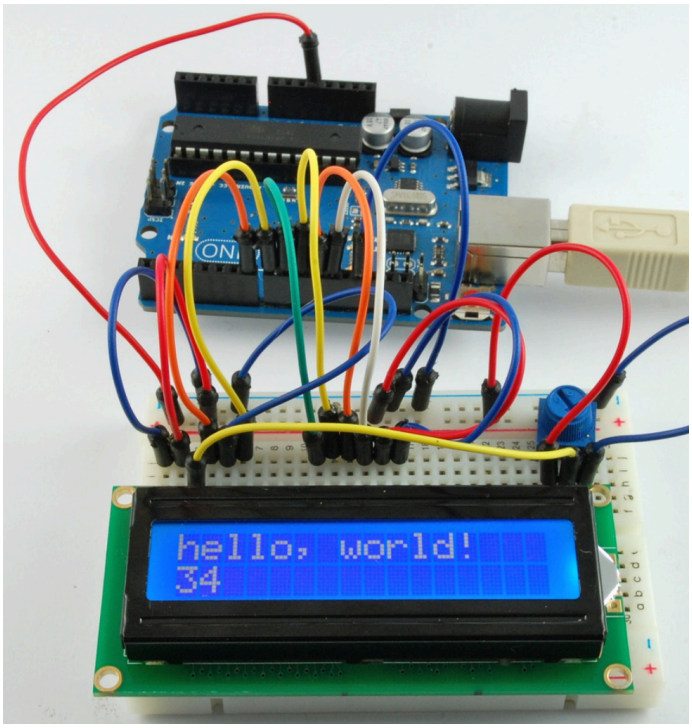
LCD

Overview

In this lesson, you will learn how to wire up and use an alphanumeric LCD display.



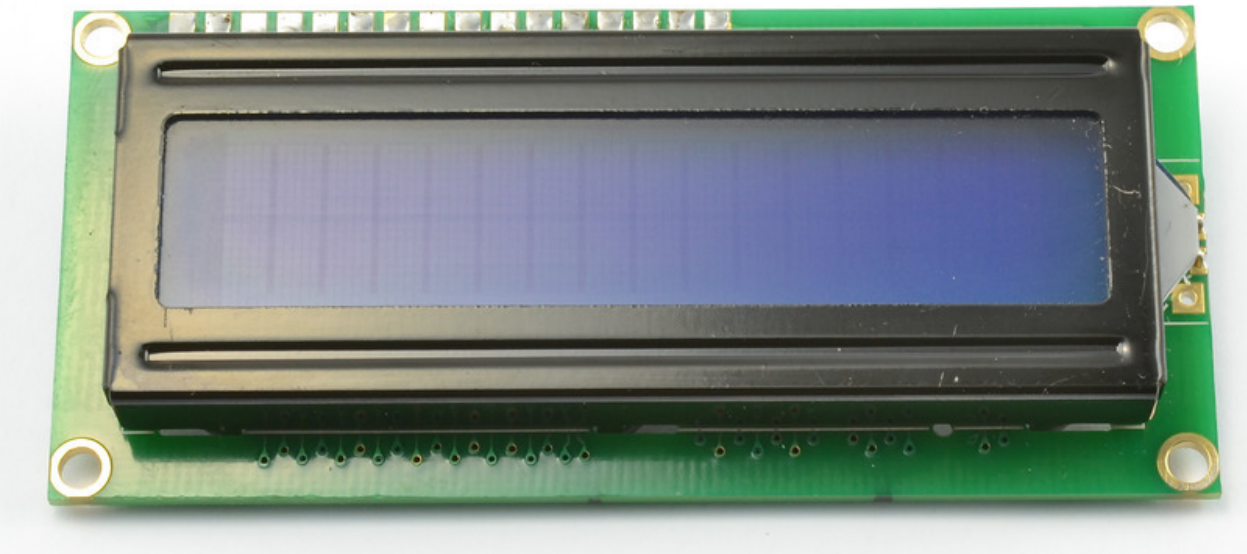
The display has an LED backlight and can display two rows with up to 16 characters on each row. You can see the rectangles for each character on the display and the pixels that make up each character. The display is just white on blue and is intended for showing text.

Parts

To build the project described in this lesson, you will need the following parts.

Part

Qty

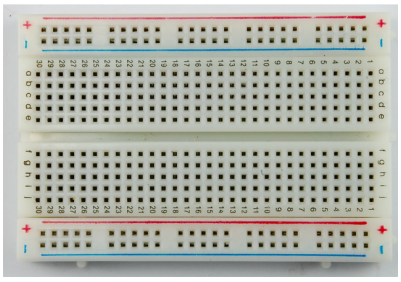


LCD Display (16x2 characters)

1

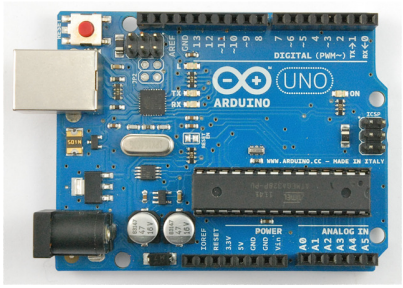
10 kΩ variable resistor (pot)

1



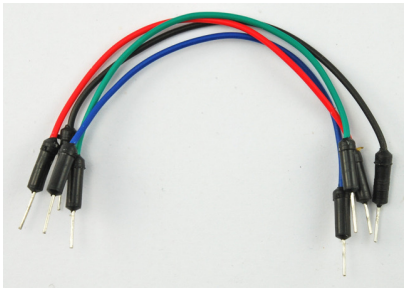
Half-size Breadboard

1



Arduino Uno R3

1



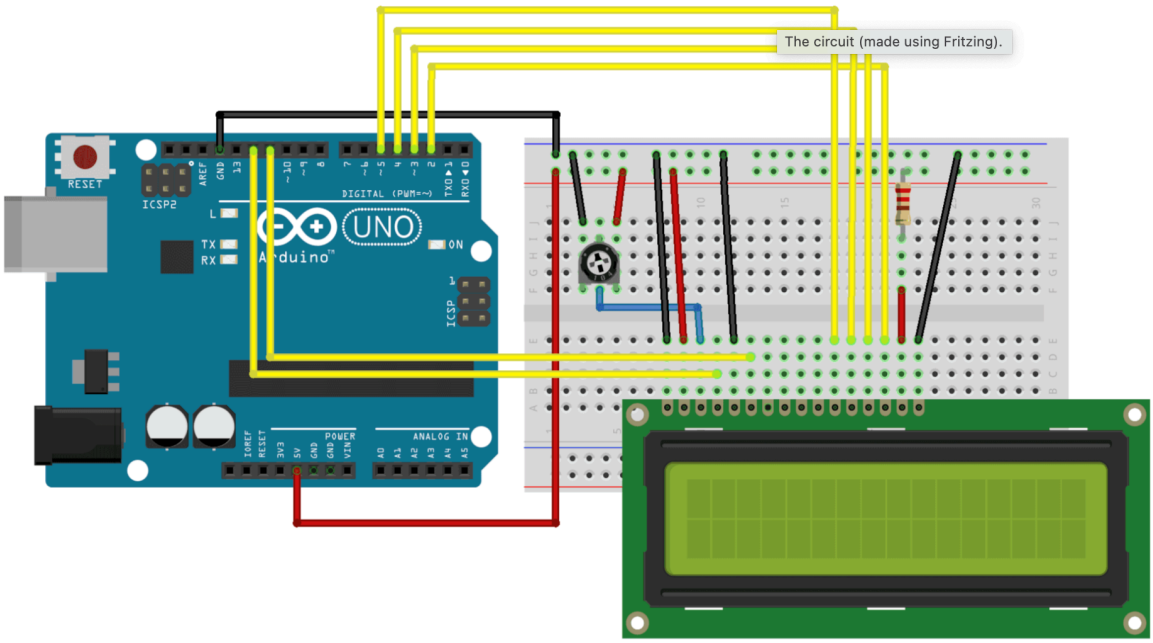
Jumper wire pack

1

In this lesson, we will run the Arduino example program for the LCD library, but in the next lesson, we will get our display to show the temperature and light level, using sensors.

Breadboard Layout

The LCD display needs six Arduino pins, all set to be digital outputs. It also needs 5V and GND connections. (The black circle is a Potentiometer…)



https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld

There are quite a few connections to be made. Lining up the display with the top of the breadboard helps to identify its pins without too much counting,

especially if the breadboard has its rows numbered with row 1 as the top row of the board. Do not forget, the long yellow lead that links the slider of the pot to pin 3 of the display. The 'pot' is used to control the contrast of the display.

You may find that your display is supplied without header pins attached to it. If so, follow the instructions in the next section.

Arduino Code

The Arduino IDE includes an example of using the LCD library which we will use. You can find this on the File menu under Examples → Liquid Crystal → HelloWorld.

This example uses different pins to the ones we use, so find the line of code below:

```
1. LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

and change it to be:

```
1. LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Upload the code to your Arduino board and you should see the message 'hello, world' displayed, followed by a number that counts up from zero.

The first thing of note in the sketch is the line:

```
1. #include <LiquidCrystal.h>
```

This tells Arduino that we wish to use the Liquid Crystal library.

Next we have the line that we had to modify. This defines which pins of the Arduino are to be connected to which pins of the display.

```
1. LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

The arguments to this are as follows:
Display Pin Name Display Pin Number Arduino Pin (in this example) RS 4 7 E 6 8 D4 11 9 D5 12 10 D6 13 11 D7 14 12

After uploading this code, make sure the backlight is lit up, and adjust the potentiometer all the way around until you see the text message
In the 'setup' function, we have two commands:

```
1.    lcd.begin(16, 2);
2.    lcd.print("hello, world!");
```

The first tells the Liquid Crystal library how many columns and rows the display has. The second line displays the message that we see on the first line of the screen.

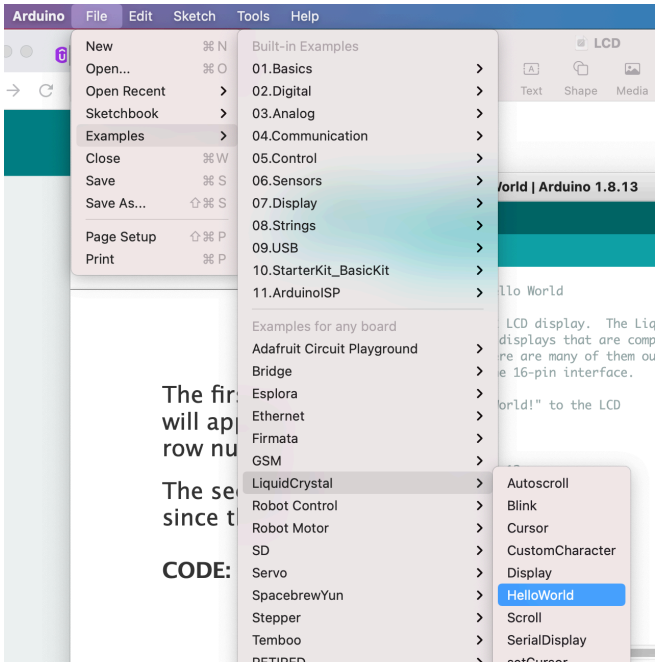
In the 'loop' function, we also have two commands:

```
1.    lcd.setCursor(0, 1);
2.    lcd.print(millis()/1000);
```

The first sets the cursor position (where the next text will appear) to column 0 & row 1. Both column and row numbers start at 0 rather than 1.

The second line displays the number of milliseconds since the Arduino was reset.

CODE: “Hello world” can be found as shown below



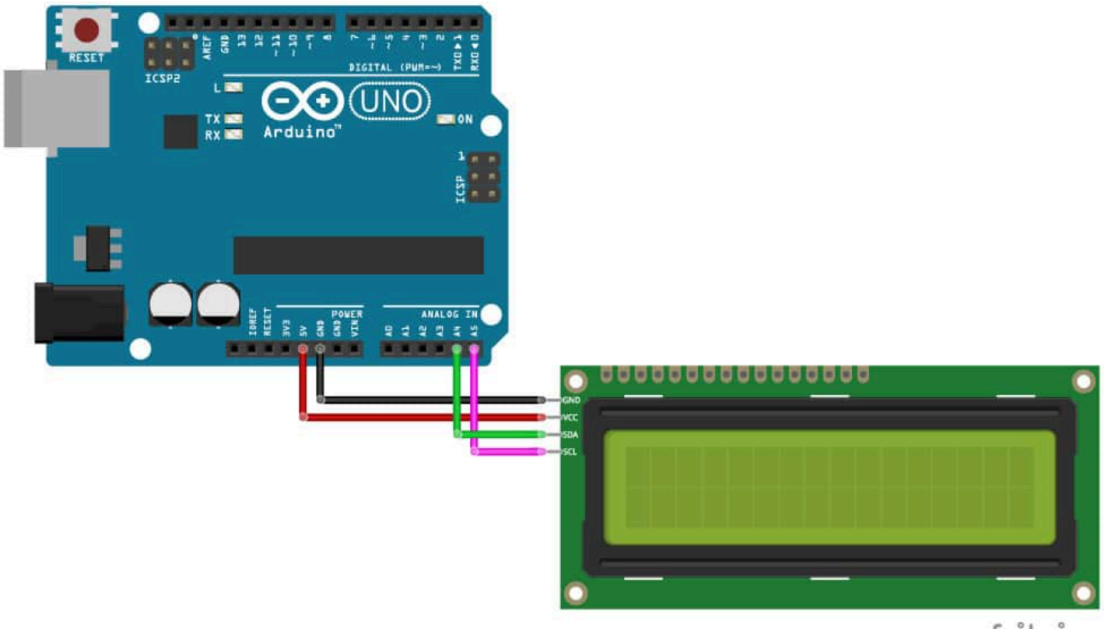
****

I2c LCD

USING I2C LCD(MUCH DIFFERENT):

i2c uses a serial bus instead of the more ‘pin consuming’ parralell bus system. The wiring will not use a single pin on the Digital side, but rather 2 A

pins from the Analog side, power and ground. You will also need to install the i2c library.



Displaying Text on I2C LCD

```
#include <LiquidCrystal_I2C.h>

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns,
totalRows);

void setup(){
lcd.init();
```

```
lcd.backlight(); // use to turn on and turn off
LCD back light
}

void loop()
{
lcd.setCursor(0, 0);
lcd.print("Microcontrollers");
lcd.setCursor(0,1);
lcd.print("I2C LCD tutorial");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Static text");
delay(1000);
lcd.setCursor(0,1);
lcd.print("I2C LCD tutorial");
delay(1000);
lcd.clear();
}
```

Scrolling text:

```
#include <LiquidCrystal_I2C.h>

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns,
totalRows);
```

```
String staticMessage = "I2C LCD Tutorial";
String scrollingMessage = "Welcome to
Microcontrollerslab! This is a scrolling
message.";

void scrollMessage(int row, String message, int
delayTime, int totalColumns) {
  for (int i=0; i < totalColumns; i++) {
    message = " " + message;
  }
  message = message + " ";
  for (int position = 0; position <
message.length(); position++) {
    lcd.setCursor(0, row);
    lcd.print(message.substring(position, position
+ totalColumns));
    delay(delayTime);
  }
}

void setup(){
  lcd.init();
  lcd.backlight();
}

void loop(){
  lcd.setCursor(0, 0);
  lcd.print(staticMessage);
  scrollMessage(1, scrollingMessage, 250,
totalColumns);
}
```

Display Custom Characters on I2C LCD using Arduino

In this section, we will display custom characters on the LCD screen.

For our 16×2 LCD display that we are using, we have the option to display custom characters as well. In this particular LCD, each block consists of 5×8 pixels. These can be used to display custom characters by setting the state of each pixel by inside a byte variable.

There is a very simple way to generate the byte variable of your own custom character. Head over to the following custom character generator: (Custom Character Generator for HD44780 LCD Modules- https://omerk.github.io/lcdchargen/).

Specify the custom character you want to display by clicking the pixels in the 5×8 pixel block and the corresponding byte variable will be generated.

In our case, we will display a '+' character on the screen. This is the byte variable that we will use in our program code to display this particular character on the LCD.

Custom character CODE: (character was created in above pic):

```
#include <LiquidCrystal_I2C.h>

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns, totalRows);

byte customChar[8] = {
  0b00000,
  0b00100,
  0b00100,
  0b11111,
  0b00100,
  0b00100,
  0b00000,
  0b00000
};

void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.createChar(0, customChar);
}

void loop()
{
 lcd.setCursor(0, 0);
 lcd.write(0);
}
```