Arduino TFT display and font library

by **Bodmer** on January 5, 2015

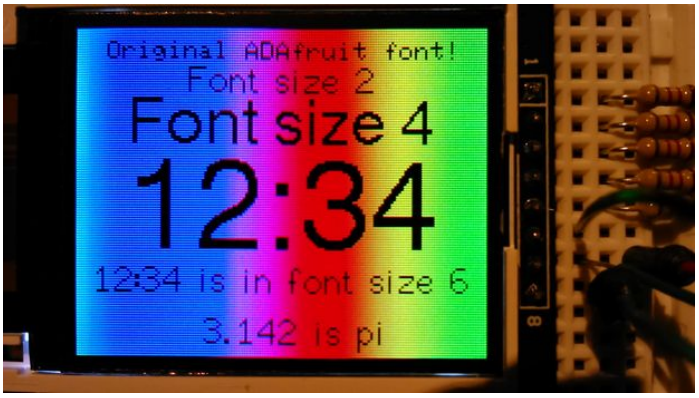
Table of Contents

Intro: Arduino TFT display and font library

I am quite a fan of the Arduino as there is so much software already available on the internet, this makes completing new projects easier!

Recently I bought a few cheap 1.8" TFT displays for a project, initially I could not get them working which was frustrating. Eventually I found that the ones I had bought used the Samsung S6D02A1 driver chip and I had been trying to use the wrong library.

After a brief search on the web I came across the Adafruit_QDTech library which worked really well. This library is used in conjunction with the Adafruit_GFX library. After trying some examples I was a little disappointed that only scaled blocky fonts were available - so I have modified the original libraries to add some new fonts and functions.



Step 1: Wiring it up

I used the Arduino UNO and linked it to the display as follows:

- UNO +5V to display pin 6 (VCC) and pin 7 (BL)
- UNO 0V (GND) to display pin 8 (GND)
- UNO digital pin 7 through a 1K2 resistor to display pin 1 (RST)
- UNO digital pin 8 through a 1K2 resistor to display pin 3 (D/C)
- UNO digital pin 9 through a 1K2 resistor to display pin 2 (CS)
- UNO digital pin 11 through a 1K2 resistor to display pin 4 (DIN)
- UNO digital pin 13 through a 1K2 resistor to display pin 5 (CLK)

The 1K2 resistors are required to protect the display being damaged by the 5V logic levels from the UNO, these limit the current flow. Ideally we could use a level shifter but the resistors work fine.

Step 2: Loading the libraries

I have modified the Adafruit libraries and added the fonts. The fonts will be stored in FLASH so will use up some program space however a reasonable amount of space is still available, if you run out then use an Arduino Mega board.

I am actually a hardware designer so the software updates could probably be improved, but they are working!

The zip file contains the two libraries you will need. These have have new names (_AS appended) so they will coexist with any libraries you already have. There are 3 example programs included in the Adafruit_QDTech_AS library.

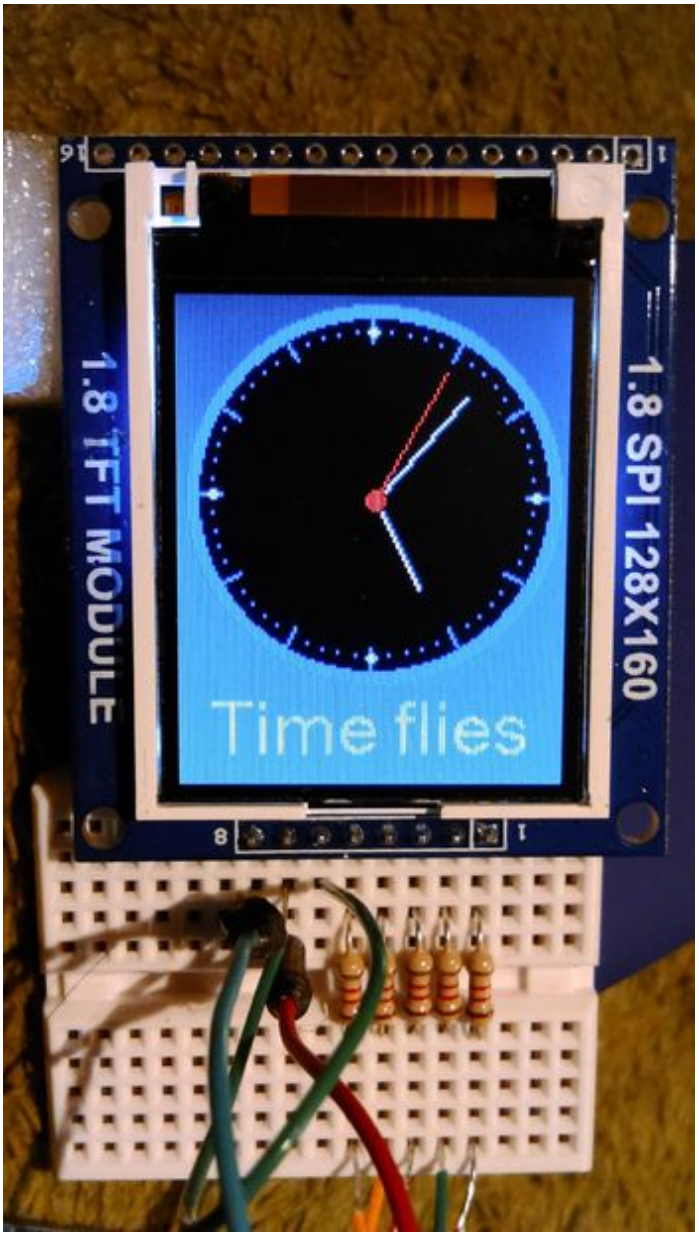
File Downloads

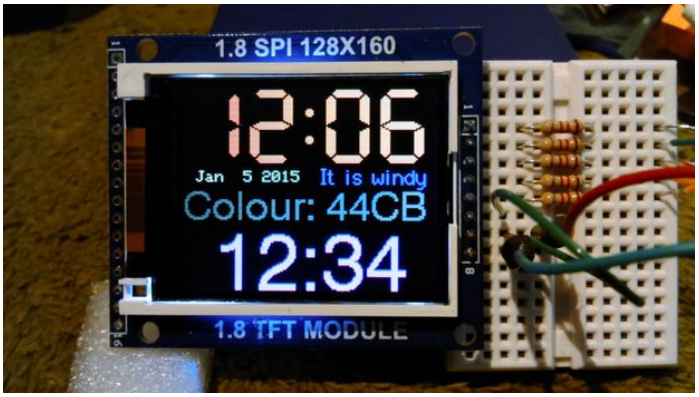
Libraries.zip (34 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Libraries.zip']

Step 3: Example programs

The example programs are:

- TFT_Rainbow - gives some examples of drawing the fonts on the display
- TFT_Clock - an analogue clock drawn with the standard Adafruit graphics routines plus a centred line of text in Font 4
- TFT_Clock_Digital - a digital clock using the 7 segment display font and other font drawing examples.





Step 4: Library functions and fonts

Here are the library functions that can be called:

- int drawUnicode(unsigned int uniCode, int x, int y, int size);
- int drawNumber(long long_num,int poX, int poY, int size);
- int drawChar(char c, int x, int y, int size);
- int drawString(char *string, int poX, int poY, int size);
- int drawCentreString(char *string, int dX, int poY, int size);
- int drawRightString(char *string, int dX, int poY, int size);
- int drawFloat(float floatNumber,int decimal,int poX, int poY, int size);

In summary, the X and Y parameters are the coordinates for the drawing.

Each function returns the X position delta to the end of the printed characters.

"size" is the font size:

- Only font numbers 2,4,6,7 are valid
- Font 6 only contains characters [space] 0 1 2 3 4 5 6 7 8 9 0 : a p m
- Font 7 is a 7 segment font and only contains characters [space] 0 1 2 3 4 5 6 7 8 9 0 : .

The fonts are proportionally spaced to give an improved appearance.

The routine for floating point numbers uses a parameter "decimal" which defines the number of decimal places to show, this aids formatting and displaying sensor readings, the number is rounded, so for example if decimal = 3, then 3.14159 will display as 3.142 and 3.14 will display as 3.140

drawCentreString will centre the string around the x position. convenient for tidy display formatting

drawRightString will draw the string right justified to the x position, useful for labels that are followed by sensor readings printed afterwards.

Step 5: Have fun!

I hope you find this Instructable useful!

P.S. I am am not connected with Adafruit, but do check out their great products and look for the support on using the standard graphics library functions (these remain within the adapted libraries).

Related Instructables

Mini Arduino environment monitor ** UPDATE - Added RTC *** by johnag

Cheap TFT 2.2 inch Display on Arduino (ILI9340C or ILI9341) by tomk11

Make a Mini Arduino programmable 4 channel DC-DVM by johnag

Mini Arduino Portable EEG - Brain Wave Monitor + by johnag

Scrolling text with Arduino and Adafruit TFT shield by turnturtle

Mini Arduino Multi Function Controller w/ LCD Display by johnag

Comments