FocusLCDs.com
LCDs MADE SIMPLE®

Ph. 480-503-4295 | NOPP@FocusLCD.com

TFT | CHARACTER | UWVD | FSC | SEGMENT | CUSTOM | REPLACEMENT

Application Note FAN4204

1.8” TFT and TI MSP-EXP432P401R Microcontroller

This application note discusses how to setup a 1.8” TFT display with a TI microcontroller.

1.8" TFT and TI MSP-EXP432P401R Microcontroller

The display used in this project is a 1.8" TFT with 128x160 pixels of resolution. The microcontroller used is the SimpleLink MSP-432P401R from Texas Instruments. The TFT display will be interfaced with the microcontroller via a 4-wire serial connection and programmed using the Energia IDE platform.

In just a few steps the TFT can be wired and programmed to display up to 65K colors and 128x160 pixels of resolution. The display can be powered from the 3.3V output of the TI. Various wiring and interface options are available, from 3-4 wire serial, to 16/18-bit RGB and 8/9/16/18-bit MCU parallel. Additional features of this display are below. As always, check out the data sheet for the specs of this specific display. (datasheet)

- 45 pins, 34.0x47.0 mm
- 128(RGB)x160 pixels, TN
- ILI9163 controller
- White LED Backlight
- Transmissive/Normally White
- Serial and Parallel interfaces
- 3.3V (TYP)

What You'll Need

1. First you will need to download the Energia IDE software if you have not already. This IDE was chosen because of the similarities is has to the Arduino IDE. This is beneficial because there is a large variety of open sourced examples that demonstrate the features of this display. An alternative programming IDE is Code Composer Studios, which is also compatible with the TI microcontroller. The IDE is up to preference. We will come back to this after the hardware is setup.

2. Below is a list of the physical materials you will need to setup the project.

QTY	Description	Note
1	TI MSP-432P401R microcontroller with micro-USB Cable	TI
1	E17RG11216LW6M300-N 1.8" TFT display	FocusLCDs
1	45-pin FPC connector board 0.5 mm pitch	
1	Male-to-Male Jumper Wires	
1	2.2kΩ resistor	
1	55Ω resistor	
1	Soldering Iron	
1	Solder	
1	Solderless breadboard	

Wiring

There are 45 pins on the ribbon cable connected to the display. You will need to connect a 45-pin FPC connector board to convert the ribbon cable to usable pin outs to connect to the TI microcontroller.

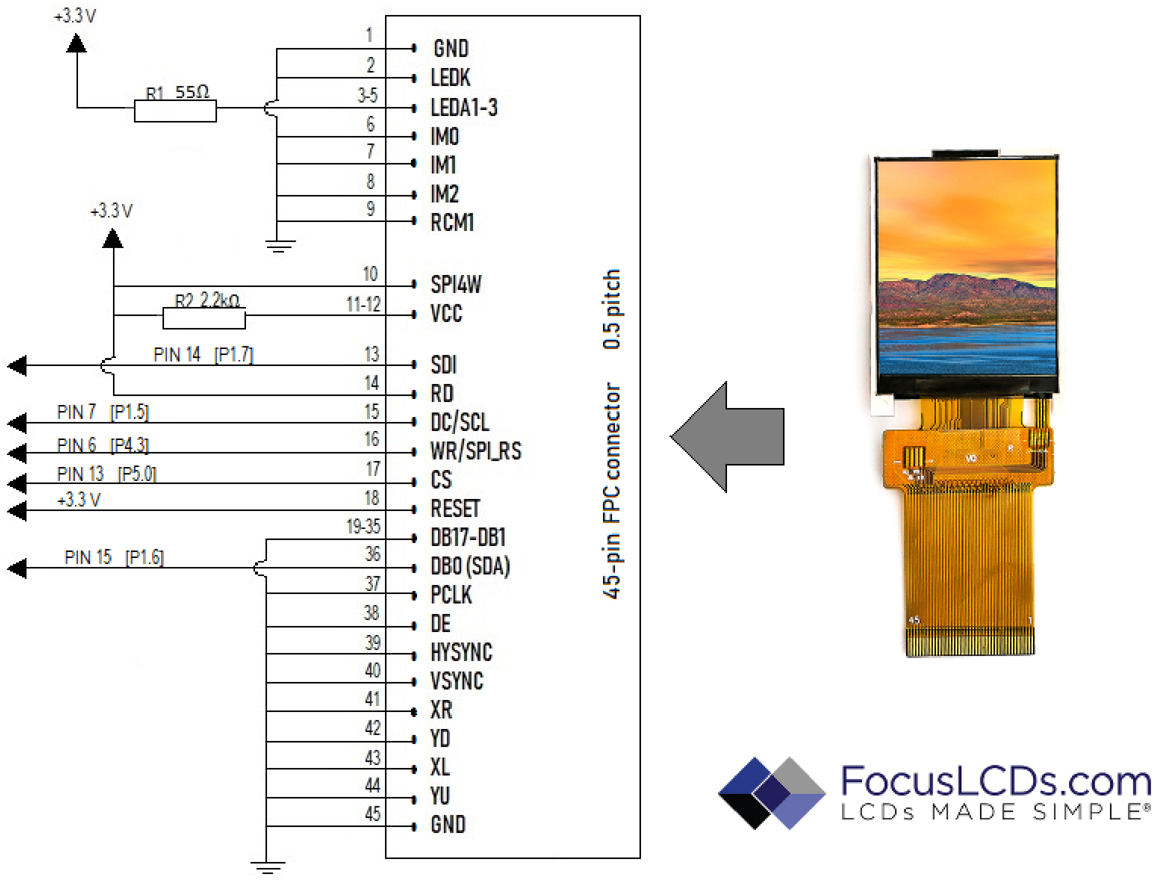


Figure 1: TFT Ribbon to FPC Pin Assignment

There are only a few connections that need to be made between the display for the 4-wire serial interface. The unused parallel data pins will be pinned to GND. Consult the datasheet for a detailed explanation of each pin assignment and their functions. The 4-wire serial data pins are connected to the TI specific serial inputs for the “Hardware SPI” programming option. While any pins can be used, their location must be defined in the “Software SPI” programming option.

Pin Assignments

Pin definitions and connection points are described in the table below. We will use the 4-wire serial interface for this example to save data pins on the TI. A more in-depth description of each of the pins can be found on the datasheet. All unused pins are connected to ground, with the exception of the reset pin. This pin is optional to connect to the TI, if not used it needs to be pulled high.

Pin No.	Pin Name	Description	Connection
1	GND	Ground	Ground
2	LEDK	Backlight cathode	GND
3-5	LEDA	Backlight anode (3.3V, 60mA)	55Ω resistor → 3.3V
6	IM0	Parallel interface selection	GND
7	IM1	Parallel interface selection	GND
8	IM2	Parallel or serial selection, serial = high	3.3V
9	RCM1	RGB or MCU	GND
10	SPI4W	3/4 wire serial selection, 4-wire = high	3.3V
11-12	VCC	Supply voltage (3.3V, 1.5mA)	2.2kΩ resistor → 3.3V
13	SDI	SDI (MISO)	pin 14 (P1.7) on TI MSP432-R
14	RD	Read enable	3.3V
15	DC/SCL	SCL in serial interface (SCLK)	pin 7 (P1.5) on TI MSP432-R
16	WR/RS	RS in serial interface	pin 6 (P4.3) on TI MSP432-R
17	CS	Chip select	pin 13 (P5.0) on TI MSP432-R
18	Reset	Reset pin	3.3V
19-35	DB17-DB1	Data pins for parallel interface	GND
36	DB0	SDA pin for serial interface (MOSI)	pin 15 (P1.6) on TI MSP432-R
37	PCLK	Clock for RGB	GND
38	DE	Data enable	GND
39	HSYNC	HSYNC for RGB	GND
40	VSYNC	VSYNC for RGB	GND
41-44		Not connected	
45	GND	Ground	GND

The TI microcontroller has dedicated serial input pins specific to the board. The pin locations can be seen below and are described in the table for how they are connected to the display. These and other hardware pin definitions can be verified on the TI website.

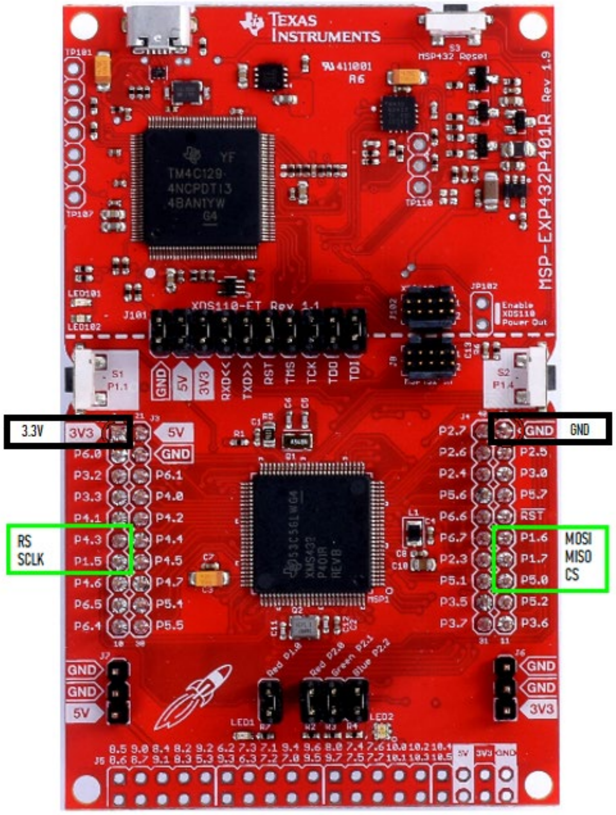


Figure 2: TI MSP-432P401R Serial Hardware Pin Definitions

After the screen is connected, and the TI microcontroller is plugged into the computer you will see the white LED backlight come on. That is a good sign that things are connected correctly.

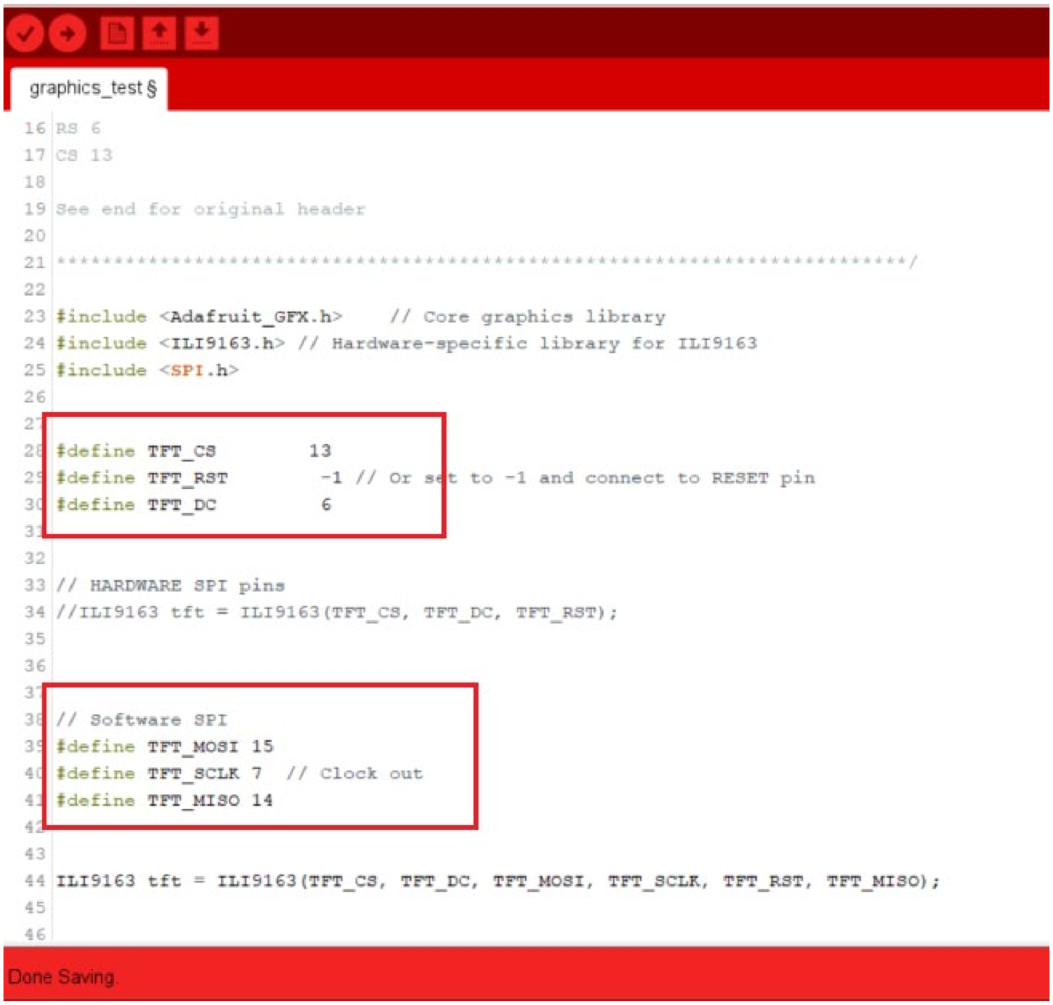
Programming the TI

Now it is time to program the microcontroller. For this example, we will be using the Energia IDE. An alternative platform that can be used is Code Composer Studios. I have found that Energia IDE is more practical for this example because of the compatibility with Arduino specific code with only minor changes.

We will need a program for the specific display chip “ILI9163”. For this example, I have created a modified version of a popular Adafruit example that demonstrates the various capabilities of the display. The example can be downloaded from Github here.

You will also need to include Adafruit’s GFX library which can be downloaded here. This is a popular library that contains examples and features for TFT displays that prove to be useful for this application.

Once these libraries are downloaded, add them to the Energia documents file. You should see the example as below:



```
RS 6
CS 13

See end for original header

*************************************************************************/

#include <Adafruit_GFX.h>    // Core graphics library
#include <ILI9163.h> // Hardware-specific library for ILI9163
#include <SPI.h>


#define TFT_CS        13
#define TFT_RST        -1 // Or set to -1 and connect to RESET pin
#define TFT_DC         6


// HARDWARE SPI pins
//ILI9163 tft = ILI9163(TFT_CS, TFT_DC, TFT_RST);



// Software SPI
#define TFT_MOSI 15
#define TFT_SCLK 7  // Clock out
#define TFT_MISO 14


ILI9163 tft = ILI9163(TFT_CS, TFT_DC, TFT_MOSI, TFT_SCLK, TFT_RST, TFT_MISO);


```

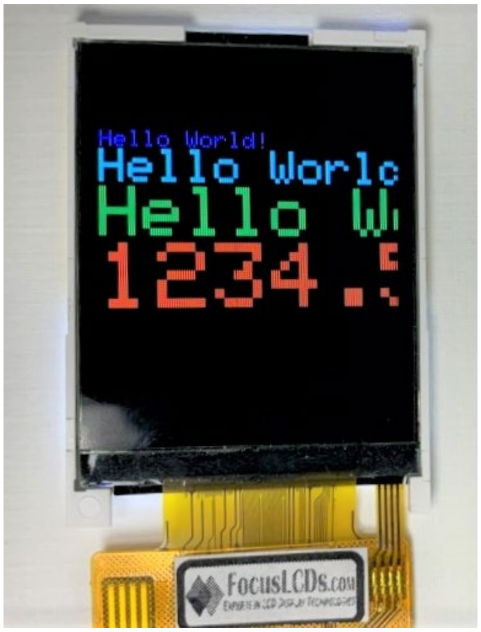
Done Saving.

Hardware or software definitions are available. The wiring is set up so that the 4-wire SPI pins are at the correct location for the hardware definition. The hardware option is much faster than the software and is the recommended option.

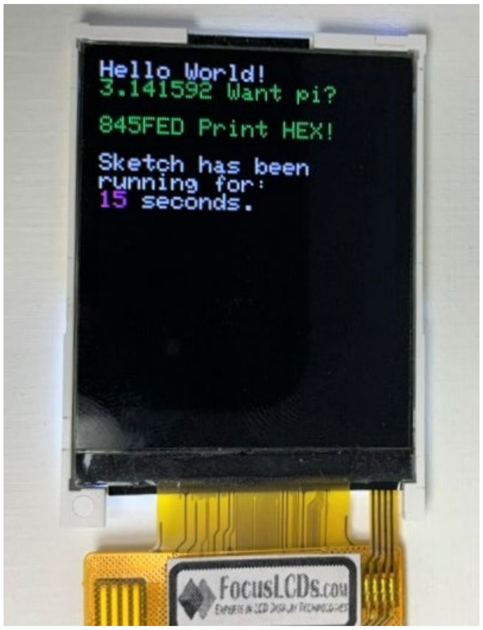
The example in the image above is using the software declaration of the data pins connected to the TI board. Verify that these values are correct as shown above.

You can now test any of these various programs that might be good reference for your specific project. There a lot of good examples in the Adafruit GFX library as a resource to test the various capabilities of the display.

The example is a good first program to run just to see if everything is connected correctly and to test the graphics. Some pictures of the example are shown below.



```
Hello World!
Hello Worlc
Hello W
1234.
```



```
Hello World!
3.141592 Want pi?

845FED Print HEX!

Sketch has been
running for:
15 seconds.
```

Summary

This 1.8" TFT is a good option for displaying 16-bit 65K color images. This is compatible with most microcontrollers as it saves on-board memory. This is beneficial for storing bitmaps on flash memory since the screen is small and the 65K color bitmap image won't take up all the on-board storage on the TI. This display also has a version with a resistive touch screen. This would be a good option for a digital push button. This may be further discussed in a future application note.

DISCLAIMER

Buyers and others who are developing systems that incorporate FocusLCDs products (collectively, “Designers”) understand and agree that Designers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Designers have full and exclusive responsibility to assure the safety of Designers' applications and compliance of their applications (and of all FocusLCDs products used in or for Designers’ applications) with all applicable regulations, laws and other applicable requirements.

Designer represents that, with respect to their applications, Designer has all the necessary expertise to create and implement safeguards that:

(1) anticipate dangerous consequences of failures

(2) monitor failures and their consequences, and

(3) lessen the likelihood of failures that might cause harm and take appropriate actions.

Designer agrees that prior to using or distributing any applications that include FocusLCDs products, Designer will thoroughly test such applications and the functionality of such FocusLCDs products as used in such applications.