

Print Text on LCD Character Display

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

The LiquidCrystal library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface. This example sketch prints "Hello World!" to the LCD and shows the time in seconds since the Arduino was reset. The pot controls the brightness of the display.

HARDWARE REQUIRED

Arduino Board
LCD Screen (compatible with Hitachi HD44780 driver)
pin headers to solder to the LCD display pins
10k Potentiometer
breadboard
hook-up wire

CIRCUIT

The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins:

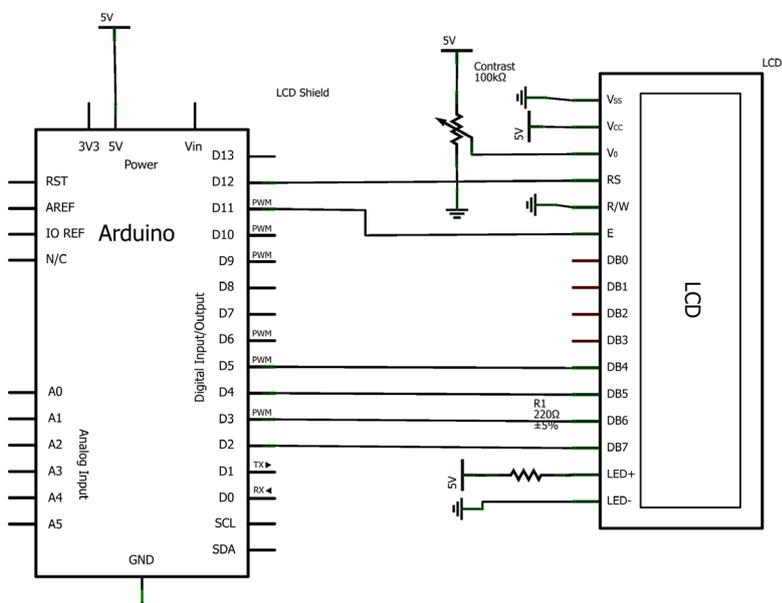
A register select (RS) pin that controls where in the LCD's memory you're writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.
A Read/Write (R/W) pin that selects reading mode or writing mode
An Enable pin that enables writing to the registers
8 data pins (D0 -D7). The states of these pins (high or low) are the bits that you're writing to a register when you write, or the values you're reading when you read.

There's also a display contrast pin (Vo), power supply pins (+5V and Gnd) and LED Backlight (Bklt+ and Bklt-) pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

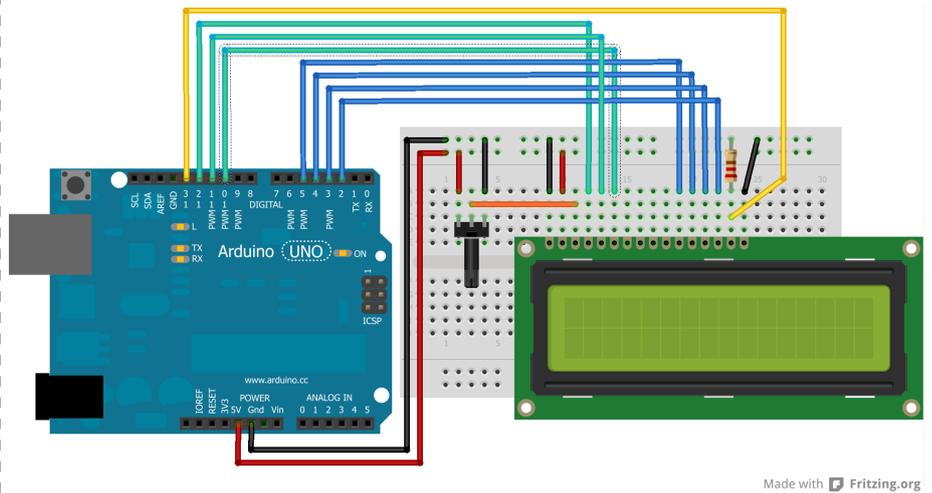
The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. The LiquidCrystal Library simplifies this for you so you don't need to know the low-level instructions.

The Hitachi-compatible LCDs can be controlled in two modes: 4-bit or 8-bit. The 4-bit mode requires seven I/O pins from the Arduino, while the 8-bit mode requires 11 pins. For displaying text on the screen, you can do most everything in 4-bit mode, so example shows how to control a 2x16 LCD in 4-bit mode.

SCHEMATIC



IMAGE



Made with Fritzing.org

CODE

```
// character LCD example code
// www.hacktronics.com

#include <LiquidCrystal.h>

// Connections:
// rs (LCD pin 4) to Arduino pin 12
// rw (LCD pin 5) to Arduino pin 11
// enable (LCD pin 6) to Arduino pin 10
// LCD pin 15 to Arduino pin 13
// LCD pins d4, d5, d6, d7 to Arduino pins 5, 4, 3, 2
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

int backlight = 13; // pin 13 will control the backlight

void setup()
{
  pinMode(backlight, OUTPUT);
  digitalWrite(backlight, HIGH); // turn backlight on. Replace 'HIGH' with
  'LOW' to turn it off.
  lcd.begin(16,2); // columns, rows. use 16,2 for a 16x2 LCD, etc.
  lcd.clear(); // start with a blank screen
  lcd.setCursor(0,0); // set cursor to column 0, row 0 (the first row)
  lcd.print("Hello, World"); // change this text to whatever you like. keep it
  clean.
  lcd.setCursor(0,1); // set cursor to column 0, row 1
  lcd.print("hacktronics.com");

  // if you have a 4 row LCD, uncomment these lines to write to the bottom rows
  // and change the lcd.begin() statement above.
  //lcd.setCursor(0,2); // set cursor to column 0, row 2
  //lcd.print("Row 3");
  //lcd.setCursor(0,3); // set cursor to column 0, row 3
  //lcd.print("Row 4");
}

void loop()
{
}
}
```

Before wiring the LCD screen to your Arduino we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above.

To wire your LCD screen to your Arduino, connect the following pins:
LCD RS pin to digital pin 12
LCD Enable pin to digital pin 11
LCD D4 pin to digital pin 5
LCD D5 pin to digital pin 4
LCD D6 pin to digital pin 3
LCD D7 pin to digital pin 2

Additionally, wire a 10K pot to +5V and GND, with it's wiper (output) to LCD screens VO pin (pin3).