

Name: Shashank Bagda	Roll Number: 92100133020
Subject Name and Code: FSSI – 01CT1103	Date of Experiment: 10-12-21

Task: Configure Counter using LCD (16*2) with the Arduino Board.

Components:

16x2 LCD Display, Arduino UNO, Jumper wire, push button, breadboard, resistor, potentiometer

About the Project:

In these project we will configure the that we button is pressed the LCD display shows he button is pressed and the connection are as follows:

So first we will connect push button to the arduino so we connect three wires to the Ardiuno board.

First wire will be conneted to 5 volt supply through one leg of push button and 2.2KOhms pull-up resistor, the second wire will be connected to ground through corresponding leg of push button, the third wire will be connected to the digital gpio pin which reads the button's state that weather the button is pressed or not,

A 16*2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5*7 pixel matrix. The 16*2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely Command and Data. It is one kind if electronic display module used in an extensive range of applications like various circuits and devices like mobile phones, calculators, computers, etc. These displays are mainly prefered 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.

Pin 1 (Ground/ Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.

Pin 2 (VCC / Source pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.

Pin 3 (V0 / VEE / Control pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.

Pin 4 (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).

Pin 5 (Read / Write / Control pin): This pin toggles the display among the read and write operations, and it is connected to a microcontroller unit pin to get either 0 or 1(0 = Write Operation and 1 = Read operation).

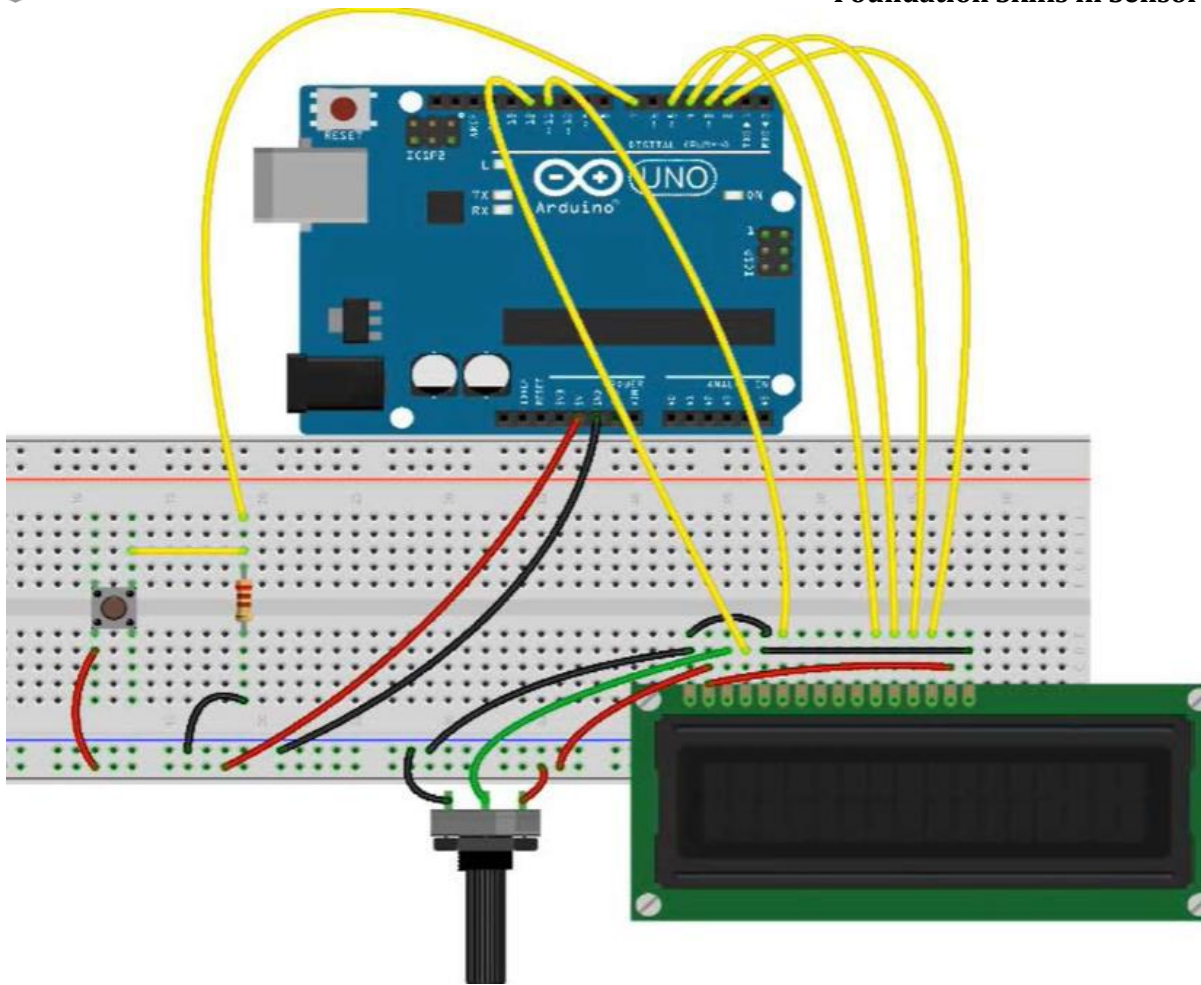
Pin 6 (Enable / Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit and constantly held high.

Pin7-14 (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.

Pin 15 (+ve pin of the LED): This pin is connected to +5V.

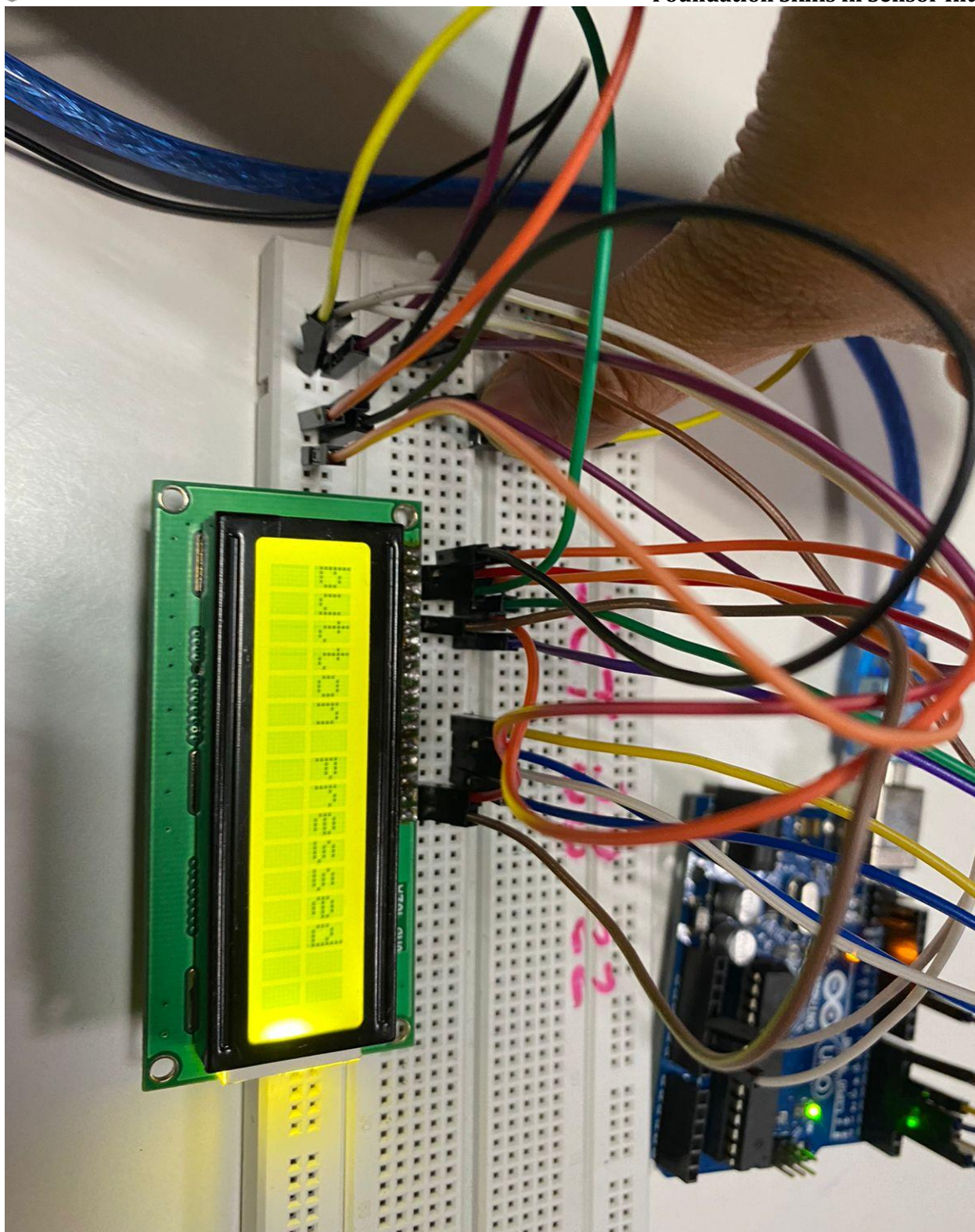
Pin 16 (-ve pin of the LED): This pin is connected to GND.

Schematic:



Output: (your circuit implementation and its working photo)





Code:

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



```
LiquidCrystal lcd(12,11,5,4,3,2);
int pbutton=7;
void setup() {
  // put your setup code here, to run once:
  lcd.begin(16,2);
  pinMode(pbutton, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(digitalRead(pbutton)==HIGH)
  {
    lcd.setCursor(0,0);
    lcd.print("button pressed");
  }
}
```

Application:

This experiment is used to for tally counter projects. Sometimes it is used in counter of the people at public places to keep watch on incoming and outgoing people entries.

Conclusion:

By doing these project we get to know that how to interface lcd with push button, the pushbutton is a component that connects two points of the circuit and when pressed, it turns on and the 16*2 LCD display it. There is a combination of LCD display and pushbutton in these project, when we push the button, LCD displays the button is pressed.

