

Using LCD Displays with Arduino 1.6.3

Pete Juliano—N6QW

radioguy90@hotmail.com

Many new technologies are now available to the homebrew constructor that only a few short years ago were on the “sure would be nice to have list.” They are now here and inexpensive. Chief among these is the Arduino Microcontroller Board, the AD9850 DDS boards and now the Si5351 PLL that can simultaneously generate three different clock frequencies. Hams across the globe have figured out how to make one of those three clock frequencies in the Si5351 into a Variable Frequency Oscillator.

Roughly speaking one can now have a digital VFO and BFO generator with some form of digital display for around \$20. Initially the displays were LCD, ranging all the way from 8x2 (8 digits, 2 lines) to 20x4s. So lots of information can be displayed. Then there was the move to the TFTs such as the black and white Nokia 5110 which is a 48x84, to the color displays such as the 128x128 to the 240x320. The latter has 256K colors. The 128x128 can be purchased from China for \$5.50 USD shipped to your door!

But like most things, there is a bit of a fly in this ointment. The problem is that there is a great deal of variation in displays, particularly LCDs and the software which controls them. Ruefully, I found that not all displays are equal and there are many versions and variants of Liquid Crystal libraries for the Arduino. Moreover, using the parallel displays chews up a lot of precious Arduino Pins. But, many of the Arduino boards and their clones have the I2C pins, which means that with an appropriate convertor only two pins from the Arduino are required, typically A4 (SDA) and A5 (SCL). Those Arduino pins, plus supplying 5 VDC and Ground, allow the LCD display to be included in the project.

This convertor, which is frequently called the “I2C Backpack,” is a small circuit board that has four pins on one end that act as the interface to the Arduino and sixteen pins along the side that can be soldered to the LCD or plugged into a 16 pin socket fitted to the LCD (which is the method I prefer). The board also requires a library that is not included in the Arduino IDE to be made a part of any Arduino sketch.

Variations in later versions of the Arduino IDE also make some differences. When I was working on these projects I used an older version of Arduino, namely v1.0.5. It works well and so I found no need to update it. I guess I believe in—if it works, don’t fix it. So, when I opened the Arduino environment I always got a message on my screen advising me that there was a later version of the IDE and asking if I wanted to download it. I always summarily said NO! However, other users have upgraded to v. 1.6.3 before attempting my projects. When trying to use some of my sketches, they would not compile using the later environment nor would they upload to the Arduino microprocessor. And yes, one of the huge error messages was about the displays. Moreover, one of the experts I consulted with made an excellent point about using the latest IDE. Basically the Arduino is an open source development tool and as more use is made of the software, problems/bugs that are found are fixed with the later releases. This is important when new hardware is being fitted to an Arduino as the new hardware is relying on the later IDEs.

Thus, I felt compelled to test drive the v.1.6.3 version even though the newer IDE was causing problems with my sketches. In

the Soldersmoke Podcast 175 (April 18, 2015) where I participated alongside Bill Meara, N2CQR, we discussed this issue and asked if anyone knew the why of the problem. Mark VanderWetting, K6HX stepped into the picture and did a lot of research and development to find the answer. We are deeply indebted to Mark!

Learning long ago about test driving new software, I used a separate computer for this version. Again I want to emphasize that if you have two computers keep the new IDE (V1.6.3) separated from what is working until such time that you are comfortable with the latest version. I should report it maybe as long as six-months, before I will make the full conversion. “Don’t burn your bridges” is sage advice!

Let us start with the basics. Mark, K6HX has documented his sojourn in this blogpost which can be found here. <http://brainwagon.org/2015/04/22/using-a-sainsmart-lcd-panel-with-the-arduino-1-6-3-ide/> . We thank Mark for his excellent work in defining and resolving the problem! The new Liquid Crystal Libraries compatible with V 1.6.3 can be found here:

<https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads>

I used the latest version and downloaded that to my computer and in unzipping the files renamed the folder LiquidCrystal_I2C. Also note that in the unzipping process I got warnings about some encrypted files that were not downloaded—that does not seem to be a problem.

Just having the new library for LiquidCrystal_I2C is not enough as the actual sketch code must be modified with the following four entries.

```
#include "LiquidCrystal_I2C.h"
#define BACKLIGHT_PIN (3)
#define LED_ADDR (0x27) // might need to be 0x3F, if 0x27
                        // doesn't work
LiquidCrystal_I2C lcd(LED_ADDR, 2, 1, 0, 4, 5, 6, 7, BACK-
LIGHT_PIN, POSITIVE);
```

Again, note that for the Sain Smart, the address is 0x3F and most of the generic ones take 0x27.

BUT you are not DONE!!! For sketches done in other versions of the IDE it does not compile if you simply open the earlier version with 1.6.3. You must open the old sketch then using the Edit Tab-Select All and Copy. Next open a blank V 1.6.3 Sketch and paste the information into the blank Sketch. Note in a 1.6.3 blank sketch first delete the setup and void wording so you have a clean blank sketch and THEN paste the information. Now save that sketch with a new name.

Also, all of the libraries specified in the sketch must be in the Arduino Library file folder such as LiquidCrystal_I2C, Rotary, si5351 or Jsi5351. If you are using the TFT displays then you need the Adafruit Graphics Libraries [Adafruit_GFX, Adafruit_PCD8544] as well as the libraries for the displays such as TFT_ILI9163C and TFT_ILI9340. ●●