Chapter 11 – Using an Alphanumeric Display

April, 2020
David Ackmann – Gateway Division NMRA

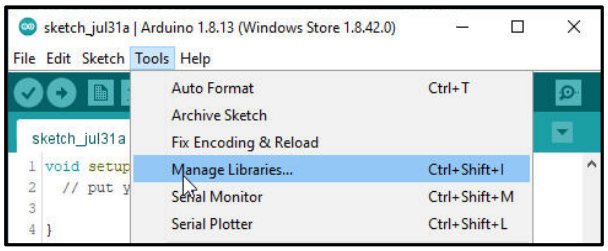
Why would you want to add a Liquid Crystal Display to a piece of model railroad animation? Well, you probably have many different static signs on your layout already, things like city names, posters, billboards and the like. A computer controlled alphanumeric display can provide all sorts of dynamic information to operators and visitors. I use displays on over half of my animation projects. On the Speedometer project it provides the elapsed time to complete a loop and then displays the scale miles per hour every time a train makes a round. If some animation is not continuously running, like the Lumberjack project, a display can encourage a visitor to push a button to start the action. Just think of all the information found on exhibits at a museum; displays can provide all kinds of information and encourage interaction..

I like to use 4 line displays with 20 characters per line, and I use one from Sunfounder, available from AMAZON (ASIN: B01GPUMP9C, $12.99). There are less expensive units without the I2C "piggyback" chip, but these are more difficult to interface with Arduinos, and I will not use them. There are also displays with only 2 lines and 16 characters per line, and they are $4.00 less expensive than the larger displays, but I don't use them either because I find I often want to display more information than they can handle. But it's your choice.

So how do we connect them to an Arduino? It takes only 4 wires. I like to run a black wire from Ground to the GND pin on the display; the display has pins on its connectors, so male-female jumpers tend to work well for these connections. I run a red wire from the 5V position on the Arduino to the "VCC" pin on the display. Finally, I run a yellow wire from Arduino position A4 to the Display's SDA pin, and a blue wire from the Arduino's A5 position to the display's SCL pin. So make your connections and power up your Arduino.

Did you "see the light"? If not, perhaps you need to adjust the contrast. On the back of the display, on the "piggyback board", there is a small blue potentiometer with a screw adjustment. Turn it a bit clockwise to make it brighter; you won't see any characters in the display just yet, but you will see some brighter rectangles, indicating that the display has power. But don't turn it up too bright, as all you will see then are the rectangles. And too bright may burn character images into the display.

Although I have never used more than one display on a project, it can be done because displays can have different addresses, and before you can use yours you need to know its address. So connect those four display wires to your Arduino, visit https://daackm.github.io , scroll down to Chapter 11 and download and execute the "SCANNER" program. When you run this program and open the Serial Monitor, the SCANNER program will tell you the address of your display. Displays generally refer to their addresses in hexadecimal, and chances are your display will be "0x27"; if not, then it is probably "0x3F". Why these two? I don't know. But I do know that once I discover the address of a board, I write the address on the board with a Sharpie so I don't forget.



But before you can run a real program and display information, the IDE must have access to a display library. So, open the IDE and click on the "Tools" tab, then on the "Manage Libraries" sub-tab. The "Library Manager" window will appear. Scroll down until you see a library named "LiquidCrystal I2C" and click on the "Install". Note that there are several libraries that begin with "LiquidCrystal", and you want the one with "I2C" in the name.

Now for the software. The is a small test program under the Chapter 11 heading with will display a counter on the Display, so click on its entry, download and execute the test code. If nothing displays, check the address with the SCANNER program, check the wiring, and check the potentiometer on the back of the Display.

Have fun!