

A REPORT ON LCD Display

INTRODUCTION

The LCD (**Liquid Crystal Display**) is a type of display that uses the liquid crystals for its operation.

Here, we will accept the serial input from the computer and upload the sketch to the [Arduino](#). The characters will be displayed on the LCD.

The library that allows us to control the [LCD](#) display is called **Liquid Crystal Library**, which is discussed below:

The library is declared as:

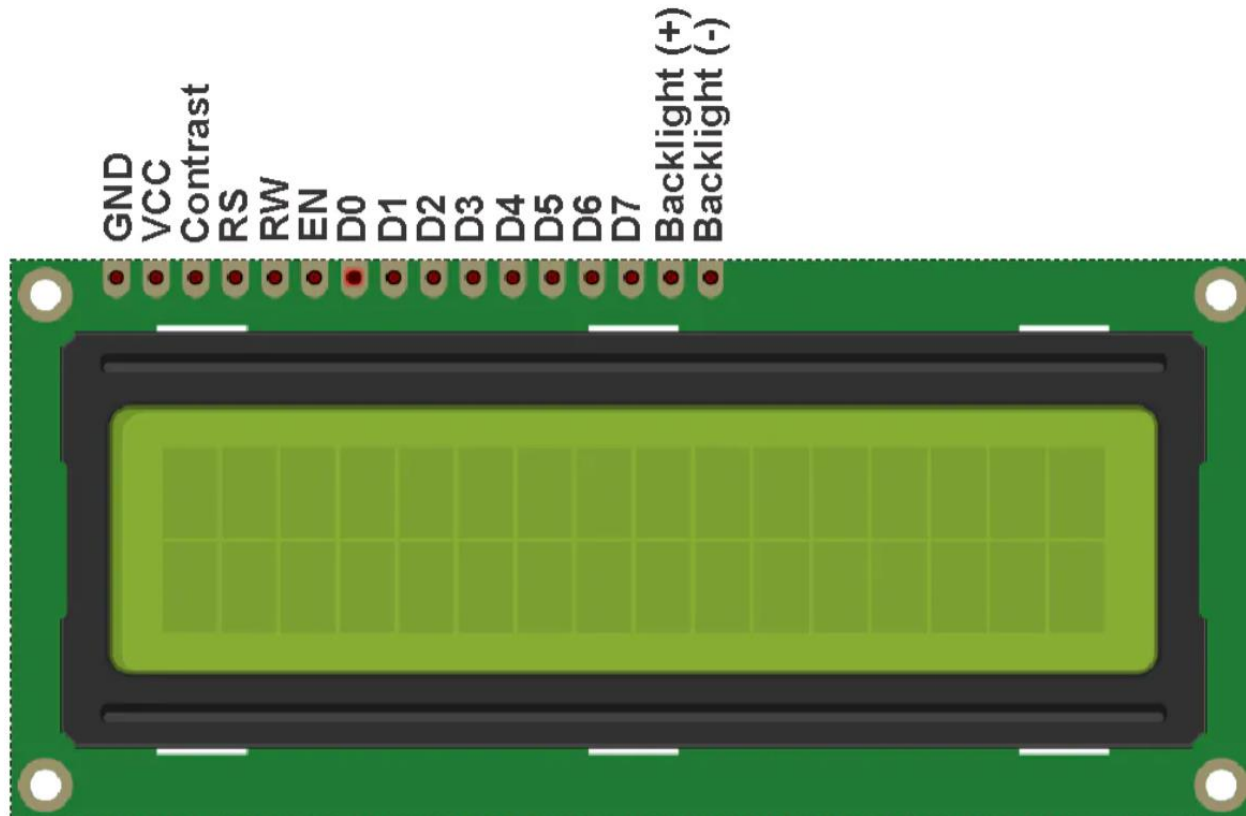
```
#include <LiquidCrystal.h>
```

The library is based on a compatible chipset called **Hitachi HD44780**. It is found on most of the LCDs that are based on text. It works with either an 8-bit mode or 4-bit mode. Here, the bit mode signifies the data lines in addition to the enable, rs, and rw control lines

LCD Structure

The LCD display has a 16-pin interface.

The structure of the LCD is shown below:



The Liquid Crystal Display has a parallel interface. It means that the microcontroller operates several pins at once to control the LCD display.

The 16-pins present on the LCD display are discussed below:

RS

The **Register Select (RS)** pin controls the memory of the LCD in which we write the data. We can select either the **data register** or the **instruction register**. The LCD looks for the upcoming instruction, which is present in the instruction register.

R/W

The Read/Write pin selects the reading or writing mode.

E

The **Enable (E)** mode is used to enable the writing to the registers. It sends the data to the data pins when the mode is HIGH.

D0 to D7

These are eight data pins numbered as D0, D1, D3, D3, D4, D5, D6, and D7. We can set the state of the data pin either HIGH or LOW.

Pin 1 of the LCD is the **Ground** pin, and pin 2 is the **Vcc** or the voltage source pin.

The pin 3 of the LCD is the **VEE** or the **contrast pin**. For example, we can connect the potentiometer's output to the VEE and can adjust the contrast of the LCD.

The A and K pins are also called as **Backlight pins** (Bklt+ and Bklt-).

Principle of LCD

The process includes putting the data (to be displayed on the LCD screen) into the data registers. The instructions in the Register Select are kept in the instruction register. The liquid crystal library has simplified process to display the characters on the LCD.

The LCDs can be controlled in 4-bit or 8-bit modes, which requires 7 and 11

Input/Output pins from the particular Arduino board.

Let's discuss a project to display the text on the LCD.

Hardware Required

The components required for the project are listed below:

LCD Screen (Hitachi HD44780 compatible driver display)

1 x 220 Ohm Resistor

1 x 10K Ohm Resistor

Arduino UNO board or Genuino board

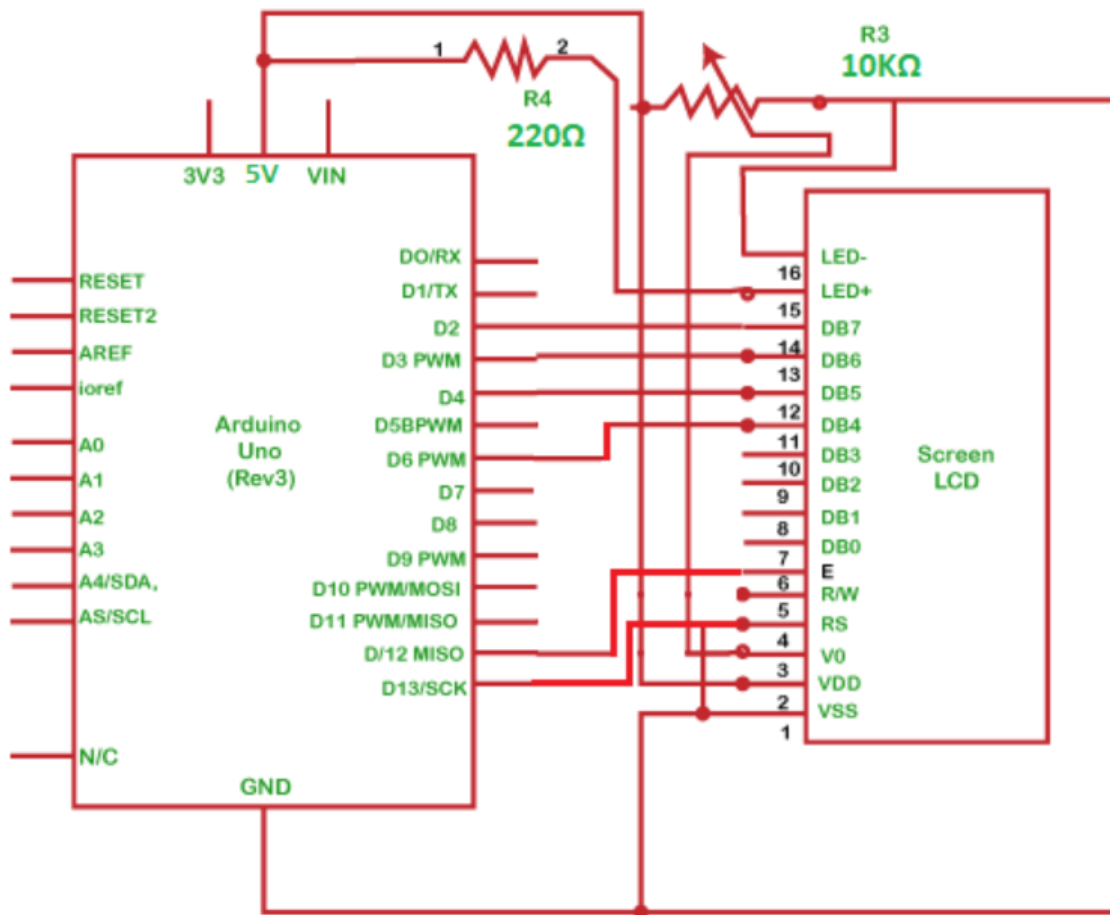
Jump wires

Pin header required to solder the LCD display pins

Breadboard

Structure of the project

The structure of the project is shown below:



Procedure

The connection is explained below:

We need to first connect the data pins of LCD to the digital pins.

Connect the RS pin of LCD to pin 13 of the Arduino board.

Connect the Enable pin of LCD to pin 12 of the Arduino board.

Connect the D4 pin of LCD to pin 6 of the Arduino board.

Connect the D5 pin of LCD to pin 4 of the Arduino board.

Connect the D6 pin of LCD to pin 3 of the Arduino board.

Connect D7 pin of LCD to pin 2 of the Arduino board.

Connect the middle terminal of the potentiometer to the VEE (contrast pin).

Connect the two ends of the potentiometer to the Ground and 5V.

Connect one end of a resistor to the A and K of the LCD and another end to 5V.

Sketch

The code to display the specified message on the LCD display is given below:

```
// here, we will include the liquid crystal library:
#include <LiquidCrystal.h>

// initialize the library with the pins on the Arduino board
LiquidCrystal lcd(13, 12, 6, 4, 3, 2);

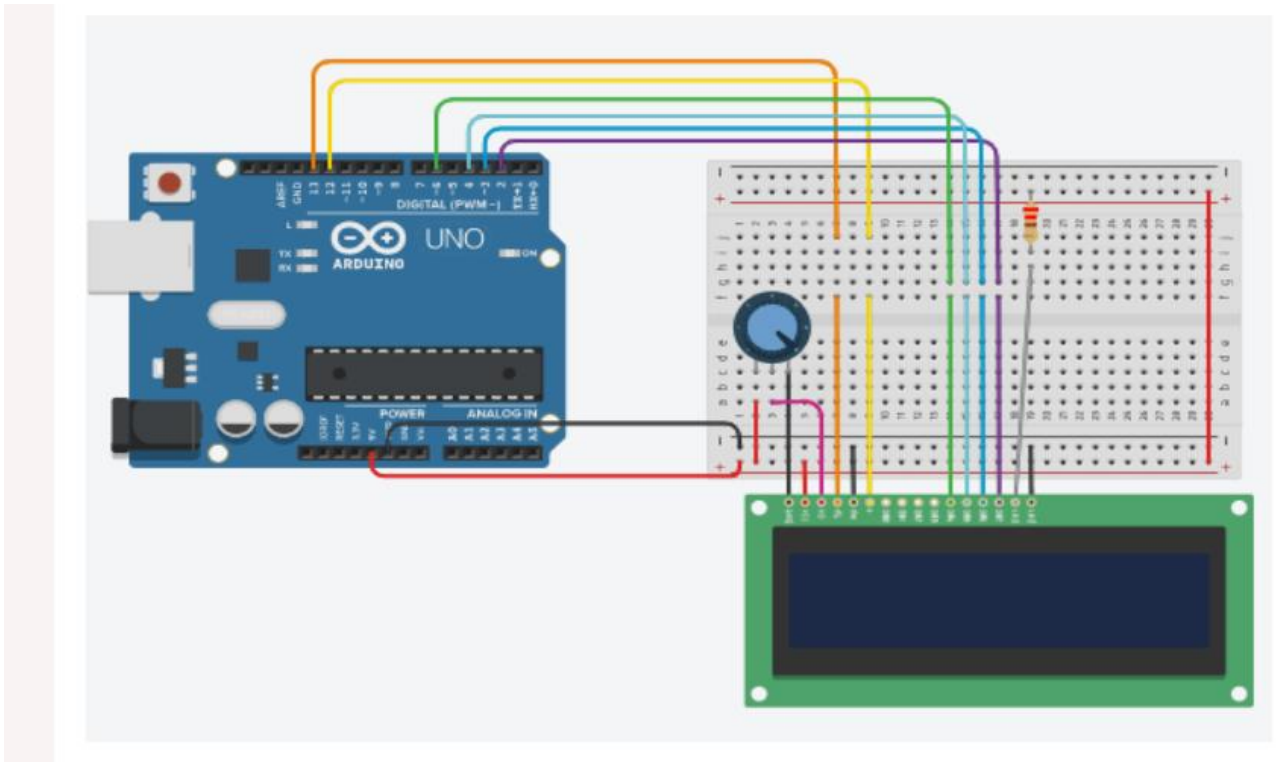
void setup() {
    // Here, 16 and 2 are the columns and rows of the LCD
    lcd.begin(16, 2);
    // It prints the message on the LCD.
    lcd.print("hello Arduino");
    // We can modify the message as per our choice.
}

void loop() {
    // It sets the cursor to column 0, line 1
    // Since counting begins with 0, line 1 is the second row
    lcd.setCursor(0, 1);
    // print the number of seconds
    lcd.print(millis() / 1000);
    // Here, millis() is the return type of the timer in milliseconds
}
```

Connection Diagram

We will show the connection using the Simulator so that the connections become clearer and more precise.

We can make the same connection using the hardware devices.



Output

The message 'Hello Arduino' will be printed on the LCD display, as shown below:

