Enrolment/ Roll No.92100588007

Practical No. 7

Aim: Introduction to Basic IoT Components.

Objectives:

1. To learn Arduino UNO basics
2. To interface Seven Segment Display (SSD) with Arduino and write a program to print numbers from 1 to 4 on SSD.

Theory:

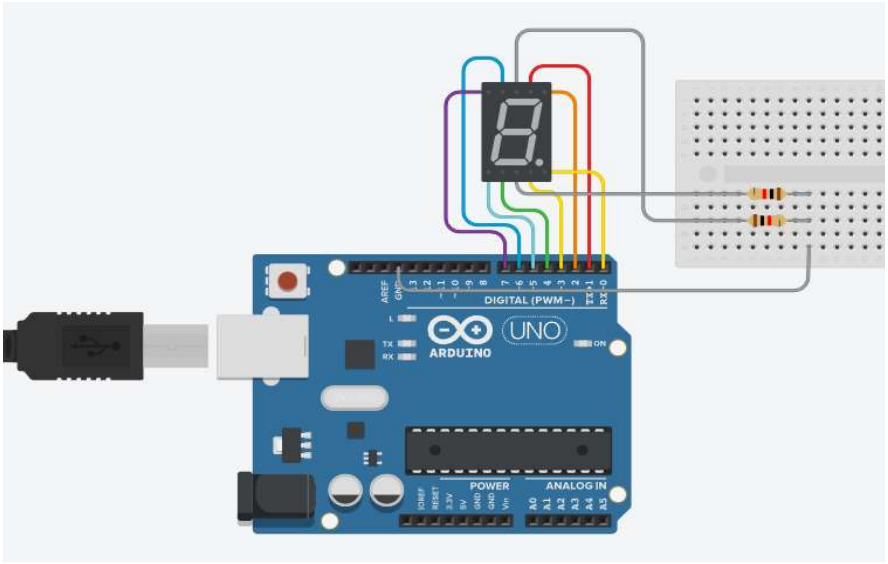
Hardware:

- Arduino board
- Seven segment display
- 220 ohm resistor
- Jumper wires

Function:

- To display a number on the SSD, you need to set the corresponding segments to HIGH. For example, to display the number "1", you would set segments "b" and "c" to HIGH. To display the number "4", you would set segments "b", "c", and "f" to HIGH.

Circuit Diagram: (Download from tinkercad.com)



Program:

```
// Seven Segment Display Program

const int SSD_PINS[] = {1, 2, 3, 4, 5, 6, 7, 0}; // Digital pins connected to the SSD
const int NUM_DIGITS = 4; // Number of digits on the SSD

void setup() {
  // Set all SSD pins to output
  for (int i = 0; i < NUM_DIGITS; i++) {
    pinMode(SSD_PINS[i], OUTPUT);
  }
}

void loop() {
  // Display the numbers from 1 to 4
  for (int i = 0; i < 4; i++) {
    displayDigit(i);
    delay(1000); // Wait for 1 second
  }
}

void displayDigit(int digit) {
  // Turn on the appropriate segments of the SSD to display the given digit
  switch (digit) {
    case 0:
      digitalWrite(SSD_PINS[0], HIGH);
      digitalWrite(SSD_PINS[1], HIGH);
      digitalWrite(SSD_PINS[2], HIGH);
      digitalWrite(SSD_PINS[3], HIGH);
      digitalWrite(SSD_PINS[4], HIGH);
```

```
    digitalWrite(SSD_PINS[5], HIGH);
    digitalWrite(SSD_PINS[6], LOW);
    break;
  case 1:
    digitalWrite(SSD_PINS[0], LOW);
    digitalWrite(SSD_PINS[1], HIGH);
    digitalWrite(SSD_PINS[2], HIGH);
    digitalWrite(SSD_PINS[3], LOW);
    digitalWrite(SSD_PINS[4], LOW);
    digitalWrite(SSD_PINS[5], LOW);
    digitalWrite(SSD_PINS[6], LOW);
    break;
  case 2:
    digitalWrite(SSD_PINS[0], HIGH);
    digitalWrite(SSD_PINS[1], HIGH);
    digitalWrite(SSD_PINS[2], LOW);
    digitalWrite(SSD_PINS[3], HIGH);
    digitalWrite(SSD_PINS[4], LOW);
    digitalWrite(SSD_PINS[5], HIGH);
    digitalWrite(SSD_PINS[6], HIGH);
    break;
  case 3:
    digitalWrite(SSD_PINS[0], HIGH);
    digitalWrite(SSD_PINS[1], HIGH);
    digitalWrite(SSD_PINS[2], HIGH);
    digitalWrite(SSD_PINS[3], HIGH);
    digitalWrite(SSD_PINS[4], LOW);
    digitalWrite(SSD_PINS[5], LOW);
    digitalWrite(SSD_PINS[6], HIGH);
    break;
```

```
    case 4:
      digitalWrite(SSD_PINS[0], LOW);
      digitalWrite(SSD_PINS[1], HIGH);
      digitalWrite(SSD_PINS[2], HIGH);
      digitalWrite(SSD_PINS[3], LOW);
      digitalWrite(SSD_PINS[4], HIGH);
      digitalWrite(SSD_PINS[5], LOW);
      digitalWrite(SSD_PINS[6], HIGH);
      break;
  }
}
```

Output: (Screenshot of LED On)

Conclusion:Interfacing an SSD with Arduino is a simple way to display numbers and other characters on your projects. The code above can be easily modified to display any number or character that you want.