MECHATRONICS SYSTEM INTEGRATION

MCTA 3203

LAB 02:

DIGITAL LOGIC SYSTEM

SECTION 1

SEMESTER 1, 2025/2026

INSTRUCTOR:

ASSOC. PROF. EUR. ING. IR. TS. GS. INV. DR. ZULKIFLI BIN ZAINAL ABIDIN

DATE OF EXPERIMENT: 15 OCTOBER 2025

DATE OF SUBMISSION: 21 OCTOBER 2025

GROUP 3

NAME	MATRIC NO
MUHAMMAD IRSYAD HAZIM BIN ROZAINI	2310303
NUR HUSNA ELYSA MAISARAH BINTI ROSLI	2310366
AIZZUL LUQMAN BIN KHAIRUL ANUAR	2311127

ABSTRACT

This experiment focuses on understanding the fundamental principles of digital logic systems and their application in mechatronic system integration. The study emphasizes the use of basic logic gates, arithmetic logic units (ALU), electronic circuit interfacing, and display modules such as the 7-segment display. Through practical implementation using the Arduino Uno microcontroller, the experiment demonstrates the interfacing and control of a 7-segment display, simulating numeric counting from 0 to 9 using push buttons. The activity reinforces the importance of digital logic concepts, signal control, and hardware-software integration in mechatronic system design.

1.0 INTRODUCTION

The Digital Logic System module introduces students to the foundational concepts of digital electronics essential for mechatronic system development. This includes the study of basic logic gates (AND, OR, NOT, NAND, NOR, XOR), digital interfacing techniques, and arithmetic operations using basic ALU circuits. In modern mechatronic applications, digital logic enables the control and coordination of sensors, actuators, and microcontrollers to perform intelligent operations.

In this experiment, the focus is on interfacing a 7-segment display with an Arduino Uno to display numerical sequences. The experiment involves connecting each segment to the Arduino's digital output pins and using push buttons to increment or reset the count. The hands-on exercise enhances understanding of logic-level signal control, current-limiting resistors, and the role of programming in digital hardware control. The project also serves as an introduction to more complex interfacing systems such as I2C LCDs and matrix LEDs, highlighting the transition from basic to advanced digital display technologies.

2.0 MATERIALS AND EQUIPMENTS

Here are the list of all equipments used in the experiment:

1. 1 Arduino Uno board
2. 1 Common cathode 7-segment display
3. 8 220-ohm resistors
4. 2 10k-ohm resistors
5. 2 Pushbuttons
6. Jumper wires
7. 1 Breadboard
8. USB Cable

3.0 EXPERIMENTAL SETUP

1. The common cathode 7-segment display was connected to the Arduino Uno as follows:

- Each of the 7 segments (a, b, c, d, e, f, g) of the display was connected to separate digital pins on the Arduino (D0 to D6).
- The common cathode pin of the display was connected to one of the GND (ground) pins on the Arduino.
- The 220-ohm resistors were used to connect each of the segment pins to the Arduino pins to limit the current.

2. The pushbuttons were connected to the Arduino:

- One leg of each pushbutton was connected to a separate digital pin (D10 and D12) and the other leg of each pushbutton was connected to the GND (ground) pin.
- The 10K-ohm pull-up resistors were used 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.

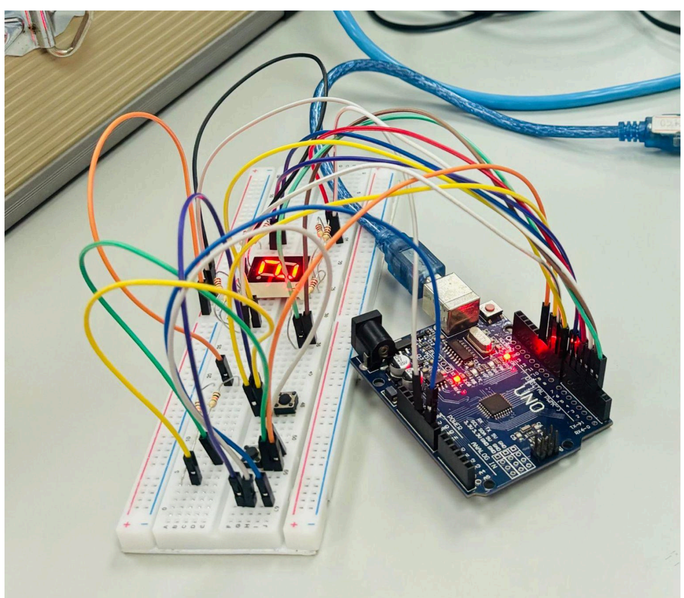

Figure 3.1 Experimental Circuit Setup

4.0 METHODOLOGY

1. **Circuit Construction:**

   The circuit was built on a breadboard following the connection setup described in the experimental setup. All segment pins (a–g) were connected through 220 Ω resistors to the Arduino pins (D0–D6). Push buttons were connected to D10 and D12 with 10 kΩ pull-up resistors. Resistors were used to protect each LED segment from excessive current.

2. **Programming**:

   The Arduino IDE was used to upload a C++ program that controls the display. The program defines digital pins D0–D6 for segments A–G and configures them as outputs. Push buttons were configured as inputs with pull-up resistors. The program used conditional statements to detect button presses and control which segments of the display were turned on to represent digits 0–9.

3. **Operation:**

   - After uploading the code, the system was powered via the USB connection.
   - When the increment button is pressed, the program increases the count variable by one and updates the 7-segment display accordingly.
   - When the reset button is pressed, the count variable resets to zero, and the display shows “0”.

4. **Observation:**

   Upon pressing the increment button repeatedly, the 7-segment display cycles through digits 0–9 in sequence. Pressing the reset button immediately resets the display to “0”.

5. **Code Behavior:**

   The program uses a series of digitalWrite() commands to turn on and off specific segments representing each digit. A short delay is included between transitions for visual clarity. Shown below are the programming codes used in the experiment using Arduino IDE:

File Edit Sketch Tools Help

PushButton_Lab2

```
// Segment pins
const int segmentA = 2; // D0
const int segmentB = 3; // D1
const int segmentC = 4; // D2
const int segmentD = 5; // D3
const int segmentE = 6; // D4
const int segmentF = 7; // D5
const int segmentG = 8; // D6

// Button pins
const int buttonInc = 10;   // Increment
const int buttonReset = 12; // Reset

int number = 0;
int lastInc = HIGH;
int lastReset = HIGH;

void setup() {
  pinMode(segmentA, OUTPUT);
  pinMode(segmentB, OUTPUT);
  pinMode(segmentC, OUTPUT);
  pinMode(segmentD, OUTPUT);
  pinMode(segmentE, OUTPUT);
  pinMode(segmentF, OUTPUT);
  pinMode(segmentG, OUTPUT);

  pinMode(buttonInc, INPUT_PULLUP);
  pinMode(buttonReset, INPUT_PULLUP);

  displayNumber(number); // start at 0
}
```

```
void loop() {
  int curInc = digitalRead(buttonInc);
  int curReset = digitalRead(buttonReset);

  // Increment button pressed
  if (lastInc == HIGH && curInc == LOW) {
    number++;
    if (number > 9) number = 0;
    displayNumber(number);
    delay(200); // debounce
  }

  // Reset button pressed
  if (lastReset == HIGH && curReset == LOW) {
    number = 0;
    displayNumber(number);
    delay(200); // debounce
  }

  lastInc = curInc;
  lastReset = curReset;
}

// Display function using your original patterns
void displayNumber(int n) {
  switch (n) {
    case 0:
      digitalWrite(segmentA, LOW);
      digitalWrite(segmentB, LOW);
      digitalWrite(segmentC, LOW);
      digitalWrite(segmentD, HIGH);
      digitalWrite(segmentE, LOW);
```

```
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, LOW);
    break;

  case 1:
    digitalWrite(segmentA, LOW);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, HIGH);
    digitalWrite(segmentE, HIGH);
    digitalWrite(segmentF, HIGH);
    digitalWrite(segmentG, LOW);
    break;

  case 2:
    digitalWrite(segmentA, LOW);
    digitalWrite(segmentB, LOW);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, LOW);
    digitalWrite(segmentE, LOW);
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, HIGH);
    break;

  case 3:
    digitalWrite(segmentA, LOW);
    digitalWrite(segmentB, LOW);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, LOW);
    digitalWrite(segmentE, HIGH);
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, LOW);



    break;

  case 4:
    digitalWrite(segmentA, LOW);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, LOW);
    digitalWrite(segmentD, LOW);
    digitalWrite(segmentE, HIGH);
    digitalWrite(segmentF, HIGH);
    digitalWrite(segmentG, LOW);
    break;

  case 5:
    digitalWrite(segmentA, HIGH);
    digitalWrite(segmentB, LOW);
    digitalWrite(segmentC, LOW);
    digitalWrite(segmentD, LOW);
    digitalWrite(segmentE, HIGH);
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, LOW);
    break;

  case 6:
    digitalWrite(segmentA, HIGH);
    digitalWrite(segmentB, LOW);
    digitalWrite(segmentC, LOW);
    digitalWrite(segmentD, LOW);
    digitalWrite(segmentE, LOW);
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, LOW);
    break;
```

```
    case 7:
      digitalWrite(segmentA, LOW);
      digitalWrite(segmentB, LOW);
      digitalWrite(segmentC, HIGH);
      digitalWrite(segmentD, HIGH);
      digitalWrite(segmentE, HIGH);
      digitalWrite(segmentF, HIGH);
      digitalWrite(segmentG, LOW);
      break;

    case 8:
      digitalWrite(segmentA, LOW);
      digitalWrite(segmentB, LOW);
      digitalWrite(segmentC, LOW);
      digitalWrite(segmentD, LOW);
      digitalWrite(segmentE, LOW);
      digitalWrite(segmentF, LOW);
      digitalWrite(segmentG, LOW);
      break;

    case 9:
      digitalWrite(segmentA, LOW);
      digitalWrite(segmentB, LOW);
      digitalWrite(segmentC, LOW);
      digitalWrite(segmentD, LOW);
      digitalWrite(segmentE, HIGH);
      digitalWrite(segmentF, LOW);
      digitalWrite(segmentG, LOW);
      break;
  }
}
```

5.0 DATA COLLECTION

- **Description of Data and Instrument**

The experiment was conducted using an Arduino Uno microcontroller, a common cathode 7-segment display, two push buttons (increment and reset), and 220Ω resistors. The Arduino board served as the main data acquisition and control device, sending digital signals to each segment (a–g) of the display. The push buttons acted as manual inputs to increase or reset the counter.

- **Observation Data**

During the experiment, data was recorded based on the display response to button presses. The expected and observed outcomes are shown below.

Button Presses	Expected Display	Observed Display
-	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	0	0
Reset	0	0

6.0 DATA ANALYSIS

The data collected was analyzed to verify the logical consistency of the Arduino code with the display outputs. Each number (0–9) corresponds to a unique binary pattern that activates specific segments (a–g).

Decimal	Binary Input	Segment ON (a-g)
0	0000	a,b,c,e,f,g
1	0001	a,g
2	0010	a,b,d,e,f
3	0011	a,b,d,f,g
4	0100	a,c,d,g
5	0101	b,c,d,f,g
6	0110	b,c,d,e,f,g
7	0111	a,b,g
8	1000	a,b,c,d,e,f,g
9	1001	a,b,c,d,f,g

The results shows that:

- ➢ The Arduino successfully implemented digital logic functions to represent decimal digits.
- ➢ The push-button interface allowed manual binary incrementation and reset behavior.
- ➢ The experiment reflects how combinational logic can be implemented in hardware systems using microcontrollers.

7.0 RESULTS

The experiment successfully achieved its objectives:

- ➢ The 7-segment LED display correctly displayed digits 0–9 in sequence.
- ➢ Each button press resulted in accurate increment or reset behavior.
- ➢ All seven segments functioned as intended, with consistent illumination.
- ➢ The data validated the correctness of the Arduino logic design.

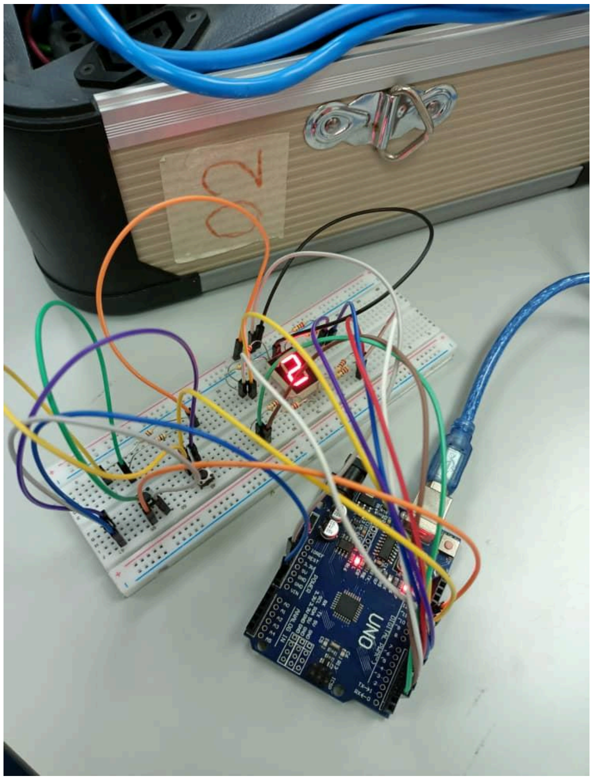

Figure 7.1 Circuit Setup

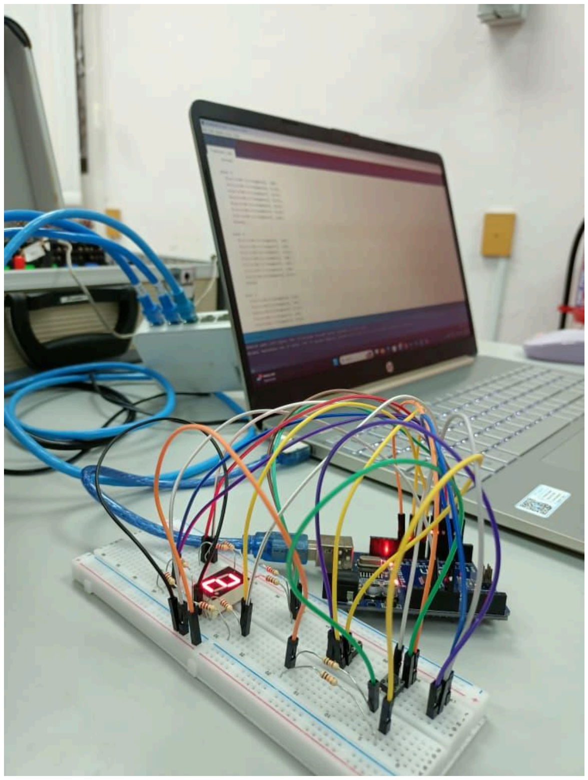

Figure 7.2 Code uploaded to hardware

8.0 DISCUSSION

The experiment successfully demonstrated the interfacing of a 7-segment display with an Arduino Uno to display digits from 0 to 9. The system responded as intended when the increment button was pressed, allowing the digits to advance sequentially, while the reset button was used to return the display to zero. This confirmed that both the hardware setup and the Arduino program were functioning effectively to achieve digital counting control. The experiment provided practical understanding of digital logic applications, electronic interfacing, and the relationship between microcontroller programming and physical circuit behavior. It also showed how simple control logic can be extended to more complex systems such as digital counters and timing devices.

However, several discrepancies were observed during the experiment. Initially, the circuit did not function as intended due to variations in the pin configuration of the 7-segment display. The segment arrangement (a–g) differed from the assumed wiring in the original Arduino code, which caused incorrect numbers or incomplete patterns to appear on the display. The issue was resolved by identifying the correct pin configuration and adjusting the connections and the coding accordingly. Another problem encountered was with the reset push button, which initially failed to function properly. After troubleshooting, it was discovered that the component itself was faulty. Replacing the push button with a new one restored full functionality, and the reset feature worked as intended thereafter.

Some limitations and sources of error were also noted throughout the experiment. The use of mechanical push buttons introduced switch bounce, causing multiple unintended increments or resets when pressed rapidly. Additionally, loose jumper connections on the breadboard sometimes led to inconsistent illumination or partial lighting of segments. Variations in resistor tolerances and contact resistance could also result in uneven brightness among the display segments. Furthermore, minor timing inconsistencies in the program's delay functions might have slightly affected the responsiveness of the display updates. Despite these limitations, the overall performance of the circuit was satisfactory, and the experiment successfully met its objectives by demonstrating the fundamental principles of digital interfacing, troubleshooting, and system correction in a practical setting.

9.0 CONCLUSION

The experiment successfully demonstrated the practical application of digital logic principles through the control of a 7-segment display using an Arduino microcontroller. By building and programming the circuit, students gained a deeper understanding of logic gate functions, digital output manipulation, and sequential control systems. The results showed that digital logic serves as the core foundation for data representation and hardware interfacing in mechatronic systems. Additionally, the activity fostered skills in troubleshooting, coding, and component connection—key competencies in embedded system development.

10.0 RECOMMENDATIONS

1. Enhance the Experiment: Extend the project to include dual or multiplexed 7-segment displays to show multi-digit numbers or sensor readings.

2. Incorporate I2C or SPI Displays: Implement I2C LCDs or LED matrix displays to explore more efficient communication protocols and complex visual outputs.

3. Introduce Logic Simulation Software: Before hardware implementation, use simulation tools (e.g., Proteus or TinkerCAD) to visualize circuit behavior and minimize wiring errors.

4. Integrate Sensor Input: Connect a digital sensor (e.g., temperature or proximity) to display real-time data on the 7-segment display to strengthen system integration skills.

5. Focus on Optimization: Encourage students to use modular coding techniques and power-efficient circuit designs for practical mechatronic applications.

11.0 REFERENCES

Hobby, S. 2025, *How does a 7 segment display work | Arduino with 7 segment display | Step by Step instructions*. SriTu Hobby. https://srituhobby.com/how-does-a-7-segment-display-work-arduino-with-7-segment-display-step-by-step-instructions/

SriTu Hobby. 2020, *SEGMENT DISPLAY ARDUINO tutorial | seven SEGMENT DISPLAY ARDUINO UNO [code and circuit diagram]* [Video]. YouTube. https://www.youtube.com/watch?v=wlwNwWaVT70

Tec4electronics. 2020, *How to use 7 segment Display with Arduino Uno | Interfacing | Coding* [Video]. YouTube. https://www.youtube.com/watch?v=ONpQdgXUjng

LearnElectronics India. 2020, *Interfacing a 7 Segment Display with Arduino | How to create a simple counter | Countdown display* [Video]. YouTube. https://www.youtube.com/watch?v=DZUlLy-NC-c

APPENDICES

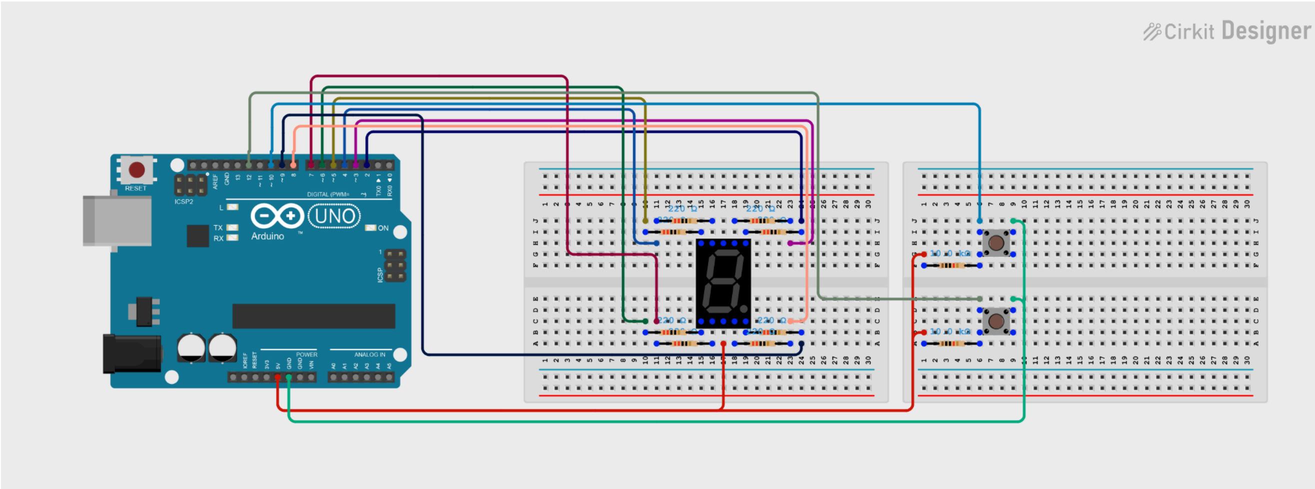

Figure 12.1 Circuit Design Schematics

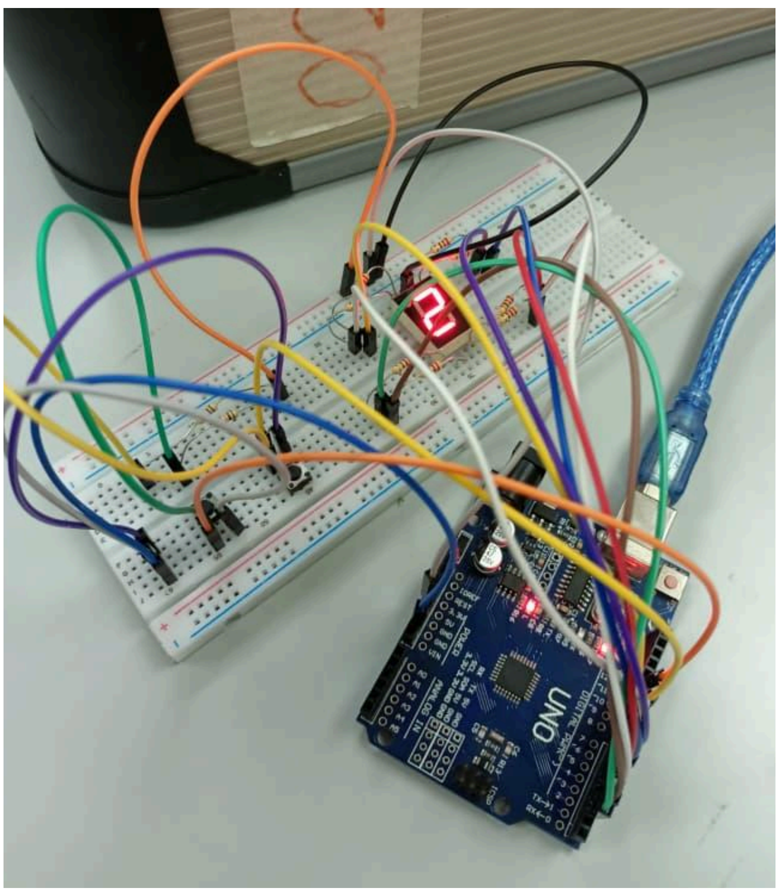

Figure 12.2 Real-life Circuit Design

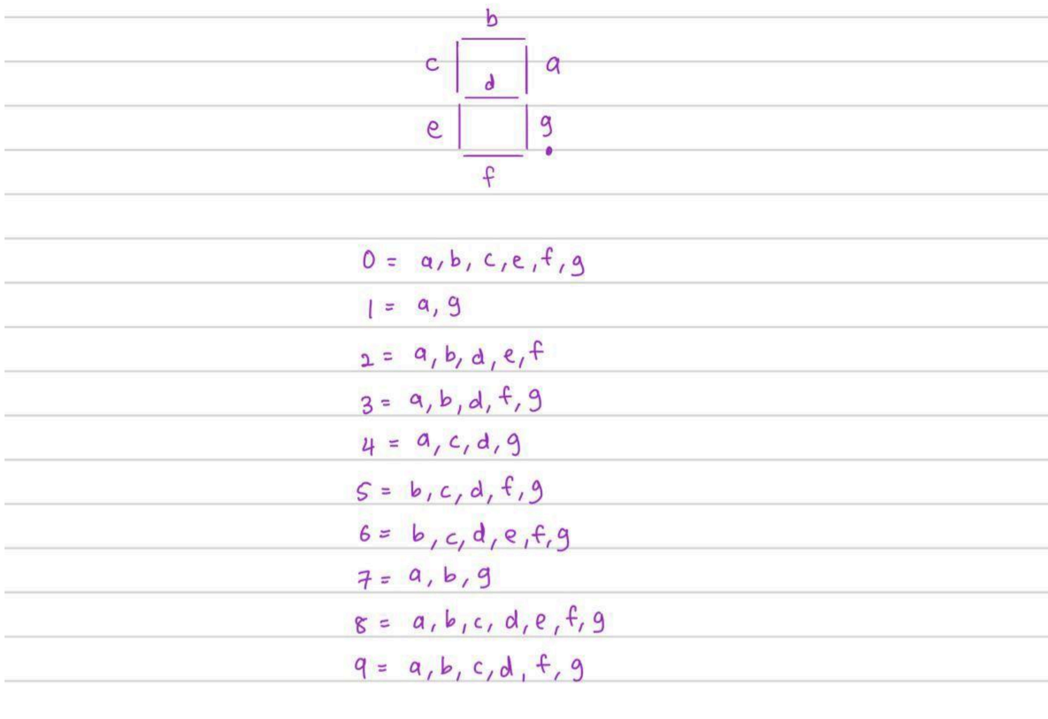

Figure 12.3 7-Segment LED Display Configuration

ACKNOWLEDGEMENTS

The group would like to express our deepest appreciation to our laboratory instructor, Assoc. Prof. Eur. Ing. Ir. Ts. Gs. Inv. Dr. Zulkifli bin Zainal Abidin for the continuous guidance, encouragement, and clear explanations provided throughout the course of this experiment. His expertise and detailed feedback greatly enhanced our understanding of digital logic systems and circuit interfacing, especially in applying theoretical knowledge to practical implementation. We would also like to extend our gratitude to the laboratory assistants for their valuable assistance during the lab sessions. Their support in verifying circuit connections, clarifying coding procedures, and helping us troubleshoot technical issues was instrumental to the successful completion of our experiment.

In addition, we would like to thank our classmates and also group members, Muhammad Irsyad Hazim bin Rozaini, Nur Husna Elysa Maisarah binti Rosli and Aizzul Luqman bin Khairul Anuar for their cooperation and willingness to share insights, suggestions, and resources during the experiment. Collaborative discussions and teamwork played a crucial role in identifying and resolving problems efficiently. Lastly, we appreciate the opportunity provided by the Department of Mechatronics Engineering to conduct this experiment, which allowed us to strengthen our practical skills in Arduino programming, electronic interfacing, and logical circuit design. The collective guidance and support from all parties involved contributed significantly to the success and learning outcome of this project.

STUDENT'S DECLARATION

<u>Certificate of Originality and Authenticity</u>

This is to certify that we are **responsible** for the work submitted in this report, that **the original work** is our own except as specified in the references and acknowledgement, and that the original work contained herein have not been untaken or done by unspecified sources or persons.

We hereby certify that this report has **not been done by only one individual** and **all of us have contributed to the report.** The length of contribution to the reports by each individual is noted within this certificate.

We also hereby certify that we have **read** and **understand** the content of the total report and no further improvement on the reports is needed from any of the individual's contributors to the report.

We therefore, agreed unanimously that this report shall be submitted for **marking** and this **final printed report** has been **verified by us.**

Signature: Read [/]
Name: Muhammad Irsyad Hazim bin Rozaini Understand [/]
Matric Number: 2310303 Agree [/]
Contribution: Data Collection, Data Analysis, Results and Appendices

Signature: Read [/]
Name: Nur Husna Elysa Maisarah binti Rosli Understand [/]
Matric Number: 2310366 Agree [/]
Contribution: Equipment, Experimental Setup, Methodology, Discussion and Acknowledgement

Signature:

Name: Aizzul Luqman bin Khairul Anuar

Matric Number: 2311127

Contribution: Abstract, Introduction, Conclusion and Recommendations

Read [/]

Understand [/]

Agree [/]