Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Second term- Lecture- 1: Liquid crystal display (LCD)

Liquid Crystal Display (LCD) Interface with Arduino

Liquid crystal displays (LCDs) offer a convenient and inexpensive way to provide a user interface for a project. This lecture explains how to connect and use common text and graphical LCD panels with Arduino. By far the most popular LCD is the text panel based on the Hitachi HD44780 chip. This displays two or four lines of text, with 16 or 20 characters per line (32- and 40-character versions are available, but usually at much higher prices).

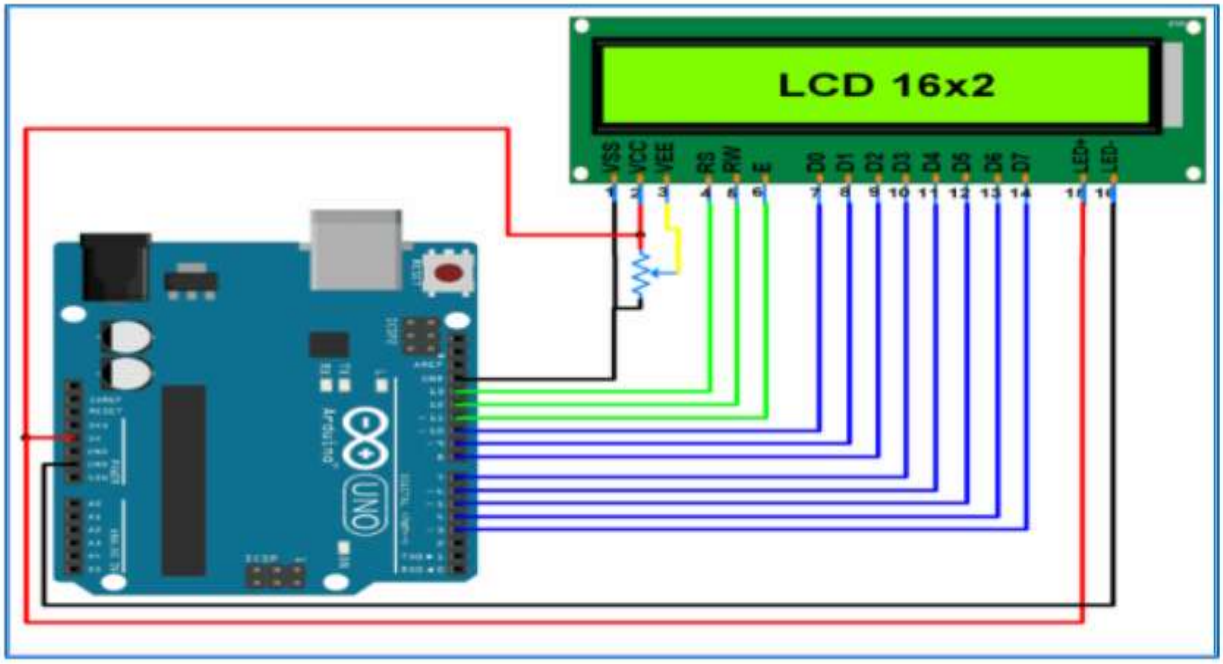
A library for driving text LCD displays is provided with Arduino, and you can print text on your LCD as easily as on the Serial Monitor because LCD and serial share the same underlying print functions. LCDs can do more than display simple text: **words** can be scrolled or highlighted and you can display a selection of **special symbols** and **non-English characters**. You can create your own symbols and block graphics with a text LCD, but if you want fine graphical detail, you need a graphical display. Graphical LCD (GLCD) displays are available at a small price premium over text displays, and many popular GLCD panels can display up to eight lines of 20 text characters in addition to graphics. LCD displays have more wires connecting to Arduino than most other things.

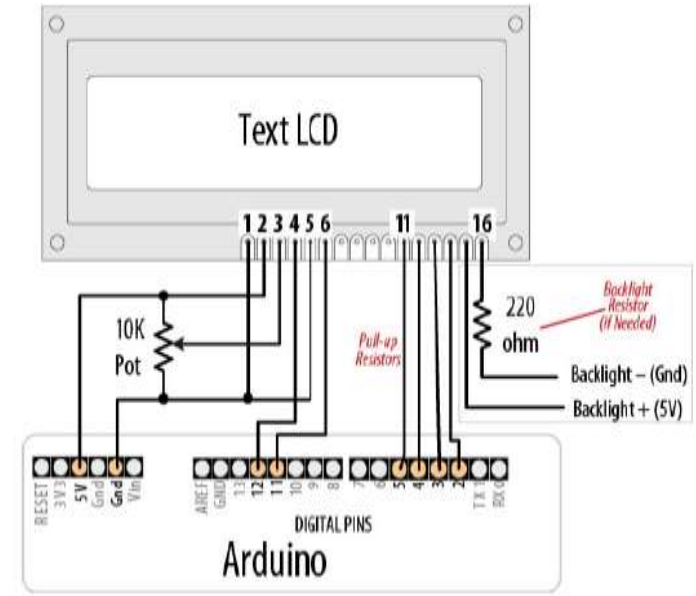
Incorrect connections are the major cause of problems with LCDs, so take your time wiring things up and triple-check that things are connected correctly. An inexpensive multimeter capable of measuring voltage and resistance is a big help for verifying that your wiring is correct. It can save you a lot of head scratching if nothing is being displayed. You don't need anything fancy, as even the cheapest multimeter will help you verify that the correct pins are connected and that the voltages are correct.

The Arduino library allows an Arduino board to control LCDs based on the Hitachi HD44780 (or a compatible) chipset, which is found on most textbased LCDs. The library works with in either 4- or 8-bit mode (i.e. using 4 or 8 data lines in addition to the rs, enable, and, optionally, the rw control lines. Figure below shows the most common LCD connections. It's important to check the data sheet for your LCD to verify the pin connections. Table below shows the most common pin connections, but if your LCD uses different pins, make sure it is compatible with the Hitachi HD44780—this project will only work on LCD displays that are compatible with that chip. The

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Second term- Lecture- 1: Liquid crystal display (LCD)

LCD will have 16 pins (or 14 pins if there is no backlight)—make sure you identify pin 1 on your panel; it may be in a different position than shown in the figure.





Connections for a text LCD

LCD pin	Function	Arduino pin
1	Gnd or 0V or Vss	Gnd
2	+5V or Vdd	5V
3	Vo or contrast	
4	RS	12
5	R/W	
6	E	11
7	D0	
8	D1	
9	D2	
10	D3	
11	D4	5
12	D5	4
13	D6	3
14	D7	2
15	A or analog	
16	K or cathode	

Email: zahraa.hashim@uomus.edu.iq

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Second term- Lecture- 1: Liquid crystal display (LCD)

Connecting and Using a Text LCD Display

Problem

You have a text LCD based on the industry-standard HD44780 or a compatible controller chip, and you want to display text and numeric values.

Solution

The Arduino software includes the LiquidCrystal library for driving LCD displays based on the HD44780 chip.

```
// 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 Students");
}
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);
```

Al-Mustaqbal University
Department of Medical Instrumentation Techniques Engineering
Class: four
Subject: Advanced logic design
Lecturer: Dr. Zahraa hashim kareem
Second term- Lecture- 1: Liquid crystal display (LCD)

Formatting Text

Problem

You want to control the position of text displayed on the LCD screen; for example, to display values in specific positions.

Solution

This sketch displays a countdown from 9 to 0. It then repeat the sequence and display on the LCD:

```
/*
LiquidCrystal Library - FormatText
*/
#include <LiquidCrystal.h> // include the library code:
//constants for the number of rows and columns in the LCD
const int numRows = 2;
const int numCols = 16;
// 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);
}
void loop()
{
lcd.begin(numCols, numRows);
lcd.print("Starting count"); // this string is 12 characters long
for(int i=9; i >= 0; i--) // count down from 9
{
// the top line is row 0
lcd.setCursor(15,0); // move the cursor to the end of the string printed above
lcd.print(i);
delay(1000);
}
}
```