

## EXPERIMENT NO. 7

**Aim :** To interface 16×2 LCD with Arduino UNO and write a program to display numbers and/or text on it.

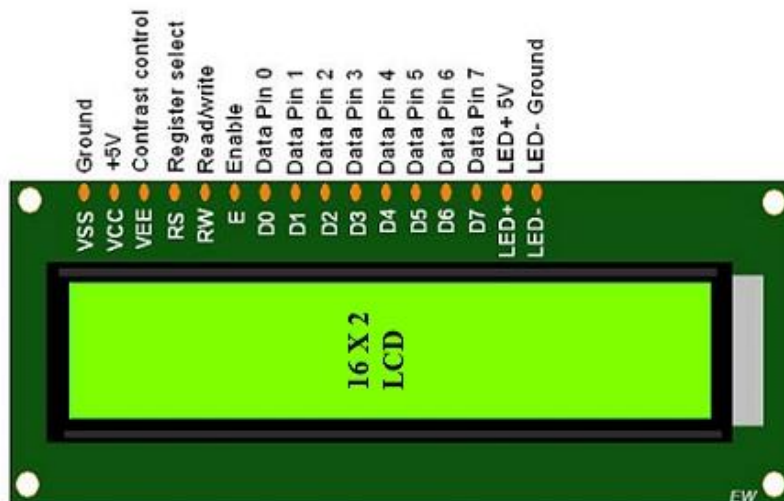
**Apparatus Required :** 16×2 LCD module, Arduino UNO board, 330  $\Omega$  resistance, 10 k $\Omega$  trimmer breadboard, jumper wires.

### **Theory :**

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

### LCD 16×2 Pin Diagram

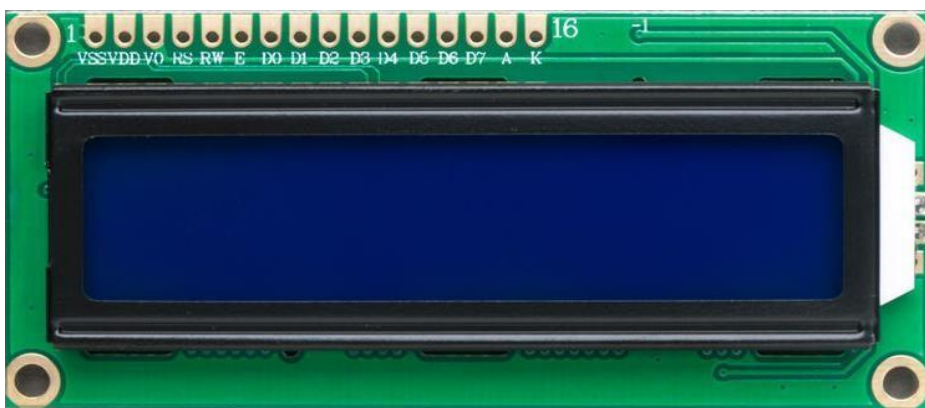
The 16×2 LCD pinout is shown below.



- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.

- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
  - Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
  - Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
  - Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
  - Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
  - Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
  - Pin15 (+ve pin of the LED): This pin is connected to +5V
  - Pin 16 (-ve pin of the LED): This pin is connected to GND.
- lcd-16x2-pin-diagram.

### **Hitachi HD44780 chipset**

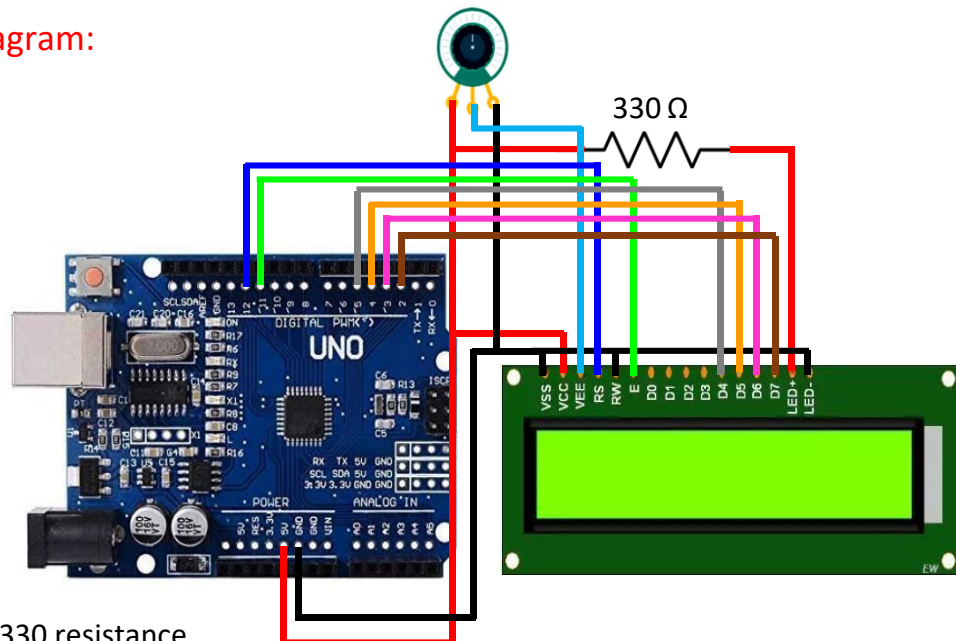


## Circuit Diagram:

### Connection diagram:

#### Legend:

1. RS to pin 12
2. En to pin 11
3. D4 to pin 5
4. D5 to pin 4
5. D6 to pin 3
6. D7 to pin 2
7. RW to GND
8. LED- to GND
9. Vss to GND
10. Vcc to 5V
11. LED+ to 5V via 330 resistance
12. V0 to trimmer variable



## Code :

//Code 1- Simple text display with time elapsed in seconds

```
#include <LiquidCrystal.h>
short int rs=12,
        en=11,
        d4=5,
        d5=4,
        d6=3,
        d7=2;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);

void setup(){
  lcd.begin(16,2);
  lcd.print("Hello World!"); //print a message on the LCD
}

void loop(){
  lcd.setCursor(0,1); //set the cursor to column 0, line 1
  lcd.print(millis()/1000); //print the number of seconds since reset
  lcd.print(" seconds");
}
```

//Code 2 - Blinking cursor

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

```
short int rs=12,
```

```
        en=11,
```

```
        d4=5,
```

```
        d5=4,
```

```
        d6=3,
```

```
        d7=2;
```

```
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
```

```
void setup() {
```

```
  lcd.begin(16,2); //set up the LCD number of columns and rows
```

```
  lcd.print("Hello World"); //print a message on the LCD
```

```
}
```

```
void loop() {
```

```
  lcd.noBlink();    //Turn off blinking cursor
```

```
  delay(500);
```

```
  lcd.blink();      //Turn on blinking cursor
```

```
  delay(500);
```

```
}
```

//Code 3- Blinking cursor (underscore)

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

```
short int rs=12,
```

```
        en=11,
```

```
        d4=5,
```

```
        d5=4,
```

```
        d6=3,
```

```
        d7=2;
```

```
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);
```

```
void setup() {
```

```
  lcd.begin(16,2); //set up the LCD number of columns and rows
```

```
  lcd.print("Hello World"); //print a message on the LCD
```

```
}
```

```
void loop() {
```

```
  lcd.noCursor();   //Turn off blinking cursor
```

```
  delay(500);
```

```
  lcd.cursor();     //Turn on blinking cursor
```

```
  delay(500);
```

```
}
```

```

//Code 4- Parse the print data from serial monitor

#include <LiquidCrystal.h>
int rs=12,
    en=11,
    d4=5,
    d5=4,
    d6=3,
    d7=2;
LiquidCrystal lcd(rs,en,d4,d5,d6,d7);

void setup() {
  lcd.begin(16,2); //set up the LCD number of columns and rows
  Serial.begin(9600); //Initialize the serial communication
}

void loop() {
  //if the character arrives on serial port
  if(Serial.available()){
    delay(100); //Wait for complete mesaage to arrive
    lcd.clear(); // Clear the display
    //Read the characters and print on LCD
    while(Serial.available()>0){
      lcd.write(Serial.read()); //display each character to the LCD
    }
  }
}

```

**Result :** Hence, We successfully 16×2 LCD with Arduino UNO and write a program to display numbers and/or text on it.

