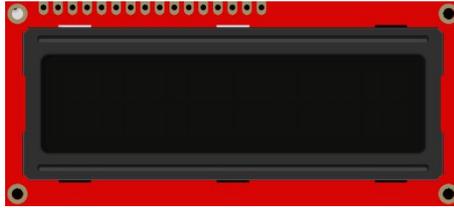


# SIK LCD Tutorial



## Introduction

An LCD is short for **Liquid Crystal Display**. When current is applied to this special kind of crystal, it turns opaque. This is used in a lot of calculators, watches, and many simple displays. Adding an LCD to your project will make it super portable and allow you to integrate up to 32 characters (16 x 2) of information into your Arduino project. The LCD has a lot of pins (16 pins in total) that we'll show you how to wire up. Lucky for you, four of the pins will not be used.

Let's take a look at how it works.

## Controlling the LCD

The LCD has 16 pins. The pins are numbered from left-to-right 1 through 16. The LCD utilizes an extremely common parallel interface LCD driver chip from Hitachi called the HD44780. Thankfully, the Arduino community has developed a library to handle a great deal of the software to hardware interface.

The pins are described as shown in the diagram below:



### Pin Descriptions:

Pin 1 -- Ground  
 Pin 2 -- VDD (5V) Power for the LCD  
 Pin 3 -- Contrast Adjust (0 - 5V)  
 Pin 4 -- Register Select (RS)  
 Pin 5 -- Read / Write Select (R/W)  
 Pin 6 -- Enable

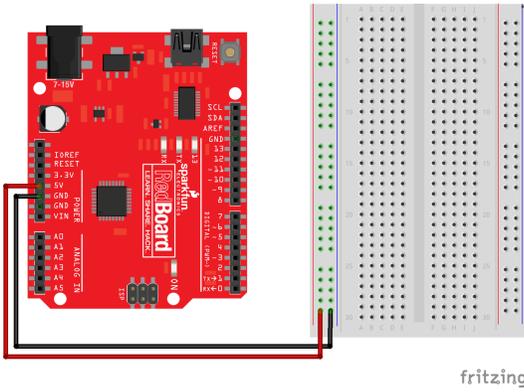
Pins 7-10 -- data lines d0 - d3 (not used)  
 Pins 11 - 14 -- data lines d4 - d7 (data transferred in 4-bits at a time)\*

Pin 15 -- Backlight Power (5V)  
 Pin 16 -- Backlight GND (GND)

## Wiring and Hook-up Guide

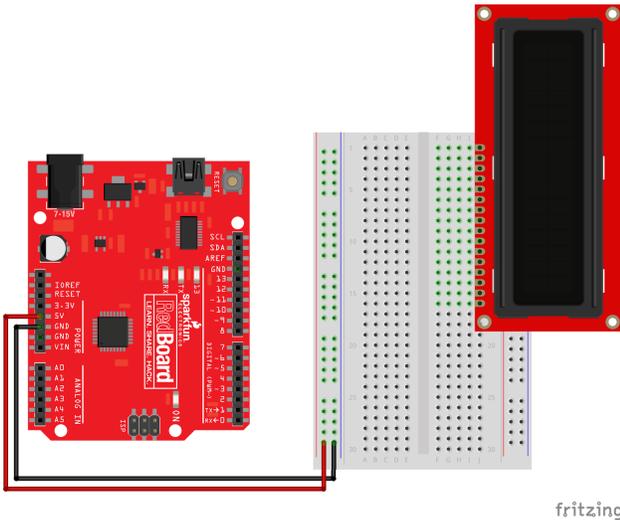
You will connect 6 pins from the Arduino to the LCD. These 6 pins will be for the register select, enable, and four data lines. The rest of the pins on the LCD will be connected to either 5V, GND, or not connected at all.

## 1. Connect 5V and GND



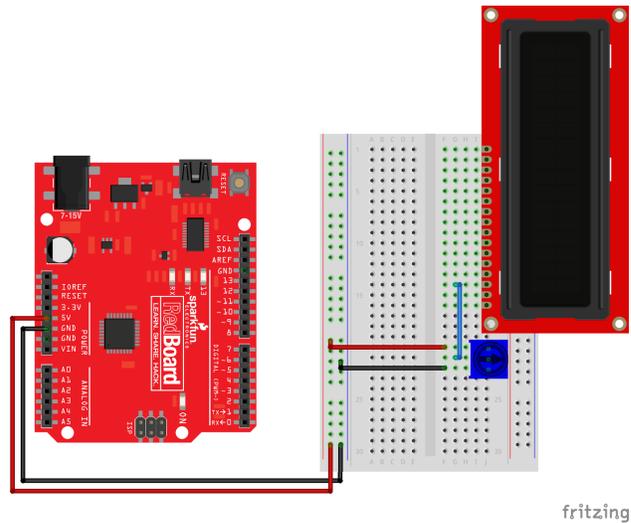
First, connect the 5V and GND pins from the Arduino to the breadboard power rails.

## 2. Mount the LCD



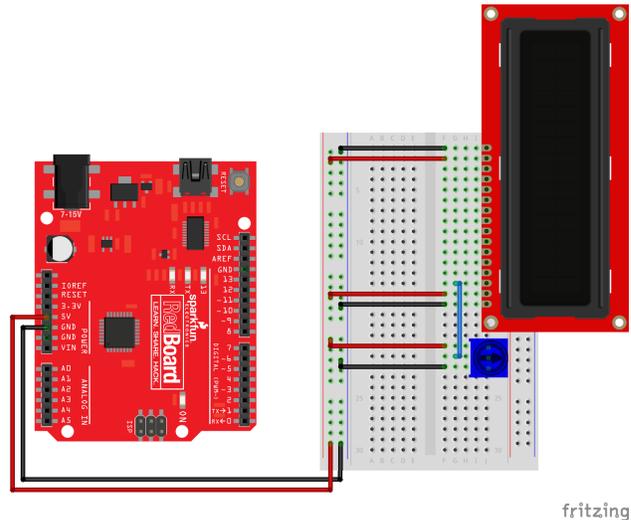
Insert the LCD starting on the top-most row of the breadboard. Notice that there are a total of 16 pins on the LCD. The last pin on the LCD should be in the first row of the breadboard.

## 3. Contrast Control



Pin 3 on the LCD controls the contrast and brightness of the LCD. Using a simple voltage divider with a potentiometer connected between 5V and GND, connect the center pin (wiper) of the potentiometer to pin3 on the LCD. As you rotate the knob on the potentiometer, you should notice that the screen will get brighter or darker (as soon as we connect up power to it).

## 4. Adding Power



The LCD has two separate power connections. One for the LCD itself and another one for the backlight. Add the following four wires to your project for both the LCD power and the backlight power.

- Connect pins 1 and 16 on the LCD to GND and pins 2 and 15 on the LCD to 5V.



# Arduino LiquidCrystal Library

The Arduino community has a library called **LiquidCrystal** that simplifies the controls and signals to the LCD. To use this, add this line to the very first line of code:

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

This includes additional code and functions needed to interface with the LCD. Pay close attention to the spelling, capitalization, and punctuation. To initialize the LCD, we declare a **LiquidCrystal** object using this line of code:

```
// LiquidCrystal lcd(RS, Enable, d4, d5, d6, d7); -- 6 control / signal pins
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
```

This initializer uses 6 parameters and specifies which Arduino pins are connected to the LCD's RS pin, Enable pin, and data pins: d4, d5, d6, and d7. In our line of code, we simply used Arduino pins 8 - 13 sequentially, but you don't have to do this. It's up to you. Use whatever I/O pins you have available on your Arduino.

## Using the LiquidCrystal Library

Now that you have declared a **LiquidCrystal** object, you can access special methods (aka functions) that are specific to the LCD. Here is an overview of a few that you will find useful:

---

```
// initializes interface to the LCD screen and specifies the size (width and height).
lcd.begin(16, 2);
```

---

```
// clears the LCD screen and moves the cursor to the upper-left corner.
lcd.clear();
```

---

```
// prints text to the LCD.
lcd.print("Your message")
```

---

```
// Position the LCD cursor; any new text written to the LCD will be displayed starting here.
// The upper left corner is considered col = 0, row = 0
lcd.setCursor(col, row)
```

---

## Example Sketch Using the LiquidCrystal Library

This is a little example code to get started with. It is an adaptation of the code from SIK Circuit #15.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

void setup()
{
  lcd.begin(16, 2); // initializes interface to LCD
  lcd.clear();

  lcd.print("hello, world!"); // prints "hello, world!" to the LCD.
}
void loop()
{
  lcd.setCursor(0, 1); // move cursor to the 2nd line (col 0, row 1)
  lcd.print(millis()/1000); // Print our value
}
```