Interfacing to an LCD Screen Using an Arduino

Matt Clary

4/3/2015

Abstract

A liquid crystal display, or LCD, is a video display that utilizes the light modulating properties of liquid crystals to display pictures or text on a screen. Since their invention in 1964, LCD screens have grown to be used in a very wide variety of applications, including computer monitors, televisions, and instrument panels. One way to utilize an LCD is with an Arduino microcontroller. By wiring an Arduino microcontroller to the pins of an LCD display it is possible to program the microcontroller to display a desired text string or image on the screen.

Table of Contents

Introduction

To fully understand how to interface an Arduino microcontroller to an LCD screen, it is important to understand how an LCD functions. An LCD display is composed of pixels made up of liquid crystals. Liquid crystals exist in a state that's between a solid and a liquid. At any time liquid crystals can be in a variety of phases, most notably the nematic phase or the smectic phase. In the nematic phase the crystals act more like a liquid, allowing the molecules of the crystals to rearrange themselves while remaining oriented in a uniform direction. In the smectic phase, the molecules can form into layers that can move past one another relatively easily. Molecules of a certain layer can move freely within that layer, but cannot move to adjacent layers.

When the liquid crystals exist in the nematic phase they tend to adapt a twisted up structure, which can be straightened out when electricity is applied to them. In an LCD, a liquid crystal pixel lies between two glass filters, one behind it and another in front at 90 degrees. When electricity is applied to the liquid crystals the twisted structures that have naturally formed are straightened out. This rotates any light passing through 90 degrees, thus allowing it to pass through the two glass filters.

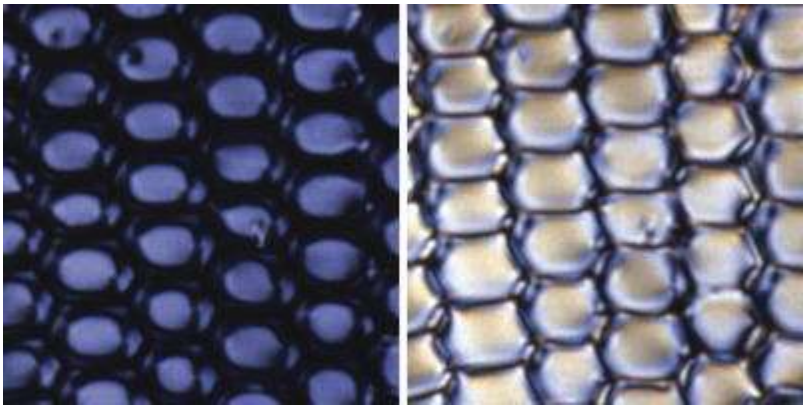


Figure 1: Liquid crystals in their “on” (Right) and “off” (Left) states

In order to give some context on how it is possible to program an LCD screen to display certain information the Arduino microcontroller should be briefly discussed. The Arduino model that is the topic of this application note is the Arduino Uno. The Arduino Uno board consists of ATmega328 microcontroller, 14 digital input/output pins, 6 analog inputs, a 16 MHz resonator, a USB connection, a power jack, and an ICSP header. For this project the most important features of the board are the microcontroller, which allows for the building of programs in the

JAVA programming language, and the digital input/output pins, which allows for the transfer of information between the board and the LCD display.

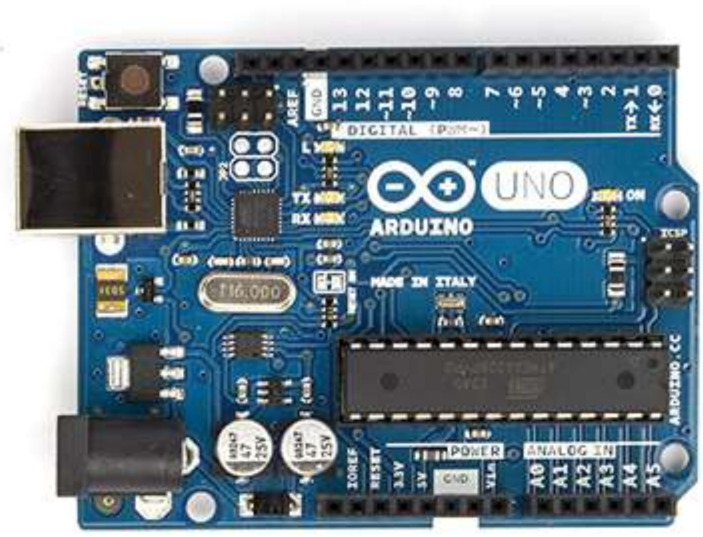


Figure 2: An Arduino Uno Board

Objective

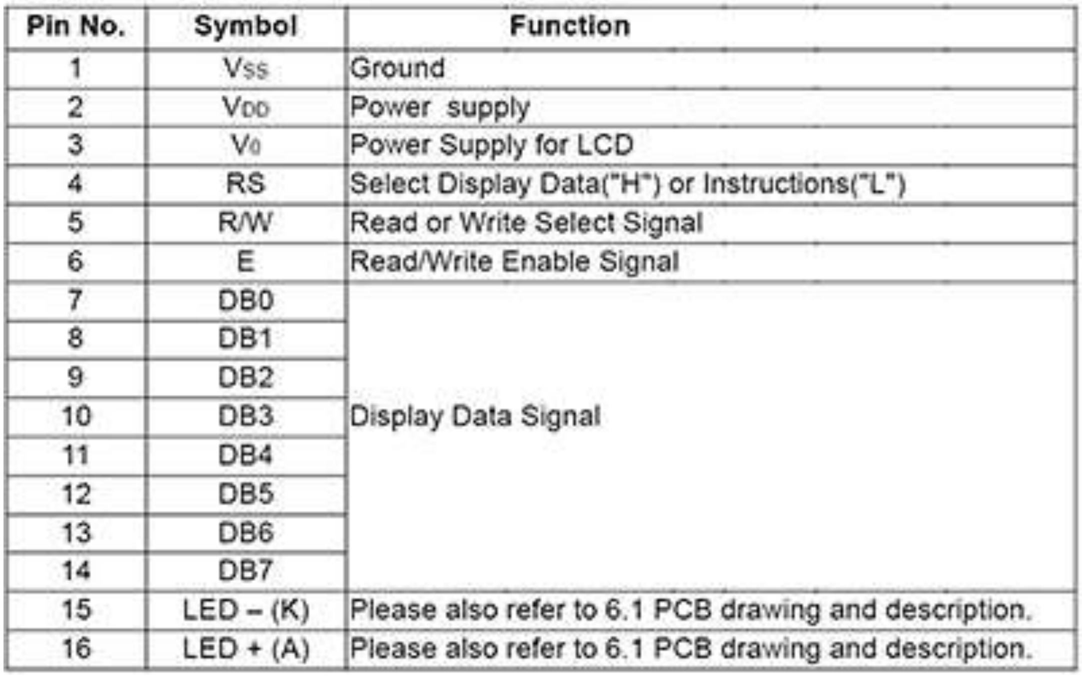
The object of this application note is to demonstrate how an Arduino microcontroller can be used to interface with an LCD screen.

Method

Interfacing an Arduino microcontroller with an LCD display consists of two parts, wiring and programming.

Wiring

A typical LCD display consists of 16 pins that control various features of the screen. A table that shows the pins and describes each function can be seen in Table 1 below. The Arduino microcontroller can output voltages of either 5 V or 3.3 V, so the LCD can be powered by wiring V_{SS} and V_{DD} to the ground and 5 V pins on the microcontroller. It is possible to adjust the contrast of the screen by wiring a variable resistor to V_0 located at pin 3 on the screen. The RS, R/W, and E pins are wired to pins 12, ground, and 11 respectively on the Arduino. The LCD screen can operate in both 8-but mode and 4-bit. For this application note only 4-bit mode will be discussed, as it requires fewer pins and is generally easier to use.



Pin No.	Symbol	Function
1	V_{SS}	Ground
2	V_{DD}	Power supply
3	V_0	Power Supply for LCD
4	RS	Select Display Data("H") or Instructions("L")
5	R/W	Read or Write Select Signal
6	E	Read/Write Enable Signal
7	DB0	Display Data Signal
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	LED – (K)	Please also refer to 6.1 PCB drawing and description.
16	LED + (A)	Please also refer to 6.1 PCB drawing and description.

Table 1: Pinout of a typical LCD screen

To interface with the LCD in 4-bit mode the Arduino only needs to be connected to pins DB4-DB7, which will connected to digital output pins 5-2 respectively. Pins 15 and 16 on the LCD screen are used to power a backlight in the screen. This makes text displayed in the screen easier to read in poorly lit environments and is optional. In order to power the backlight pin 15 should be connected to ground while pin 16 should be connected to the 5 V output of the Arduino. To power the Arduino a 9 V battery can be connected to the V_{In} and ground pins on the Arduino. If such power source is available the Arduino can be powered by using its USB connection with a computer. Figure 3 below shows what the final wiring scheme should look like after all connections are made.

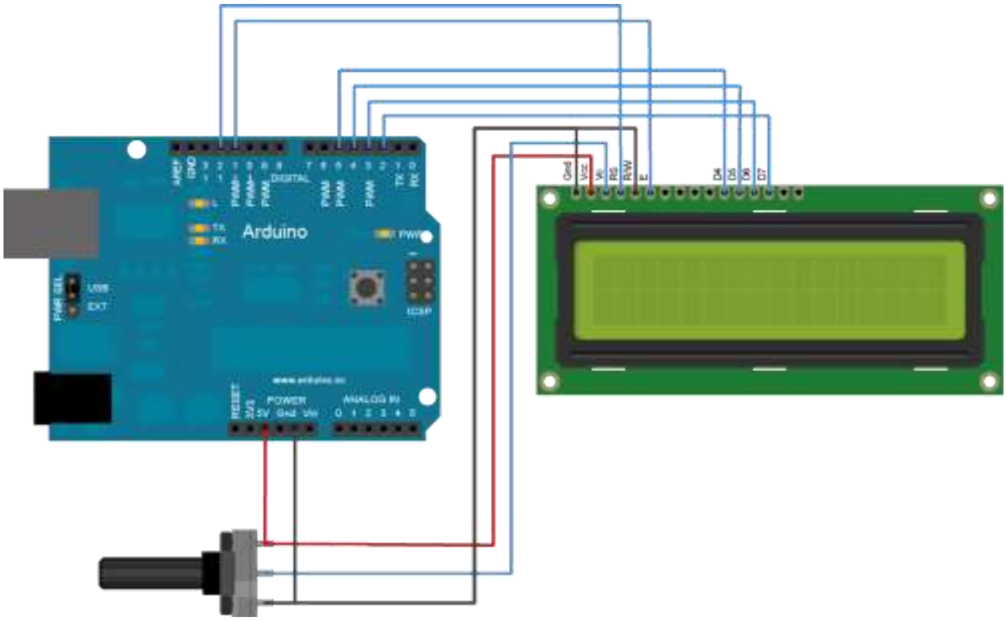
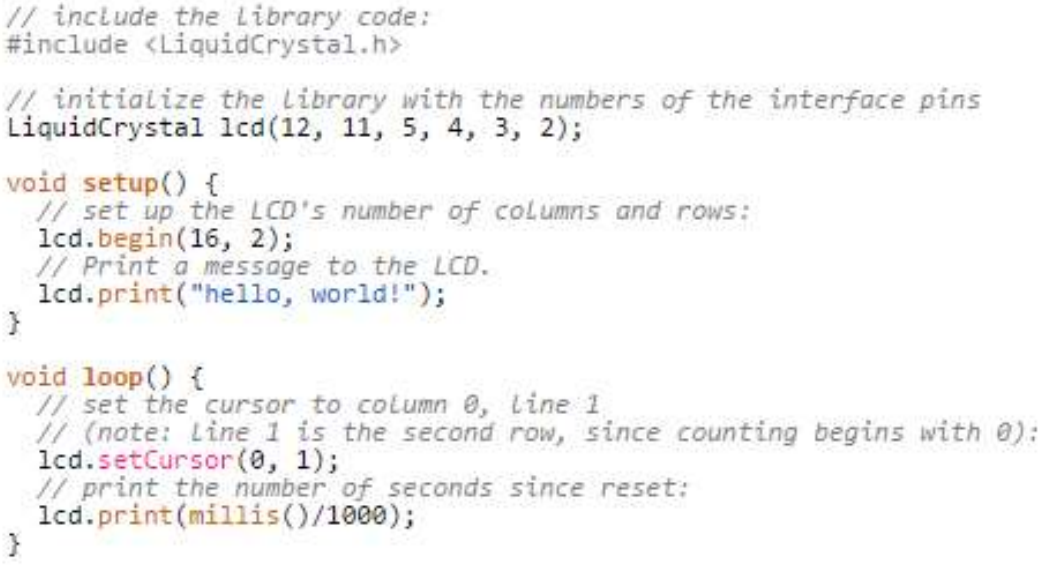


Figure 3: Wiring schematic for LCD-Arduino interfacing

Programming

Once all of the wiring us complete the Arduino can write text to the LCD via programs that are loaded onto the microcontroller. Programs utilizing the JAVA programming language can be uploaded to the microcontroller via free open source software made by Arduino. A library made by Arduino called "LiquidCrystal" can downloaded and used in any programs written to interface with the LCD display. The "LiquidCrystal" library contains a variety of built in functions that allow an Arduino to control an LCD. These functions allow the user to do things like print text strings to the screen, initialize the screen cursor to a certain pixel, scroll the screen left or right, clear the display, as well as a variety of other things. Below in Figure 4 is an example of some code to write a text string to an LCD screen.



```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis()/1000);
}
```

Figure 4: Code to write to an LCD

As you can see, the first part of the code includes the “LiquidCrystal” library that includes the LCD interface functions. It then defines pins 12, 11, 5, 4, 3, and 2 of the Arduino as the pins that are interfacing with the LCD screen. Remember, pins 2-5 control the display data signal pins, while pins 12 and 11 control RS and Enable pins respectively. The program then defines the size of the LCD connected (In this case a 16x2 display) with the lcd.begin function. With the lcd.print function, the text string “Hello World” is then printed to the screen. As you can see, the cursor that determines where the text begins can be set with the lcd.setCursor function. The Arduino has a built in function that can count the number of milliseconds since the program began to run. With the final lcd.print function, the Arduino is displaying every second that passes on the screen. An image of what the LCD screen looks like when this program is run can be seen below in Figure 5.

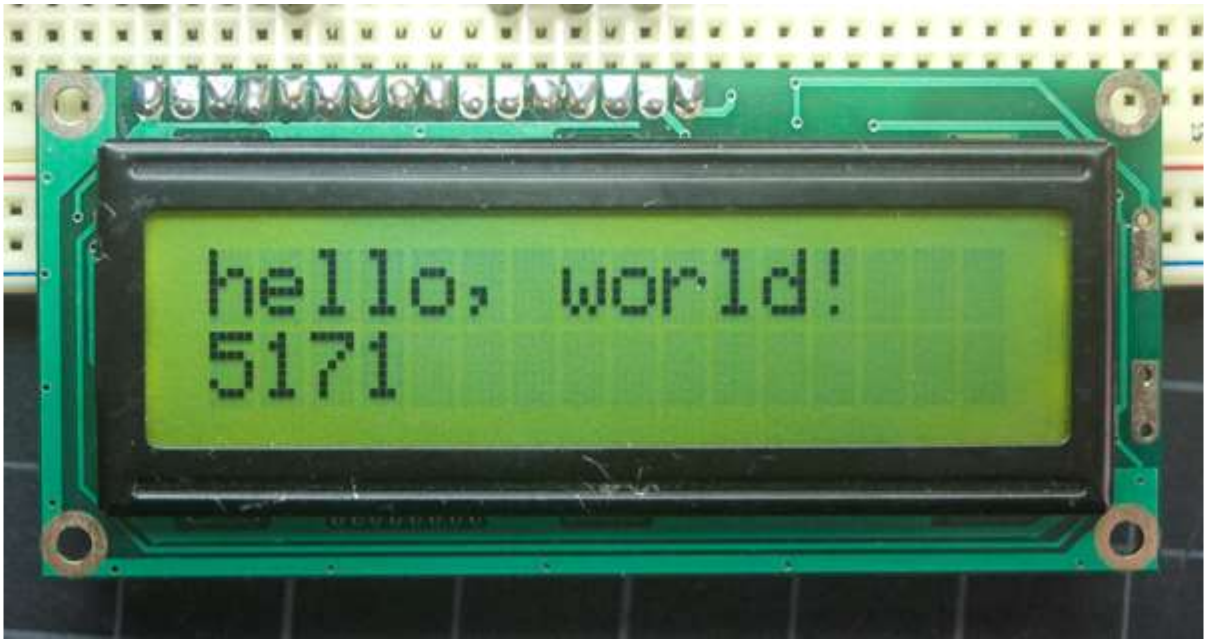


Figure 5: An LCD screen displaying text from example code

Conclusion

One of the main reasons that LCD displays have become so common in recent years is due to the ease at which they can be interfaced with. Although an Arduino microcontroller was used to demonstrate this ability in this application note, a variety of other microcontrollers can be used to achieve the same result. On a similar note, much more can be accomplished with an Arduino microcontroller and an LCD display than what was demonstrated in this application note. By adding further hardware to the circuit described here things like real time clocks and thermometers can be added, allowing the LCD to display the current time and temperature. The simplicity of their design and ease of interface allow for LCD screens to be used in a variety of projects on both a large and small scale.

References

http://en.wikipedia.org/wiki/Liquid-crystal_display

http://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems

http://www.hacktronics.com/Tutorials/arduino-character-lcd-tutorial.html

http://www.explainthatstuff.com/lcdtv.html

http://arduino.cc/en/Tutorial/LiquidCrystal