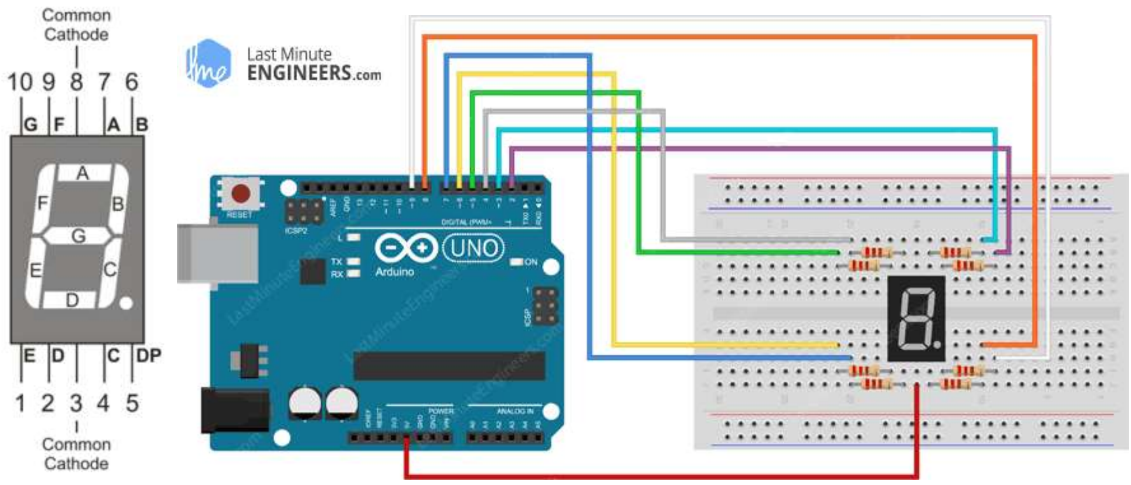
Mechatronics System Integration (MCTA3203)

Week **2**: Digital Logic System: Basic Logic Gates, Electronic Circuit Interfacing, Basic ALU, 7 segment display, ICs based interfacing application (ver. **2**).

Topics Covered:

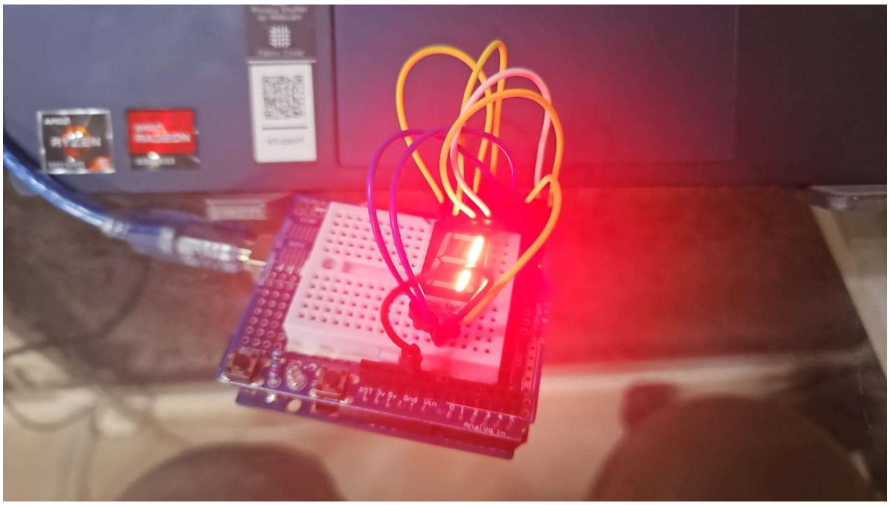
- Basic Logic Gates
- Electronic Circuit Interfacing
- Basic ALU
- 7-Segment Display
- IC-Based Interfacing Applications

Example: Interfacing a 7-segment display with Arduino. Numbering sequence from **0** to **5**.



```
// Define the pins for each segment (D0 to D6)
const int segmentA = 0; // D0
const int segmentB = 1; // D1
const int segmentC = 2; // D2
const int segmentD = 3; // D3
const int segmentE = 4; // D4
const int segmentF = 5; // D5
const int segmentG = 6; // D6

void setup() {
 // Initialize the digital pins as OUTPUTs
 pinMode(segmentA, OUTPUT);
 pinMode(segmentB, OUTPUT);
 pinMode(segmentC, OUTPUT);
 pinMode(segmentD, OUTPUT);
 pinMode(segmentE, OUTPUT);
 pinMode(segmentF, OUTPUT);
 pinMode(segmentG, OUTPUT);
}

// 0 = A,B,C,D,E,F
// 1 = B,C
// 2 = A,B,G,E,D
// 3 = A,B,C,D,G
// 4 = A,F,G,C,D

void loop() {
 // Turn on each segment one by one
 digitalWrite(segmentA, HIGH);
 digitalWrite(segmentB, HIGH);
 digitalWrite(segmentC, HIGH);
 digitalWrite(segmentD, HIGH);
 digitalWrite(segmentE, HIGH);
 digitalWrite(segmentF, HIGH);
```

```
  delay(500);
  digitalWrite(segmentA, LOW);
  digitalWrite(segmentB, LOW);
  digitalWrite(segmentC, LOW);
  digitalWrite(segmentD, LOW);
  digitalWrite(segmentE, LOW);
  digitalWrite(segmentF, LOW);

  delay(500);
  digitalWrite(segmentB, HIGH);
  digitalWrite(segmentC, HIGH);
  delay(500);
  digitalWrite(segmentB, LOW);
  digitalWrite(segmentC, LOW);

  delay(500);
  digitalWrite(segmentA, HIGH);
  digitalWrite(segmentB, HIGH);
  digitalWrite(segmentG, HIGH);
  digitalWrite(segmentE, HIGH);
  digitalWrite(segmentD, HIGH);
  delay(500);
  digitalWrite(segmentA, LOW);
  digitalWrite(segmentB, LOW);
  digitalWrite(segmentG, LOW);
  digitalWrite(segmentE, LOW);
  digitalWrite(segmentD, LOW);

  delay(500);
  digitalWrite(segmentA, HIGH);
  digitalWrite(segmentB, HIGH);
  digitalWrite(segmentC, HIGH);
  digitalWrite(segmentD, HIGH);
  digitalWrite(segmentG, HIGH);
  delay(500);
  digitalWrite(segmentA, LOW);
  digitalWrite(segmentB, LOW);
  digitalWrite(segmentC, LOW);
  digitalWrite(segmentD, LOW);
  digitalWrite(segmentG, LOW);

    delay(500);
  digitalWrite(segmentA, HIGH);
  digitalWrite(segmentF, HIGH);
  digitalWrite(segmentG, HIGH);
  digitalWrite(segmentC, HIGH);
  digitalWrite(segmentD, HIGH);
  delay(500);
  digitalWrite(segmentA, LOW);
  digitalWrite(segmentF, LOW);
  digitalWrite(segmentG, LOW);
  digitalWrite(segmentC, LOW);
  digitalWrite(segmentD, LOW);

  delay(500); // Delay for 1/2 second before repeating
}
```



Task

Materials Needed:

- Arduino Uno board
- Common cathode 7-segment display
- 220-ohm resistors (7 of them)
- Pushbuttons (2 or more)
- Jumper wires
- Breadboard

Circuit Setup:

1. Connect the common cathode 7-segment display to the Arduino Uno as follows:
   - Connect each of the 7 segments (a, b, c, d, e, f, g) of the display to separate digital pins on the Arduino (e.g., D0 to D6).
   - Connect the common cathode pin of the display to one of the GND (ground) pins on the Arduino.
   - Use 220-ohm resistors to connect each of the segment pins to the Arduino pins to limit the current.
2. Connect the pushbuttons to the Arduino:
   - Connect one leg of each pushbutton to a separate digital pin (e.g., D9 and D10) and connect the other leg of each pushbutton to GND.
   - Use 10K-ohm pull-up resistors for each pushbutton by connecting one end of each resistor to the digital pin and the other end to the 5V output of the Arduino.

Experiment Steps:

1. Build the circuit according to the circuit setup instructions.
2. Upload the provided Arduino code to your Arduino Uno.
3. Open the Serial Monitor in the Arduino IDE.
4. Press the increment button to increase the count. The 7-segment display should show the numbers from 0 to 9 sequentially.
5. Press the reset button to reset the count to 0.

This simple experiment helps you understand how to interface a 7-segment display with an Arduino Uno and how to control it manually using buttons. You can extend this experiment by adding more functionality or additional displays for more complex projects.

Questions:

How can you interface an I2C LCD with Arduino? Explain the coding principle behind it compared to a 7-segment display and a matrix LED.