial.print("Initializing SD card...");
if (!SD.begin(sd_cs)) {
    Serial.println("failed!");  //SD 卡初始化失败
    return;
}
Serial.println("OK!");  //SD 卡初始化成功

// initialize and clear the GLCD screen
TFTscreen.begin();
SPI.setClockDivider(SPI_CLOCK_DIV2);  //对应上面 SPI 速率参数的调整，再调整为高
速
TFTscreen.background(255, 255, 255);

// now that the SD card can be access, try to load the
// image file.
logo = TFTscreen.loadImage("openjumper.bmp");  //调取 bmp 图片
if (!logo.isValid()) {
    Serial.println("error while loading openjumper.bmp");
}
}

void loop() {
    // don't do anything if the image wasn't loaded correctly.
    if (logo.isValid() == false) {
        return;
    }
}

```



```
Serial.println("drawing image");

// To avoid the image to be draw outside the screen,
// take into account the image size.
int x = 2; //图片显示起始 X, Y 轴位置
int y = 13;

// draw the image to the screen
TFTscreen.image(logo, x, y); //显示图片
}
```



图 5：串口输出信息