Aim:- Write code to control the LED and SSD

1. Theory

LED (Light Emitting Diode)

An LED is a semiconductor device that emits light when current flows through it. LEDs are

commonly used in electronics to indicate status or provide visual feedback.

7-Segment Display (SSD)

A 7-segment display is an electronic display device used to display decimal numbers and some letters. It consists of 7 LEDs arranged in a figure-eight pattern and can be controlled to display numbers 0-9 by turning on specific segments.

Arduino

The Arduino is an open-source microcontroller platform used to control various electronic

components. It can be programmed to read inputs, control outputs, and interact with other

hardware.

3. Materials Required

 Arduino Uno board

 7-Segment Display (Common Cathode)

 LED

 220-ohm resistors (for LED and SSD)

 Breadboard

 Jumper wires

4. Procedure

Part 1: Connecting the LED

1. Connect the LED:

o Place the LED on the breadboard.

o Connect the anode (long leg) of the LED to digital pin 9 on the Arduino through a

220-ohm resistor.

o Connect the cathode (short leg) of the LED to the ground (GND) on the Arduino.

Part 2: Connecting the 7-Segment Display

1. Identify Pins:

o The 7-segment display typically has 8 pins: 7 for each segment (labeled A to G)

and one common cathode pin.

2. Connect the 7-Segment Display:

o Connect the segment pins (A to G) to Arduino digital pins (e.g., 2 to 8).

o Connect the common cathode pin to GND through a 220-ohm resistor.

5. Code

Arduino Sketch for LED and 7-Segment Display

cpp

Copy code

// Define pins for the 7-segment display segments

const int segA = 2;

const int segB = 3;

const int segC = 4;

const int segD = 5;

const int segE = 6;

const int segF = 7;

const int segG = 8;

// Define pin for the LED

const int ledPin = 9;

void setup() {

// Set segment pins as outputs

pinMode(segA, OUTPUT);

pinMode(segB, OUTPUT);

pinMode(segC, OUTPUT);

pinMode(segD, OUTPUT);

pinMode(segE, OUTPUT);

pinMode(segF, OUTPUT);

pinMode(segG, OUTPUT);

// Set LED pin as output

pinMode(ledPin, OUTPUT);

}

void loop() {

// Turn LED on and off

digitalWrite(ledPin, HIGH); // Turn LED on

delay(1000); // Wait for a second

digitalWrite(ledPin, LOW); // Turn LED off

delay(1000); // Wait for a second

// Display digits 0-9 on the 7-segment display

for (int i = 0; i &lt; 10; i++) {

displayDigit(i);

delay(2000); // Display each digit for 2 seconds

}

}

// Function to display a digit on the 7-segment display

void displayDigit(int digit) {

// Array to represent the segments for digits 0-9

// 0bGFEDCBA (binary representation)

const byte digitPatterns[10] = {

0b00111111, // 0

0b00000110, // 1

0b01011011, // 2

0b01001111, // 3

0b01100110, // 4

0b01101101, // 5

0b01111101, // 6

0b00000111, // 7

0b01111111, // 8

0b01101111 // 9

};

// Turn off all segments

digitalWrite(segA, LOW);

digitalWrite(segB, LOW);

digitalWrite(segC, LOW);

digitalWrite(segD, LOW);

digitalWrite(segE, LOW);

digitalWrite(segF, LOW);

digitalWrite(segG, LOW);

// Turn on segments for the current digit

byte pattern = digitPatterns[digit];

digitalWrite(segA, pattern &amp; 0b00000001);

digitalWrite(segB, pattern &amp; 0b00000010);

digitalWrite(segC, pattern &amp; 0b00000100);

digitalWrite(segD, pattern &amp; 0b00001000);

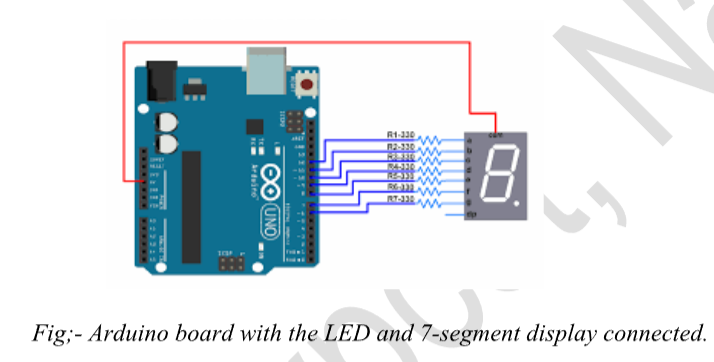
digitalWrite(segE, pattern &amp; 0b00010000);

digitalWrite(segF, pattern &amp; 0b00100000);

digitalWrite(segG, pattern &amp; 0b01000000);

}

6. Image



7. Working

LED Control:

The LED connected to pin 9 blinks on and off every second. This demonstrates basic digital

output control using the Arduino.

7-Segment Display Control:

The 7-segment display shows digits 0 through 9 sequentially, each for 2 seconds. The

displayDigit() function uses predefined patterns to illuminate the appropriate segments of the display to represent each digit.

8. Conclusion

The lab successfully demonstrated how to control an LED and a 7-segment display using an

Arduino. Students learned to wire up basic electronic components and write code to control

them, understanding the concepts of digital output and display management. This exercise

provided foundational skills in interfacing components with an Arduino.