MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 2 2024/2025

WEEK 2: DIGITAL LOGIC DESIGN

SECTION 1

GROUP 3

LECTURER: ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO

NO.	GROUP MEMBERS	MATRIC NO.
1.	AHMAD DARWISH BIN AHMAD TERMIZI	2212089
2.	MUHAMMAD AQIF BIN ROSZAHAN	2210345
3.	AKMAL ZARIF BIN NAJIB	2211971
4.	SIDDIQUI HAMMAD AHMED	2221201

Date of Experiment: Monday, 10 March 2025

Date of Submission: Monday, 17 March 2025

ABSTRACT

This experiment demonstrates the interfacing of a common cathode 7-segment display with an Arduino Uno to display numbers from 0 to 9 using digital logic. Pushbuttons allow manual control for incrementing and resetting the display. The experiment covers digital circuit interfacing, microcontroller programming, and resistor usage for current limiting. Additionally, it compares 7-segment displays, I2C LCDs, and LED matrix displays in terms of coding and efficiency. This hands-on approach reinforces key concepts in embedded systems and display control.

TABLE OF CONTENTS

INTRODUCTION

This experiment focuses on interfacing a common cathode 7-segment display with an Arduino Uno to display numbers from 0 to 9 using digital logic. Pushbuttons are used to manually control the count, allowing for incrementing and resetting the display. The experiment covers key concepts in digital logic systems, such as basic logic gates, electronic circuit interfacing, and IC-based applications. Students will learn how to configure Arduino digital output pins, use resistors for current limiting, and write code to control the display. Additionally, the lab introduces a comparison between 7-segment displays, I2C LCDs, and LED matrix displays, highlighting their differences in coding and applications. This hands-on experiment helps students understand microcontroller programming, circuit design, and display interfacing in embedded systems.

MATERIALS AND EQUIPMENTS

1. Arduino Mega 2560
2. Breadboard
3. 7 Segment Display
4. Pushbutton
5. Jumpers
6. Resistors

EXPERIMENTAL SETUP

1. Connect each of the seven segments (a, b, c, d, e, f, g) of the 7-segment display to separate digital pins on the Arduino.
2. Attach the common anode pin of the display to the 3.3V pin on the Arduino.
3. For the pushbutton, connect one of its legs to a digital pin on the Arduino, another to the ground (GND) pin, and a third to the 5V pin through a resistor, as illustrated in Figure 1.

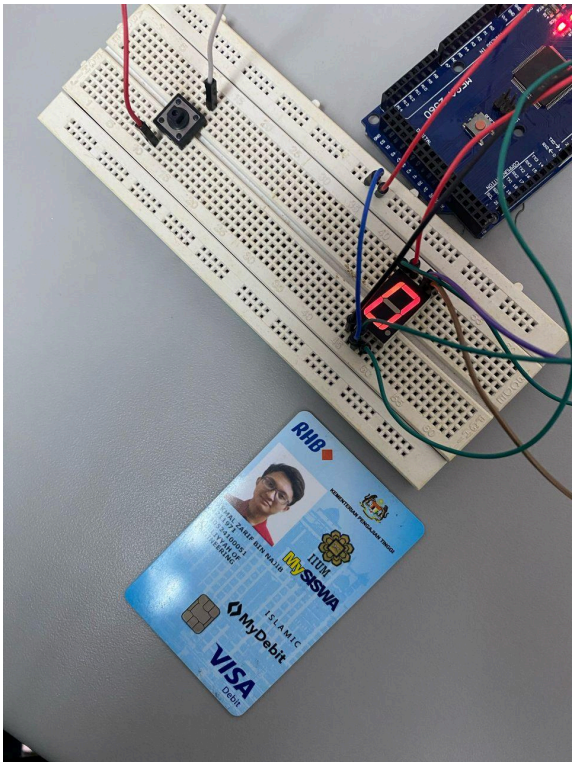


Figure 1: Hardware setup

METHODOLOGY

1. Setup the Arduino Mega 2560
2. Setup the code in Arduino IDE
3. Test the code
4. Code Snippet

Arduino

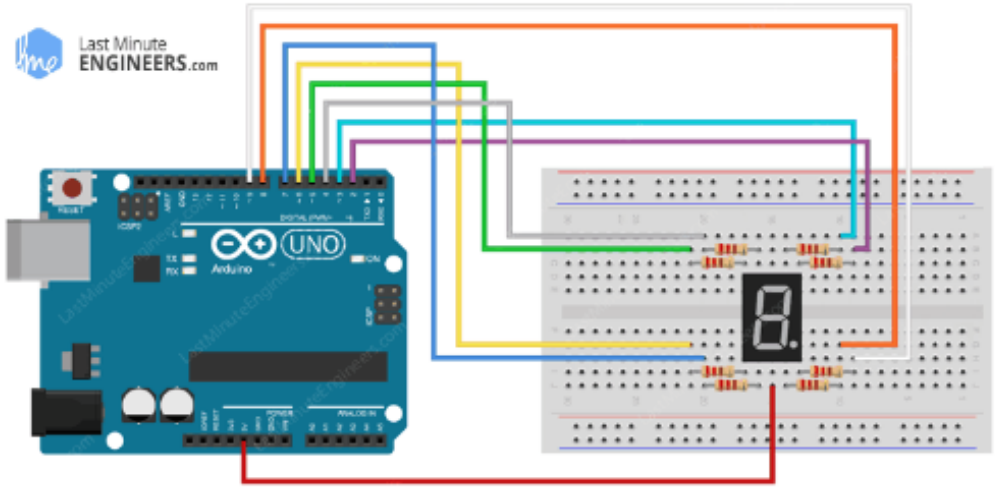
```
// Define the pins for each segment (D0 to D6)
const int segmentA = 3; // D0
const int segmentB = 2; // D1
const int segmentC = 8; // D2
const int segmentD = 7; // D3
const int segmentE = 6; // D4
const int segmentF = 4; // D5
const int segmentG = 10; // D6
int i = 0; //as counter

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);
  pinMode(12, INPUT); //BUTTON
  Serial.begin(9600);
}
/*
 0 = A,B,C,D,E,F
 1 = B,C
 2 = A,B,G,E,D
 3 = A,B,C,D,G
 4 = F,G,B,C
 5 = A,F,G,C,D
 6 = A,F,G,E,D,C
 7 = A,B,C
 8 = A,B,C,D,E,F,G
 9 = A,B,C,D,F,G
 */
void loop() {
   if(i==0){
// turn on the display according to the counter
 //0
  digitalWrite(segmentA,  LOW);
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentC,  LOW);
  digitalWrite(segmentD,  LOW);
  digitalWrite(segmentE,  LOW);
  digitalWrite(segmentF,  LOW);
  digitalWrite(segmentG,  HIGH);

 }
 else if(i==1){
  //1
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentC,  LOW);
  digitalWrite(segmentA,  HIGH);
  digitalWrite(segmentD,  HIGH);
  digitalWrite(segmentE,  HIGH);
  digitalWrite(segmentF,  HIGH);
  digitalWrite(segmentG,  HIGH);


 }
 else if(i==2){
  //2
  digitalWrite(segmentA,  LOW);
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentG,  LOW);
  digitalWrite(segmentE,  LOW);
  digitalWrite(segmentD,  LOW);
  digitalWrite(segmentF,  HIGH);
  digitalWrite(segmentC,  HIGH);

 }
 else if(i==3){
  //3
  digitalWrite(segmentA,  LOW);
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentC,  LOW);
  digitalWrite(segmentD,  LOW);
```

```
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentE,  HIGH);
 digitalWrite(segmentF,  HIGH);

}
else if(i==4){
 //4
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentB,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentA,  HIGH);
 digitalWrite(segmentE,  HIGH);
 digitalWrite(segmentD,  HIGH);

}
else if(i==5){
 //5
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentB,  HIGH);
 digitalWrite(segmentE,  HIGH);

}
else if(i==6){
 // 6
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentE,  LOW);
 digitalWrite(segmentB,  HIGH);

}
else if(i==7){
 // 7
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentB,  LOW);
 digitalWrite(segmentD,  HIGH);
 digitalWrite(segmentE,  HIGH);
 digitalWrite(segmentF,  HIGH);
 digitalWrite(segmentG,  HIGH);

}
else if(i==8){
 // 8
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentB,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentE,  LOW);
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);

}
else if(i==9){
 // 9
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentB,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentE,  HIGH);

}

 if(digitalRead(12)==1)
 {
   while(digitalRead(12)==1)
   {
    Serial.println(i);
   }
   i++; //increase counter with each push
button
 }
   //start count back to 0
 if(i == 10)
 {
   i = 0;
 }
 delay(1000);
}
```

DATA COLLECTION

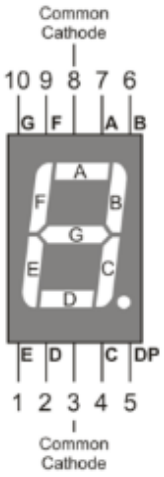
i) Figure: Circuit Diagram

Figure shows how the components were connected.

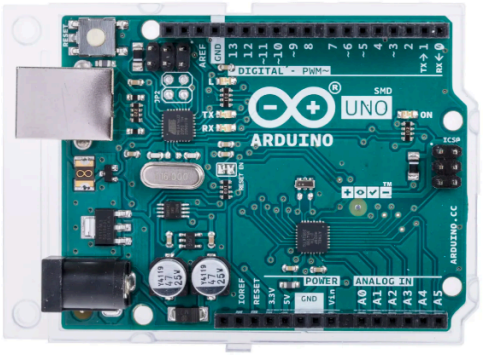


ii) Instruments

a. Common cathode 7-segment display



b. Arduino Mega 2560



c. Resistors

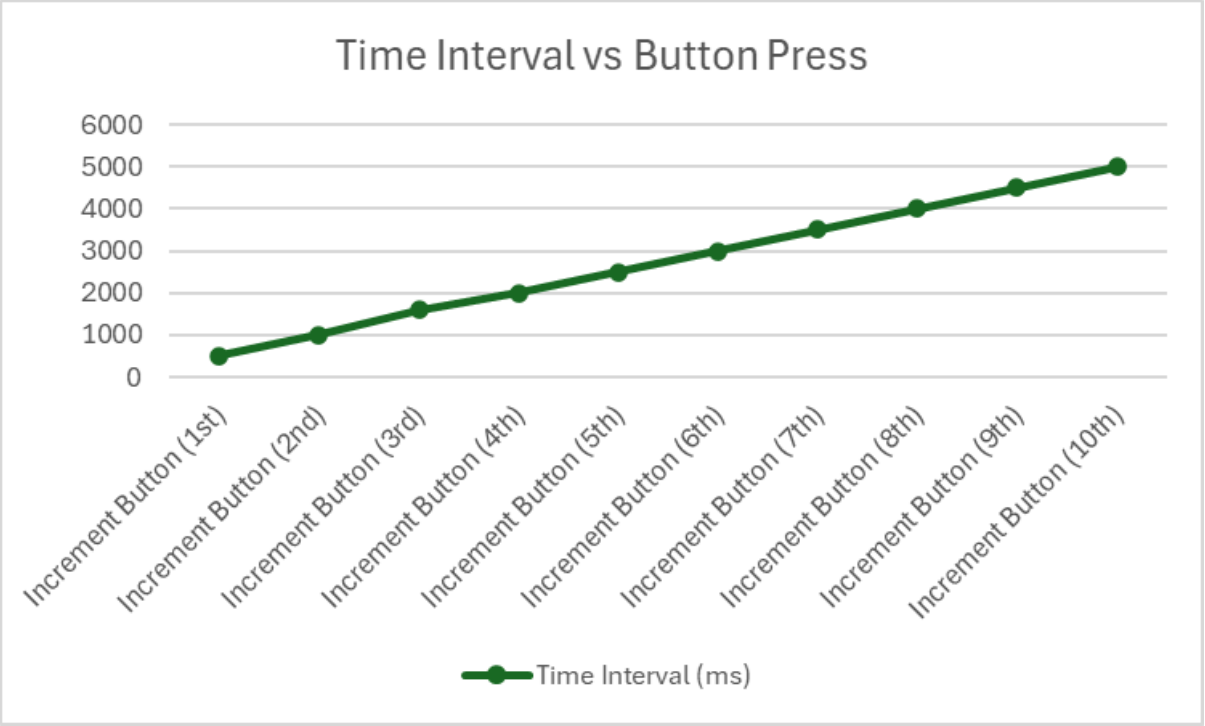
d. Pushbutton

e. Jumper wires
f. Breadboard

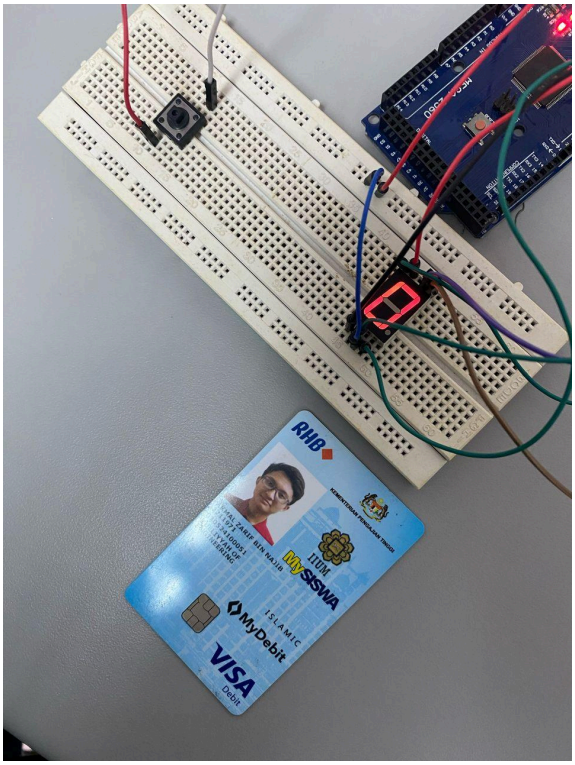
Table 1 of expected and actual display output of the 7-segment display with the time interval.

Button Press	Expected Display Output	Actual Display Output	Time Interval (ms)
Increment Button (1st)	1	1	500
Increment Button (2nd)	2	2	1000
Increment Button (3rd)	3	3	1600
Increment Button (4th)	4	4	2000
Increment Button (5th)	5	5	2500
Increment Button (6th)	6	6	3000
Increment Button (7th)	7	7	3500
Increment Button (8th)	8	8	4000
Increment Button (9th)	9	9	4500
Increment Button (10th)	0	0	5000

DATA ANALYSIS



RESULT



QUESTION

1) **How to interface an I2C LCD with Arduino? Explain the coding principle behind it compared with 7 segments display and matrix LED.**

Among the various display technologies available, I2C LCDs, 7-segment displays, and LED matrices are commonly used for different applications. Each type has its own advantages and is suited for specific use cases. Attach the Arduino board and I2C LCD. In general, connect the LCD's SDA pin to the Arduino's A4 pin and the SCL pin to the A5 pin; make sure the LCD VCC and GND pins are linked to the appropriate power and ground pins on the Arduino. Due to coding principles, each segment of the 7-segment display is controlled independently by connecting it directly to a specific pin on the Arduino. To display a digit, the corresponding pin must be set to HIGH or LOW, depending on the segment you want to light up. However, in the case of matrix LEDs, they are arranged in rows and columns. By Setting the matching row pin to HIGH and column pin to LOW, a given LED can be turned on. Multiplexing is necessary to quickly cycle between rows and columns to give the appearance of a continuous display. On The other hand, regardless of the size of the display, the 12C LCD requires only two pins(SDA and SCL)for communication. This reduces the number of pins required by the Arduino and simplifies wiring.

DISCUSSION

- **Software**

  In this experiment, Arduino software is used to program the microcontroller. Arduino offers an open-source platform, which makes it easy to write, build and upload codes to Arduino boards. It has a user-friendly interface which features a text editor, a toolbar with common actions, and various menus providing access to examples, libraries, and tools. To manage the behaviour of the microcontroller, the C/C++ programming language is used. Attached below is the coding of the project :

```
// Define the pins for each segment (D0 to D6)
const int segmentA = 3; // D0
const int segmentB = 2; // D1
const int segmentC = 8; // D2
const int segmentD = 7; // D3
const int segmentE = 6; // D4
const int segmentF = 4; // D5
const int segmentG = 10; // D6
int i = 0; //as counter

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);
  pinMode(12, INPUT); //BUTTON
  Serial.begin(9600);
}
/*
 0 = A,B,C,D,E,F
 1 = B,C
 2 = A,B,G,E,D
 3 = A,B,C,D,G
 4 = F,G,B,C
 5 = A,F,G,C,D
 6 = A,F,G,E,D,C
 7 = A,B,C
 8 = A,B,C,D,E,F,G
 9 = A,B,C,D,F,G
 */
void loop() {
   if(i==0){
// turn on the display according to the counter
 //0
  digitalWrite(segmentA,  LOW);
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentC,  LOW);
  digitalWrite(segmentD,  LOW);
  digitalWrite(segmentE,  LOW);
  digitalWrite(segmentF,  LOW);
  digitalWrite(segmentG,  HIGH);

 }
 else if(i==1){
  //1
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentC,  LOW);
  digitalWrite(segmentA,  HIGH);
  digitalWrite(segmentD,  HIGH);
  digitalWrite(segmentE,  HIGH);
  digitalWrite(segmentF,  HIGH);
  digitalWrite(segmentG,  HIGH);


 }
 else if(i==2){
  //2
  digitalWrite(segmentA,  LOW);
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentG,  LOW);
  digitalWrite(segmentE,  LOW);
  digitalWrite(segmentD,  LOW);
  digitalWrite(segmentF,  HIGH);
  digitalWrite(segmentC,  HIGH);

 }
 else if(i==3){
  //3
  digitalWrite(segmentA,  LOW);
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentC,  LOW);
  digitalWrite(segmentD,  LOW);
  digitalWrite(segmentG,  LOW);
  digitalWrite(segmentE,  HIGH);
  digitalWrite(segmentF,  HIGH);

 }
 else if(i==4){
  //4
  digitalWrite(segmentF,  LOW);
  digitalWrite(segmentG,  LOW);
  digitalWrite(segmentB,  LOW);
  digitalWrite(segmentC,  LOW);
  digitalWrite(segmentA,  HIGH);
  digitalWrite(segmentE,  HIGH);
  digitalWrite(segmentD,  HIGH);

 }
 else if(i==5){
  //5
  digitalWrite(segmentA,  LOW);
  digitalWrite(segmentF,  LOW);
```

```
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentB,  HIGH);
 digitalWrite(segmentE,  HIGH);

}
else if(i==6){
 // 6
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentE,  LOW);
 digitalWrite(segmentB,  HIGH);

}
else if(i==7){
 // 7
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentB,  LOW);
 digitalWrite(segmentD,  HIGH);
 digitalWrite(segmentE,  HIGH);
 digitalWrite(segmentF,  HIGH);
 digitalWrite(segmentG,  HIGH);

}
else if(i==8){
 // 8
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentB,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentE,  LOW);
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);

}
else if(i==9){
 // 9
 digitalWrite(segmentA,  LOW);
 digitalWrite(segmentB,  LOW);
 digitalWrite(segmentC,  LOW);
 digitalWrite(segmentD,  LOW);
 digitalWrite(segmentF,  LOW);
 digitalWrite(segmentG,  LOW);
 digitalWrite(segmentE,  HIGH);

}

 if(digitalRead(12)==1)
 {
  while(digitalRead(12)==1)
  {
   Serial.println(i);
  }
  i++; //increase counter with each push
button
 }
  //start count back to 0
 if(i == 10)
 {
  i = 0;
 }
 delay(1000);
}
```

- **Electrical**

The key electrical components utilised in this project are the 7-segment display and the push button, which are connected to the Arduino Mega 2560. Each segment of the display labelled A, B, C, D, E, F, G, and DP is connected to a specific PWM (Pulse Width Modulation) pin on the Arduino Mega. This pin allocation enables individual control of each segment. The connections are as follows :

Segment A = PWM 3

Segment B = PWM 2

Segment C = PWM 8

Segment D = PWM 7

Segment E = PWM 6

Segment F = PWM 4

Segment G = PWM 10

Moreover, the push button is connected to PWM pin 12 and GND (ground) on the Arduino Mega. This button acts as an input mechanism as it allows the user to interact with the project. By pressing the button, users can trigger specific actions programmed into the microcontroller. In this project, pressing the push button will trigger each segment in the 7-segment display accordingly.

- **Hardware**

  Using a seven-segment display, a push button, and an Arduino Mega 2560 microcontroller board, the digital counter system is constructed. By increasing the displayed number with each push button press, the Arduino manages the counting process. Male-to-male jumper wires on a breadboard are used in the circuit to link the components, ensuring precise wiring for optimum performance. To connect to the push button for input detection and the 7-segment display for number display, the operating logic relies on the Arduino's digital input/output pins. With the exception of the 7-segment display, the circuit demonstrated stability throughout the testing, with the push button and jumper wires operating dependably.

CONCLUSION

In conclusion, a 7-segment display was successfully interfaced with an Arduino board to show values ranging from 0 to 9. The Arduino controlled each of the separate segments that were shown, confirming the experiment's electrical and programming components. There are no flickering issues or wiring issues with the arrangement. The digital logic system principles

mentioned at the beginning of this article are accurately shown by this experiment. It strengthens our comprehension of hardware interfacing and offers helpful advice on creating an electrical display that is easy to use. Further enhancements could be explored to help make the system more effective.

RECOMMENDATIONS

Several modifications can be made to improve the performance of this 7-segment display system. The previously functional system would take on a new dynamic with the addition of other input devices, like a potentiometer. The segment display's LED brightness can be changed by turning the shaft of this variable resistor, which offers a changeable level of resistance. Pulse-width modulation (PWM) could potentially be used to modify the duty cycle and, consequently, the LED's brightness. Last but not least, adding error detection to the code may increase system dependability for practical uses.

REFERENCES

“Arduino Integrated Development Environment (IDE) v1 | Arduino Documentation | Arduino Documentation,” *Arduino.cc*, 2022. https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/

ACKNOWLEDGEMENTS

Special thanks to ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO for their guidance and support during this experiment.

Certificate of Originality and Authenticity

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: darwish — Read [/]
Name: AHMAD DARWISH BIN AHMAD TERMIZI — Understand [/]
Matric Number: 2212089 — Agree [/]
Contribution: Introduction, Materials & Equipments, Methodology

Signature: aqif — Read [/]
Name: MUHAMMAD AQIF BIN ROSZAHAN — Understand [/]
Matric Number: 2210345 — Agree [/]
Contribution: Result, Discussion, Experimental Setup

Signature: " — Read[/]
Name: AKMAL ZARIF BIN NAJIB — Understand [/]
Matric Number: 2211971 — Agree [/]
Contribution: Conclusion, Data Collection

Signature: Hammad — Read [/]
Name: SIDDIQUI HAMMAD AHMED — Understand [/]
Matric Number: 2221201 — Agree [/]
Contribution: Data Analysis, Recommendations