

2.8" TFT Touch Shield V2.0

This is a multifunctional Arduino/Seeeduno/Arduino Mega compatible resistive touch screen. It can be used as display device or sketch pad. Compared with the previous version, 2.8" TFT Touch Shield V1.0, we improved the screen driver with a professional chip, ILI9341 driver, providing pin-saving SPI communication without sacrificing the data transmitting speed. With a SD card module integrated also on this shield, this shield reserves great room for other expansions to your project.



Features

- Big screen for easy and comfortable experience
- Backlight controllable via programming
- 65535 rich colors
- SPI Pin-saving communication method
- Full screen touch active range

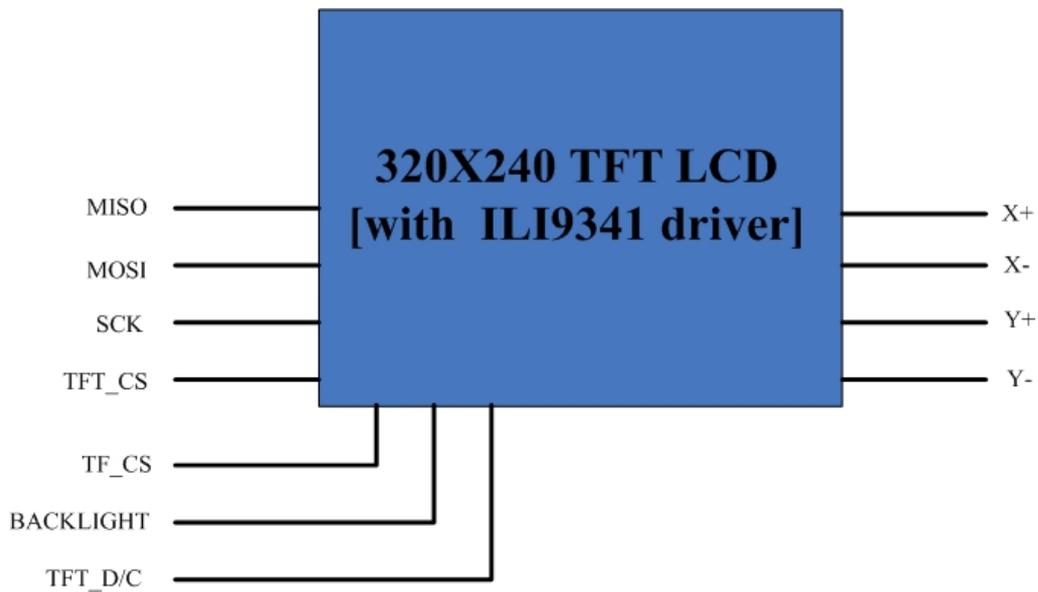
Specification

Item	Min	Typical	Max	Unit
Voltage	4.5	5	5.5	VDC
Current	/	/	250	mA
LCD Panel Size	2.8			inch
View angle	60~120			Deg
Resolution	320x240			/
LCD color	65k			/
Backlight Type	LED			/
LCD driver IC	ILI9341			/
Interface Type	SPI			/
Touch Screen	4-Wire resistive touch screen			/
Active area	43.2*57.3			mm
ESD contact discharge	±4			KV
ESD air discharge	±8			KV
Dimension	72.5x54.7x18			mm
Weight	24±2			g

Cautions

- Be careful to deposit the fragile screen.
- Don't press too hard on the screen which might cause display distortion.

Pins usage on Arduino



Pins Used for TFT Screen Control:

D4 : TF_CS, SD card select input pin

D5 : TFT_CS, TFT chip select input pin

D6 : TFT_D/C, TFT Data/Command control pin

D7 : BACKLIGHT, TFT backlight control pin

Pins Used for SPI Interface

D10 : SPI chip select

D11 : SPI data pin

D12 : SPI data pin

D13 : SPI serial clock pin

Pins Used for Touch Function

A0 - Touch Screen Y- input pin.

A1 - Touch Screen X- input pin.

A2 - Touch Screen Y+ input pin.

A3 - Touch Screen X+ input pin.

Usage

Hardware Installation

Plug the shield onto your Arduino and Connect the board to PC using USB cable..

Software Installation

1. Download the [SeeedTFTV2.0 Library for Arduino](#) and [SeeedTouchScreen Library for Arduino](#)
2. unzip them in the libraries file of Arduino IDE by the path: ..\arduino-1.0\libraries

Demo 1 : Draw A Circle

1. Restart the Arduino IDE. Open "drawCircle" example via the path: File --> Examples --> SeeedTFTv2 --> drawCircle. P.S: There are many other examples in this library, like "drawLines". You can try them out to figure out how to use corresponding commands.
2. Learn about the command. In "drawCircle", the function we use is described as follow.

Name: drawCircle(int poX, int poY, int r, INT16U color)

Function: Create a circle of which the center is located at (poX, poY) and the radius is r. The last parameter is used to define the color of the line.

Name: fillCircle(int poX, int poY, int r, INT16U color)

Function: Fill the circle of which the center is located at (poX, poY) with the color defined within the range of a radius of r.

3. Upload the code. There should come up several circles on the screen as shown below.



Please have a try on other examples fulfilling the basic patterns drawing functions and learn about how

to draw dots, lines, rectangle and other patterns. After that, you are ready to customize your own patterns.

Demo 2: Display Images

1. Open the "tftbmp" example in the same way as open the "drawCircle" above.
2. Make sure you have stored images for display in the root folder of the micro SD card. And the format of those images must be ".bmp". The image size must be 320X240.

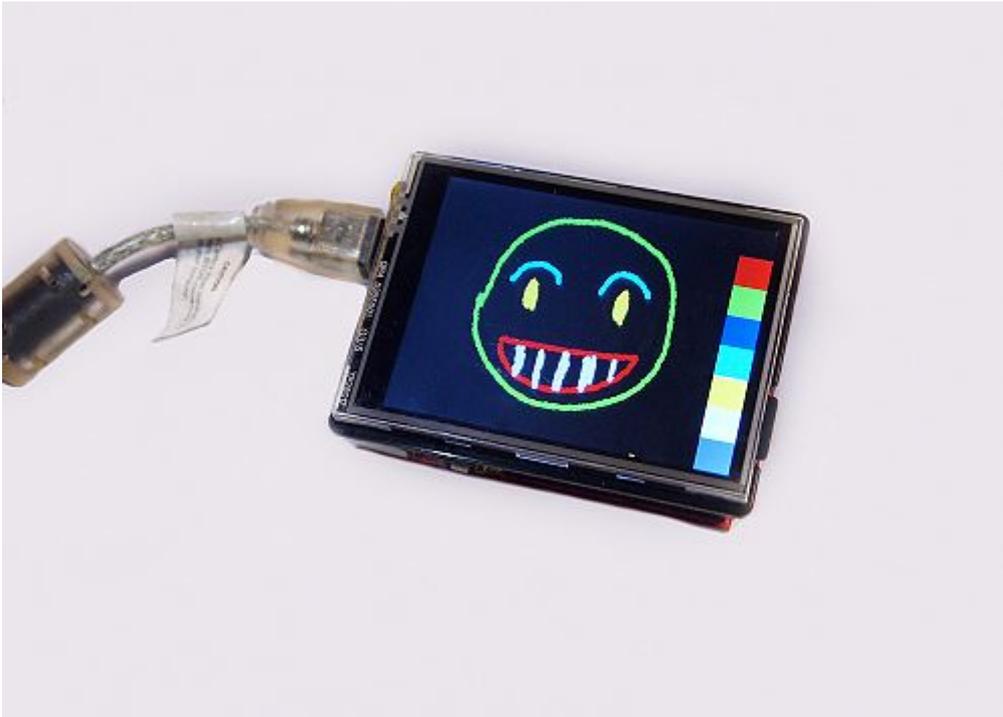
Then enter their names into the Array bmpfiles[][18].

3. Click "Upload". You can see these images you want to see as below.



Demo 3: Draw A Picture

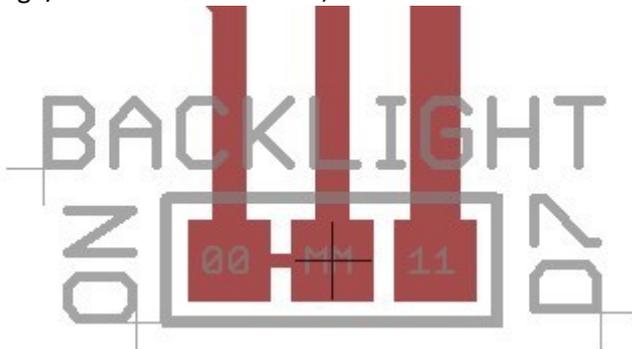
1. Open the "paint" example in the same method as open the "drawCircle" example above.
2. Upload the code. And you will find a color palette on the right of the screen. They are colors available for your painting: BLACK, RED, GREEN, BLUE, CYAN, YELLOW, WHITE, GRAY1.
3. Have a try !



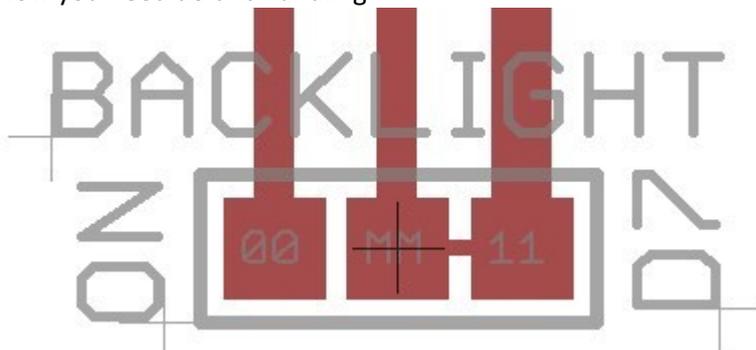
4. If you want to start painting patterns, simply press the reset button or power off and then power-on the Shield.

Backlight Control

In the default state, the backlight of TFT Touch Shield is connected to 5V. In other words, the backlight is on when power for the screen. In fact, you can use a port to control its state by giving the control port high/low level. On the back, it is this:



Now you need do this handling:



Disconnect the connection with ON pin, and connect the backlight pin with D7 pin. Of course, controlling its state is as simple as controlling a LED.

```
#define Backlight 7
void setup(void)
{
  pinMode(Backlight,OUTPUT);
}
void loop(void)
{ digitalWrite(Backlight,HIGH); // turn on the backlight
  delay(500);
  digitalWrite(Backlight,LOW); // turn off the backlight
  delay(500);
}
```