

**LABORATORY REPORT**

**MECHATRONICS SYSTEM INTEGRATION MCTA 3203**

**SEMESTER 2 2024/2025**

**WEEK 2**

**SUBMISSION DATE: 17/3/2025**

**GROUP: 6**

|  |  |  |
| --- | --- | --- |
|  | **NAME** | **MATRIC NUMBER** |
| 1. | MUHAMMAD NABIL IMAN BIN ABD RAHMAN | 2313551 |
| 2. | MUHAMMAD NAZHAN BIN MOHAMED NADZRI | 2313703 |
| 3. | MUHAMMAD HAZIQ AJMAL BIN MD KAMAL | 2312821 |

**Table of Contents**

[Abstract 3](#_Toc193058084)

[Introduction 3](#_Toc193058085)

[Materials and Equipment 3](#_Toc193058086)

[Methodology 5](#_Toc193058087)

[Question 6](#_Toc193058088)

[Discussion 6](#_Toc193058089)

[Conclusion 7](#_Toc193058090)

[Recommendations 7](#_Toc193058091)

[Appendix 7](#_Toc193058092)

[Acknowledgement 9](#_Toc193058093)

[Declaration 9](#_Toc193058094)

# Abstract

This project is about interfacing a 7-segment display with an Arduino to create a numeric display that cycles from 0 to 9 and resets to 0, with an increment button to cycle the numbers and one button to reset it back to 0. In this project, we use an Arduino Uno to control the numeric display and controls for the button. This simple circuit demonstrates basic digital logic systems and Arduino programming to help the students get used to more complex systems. The experiment highlighted the potential usage and applications of systems like digital counters, and interactive displays which showcase the core of electrical systems and Arduino programming.

# Introduction

This report describes how a 7-segment display is integrated with an Arduino Board to create a

cyclable numeric display from 0 to 9 when a button is pressed. This project aims to help students understand the key ideas in microcontroller programming, digital electronics and others. From

the manual, the students will connect the components to a breadboard, program the Arduino microcontroller, and construct an algorithm for the counter. This project allows students to dig deep into the world of microcontrollers and electrical systems and bring their creative ideas to life.

# Materials and Equipment

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

**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:

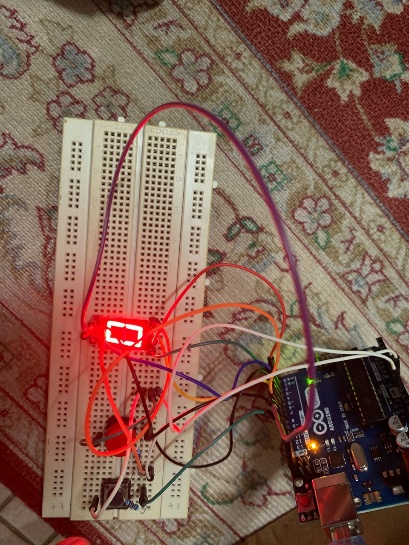
● Connect one leg of each pushbutton to a separate digital pin (e.g., D9 and D10) and connect the other leg of each pushbutton to GND.

● 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

# Methodology

1. Build the circuit according to the circuit setup instructions.
2. Upload the 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.

**Results**

A person holding a circuit board with wires

AI-generated content may be incorrect.A circuit board with wires and a red light

AI-generated content may be incorrect.

A circuit board with wires connected to it

AI-generated content may be incorrect.A circuit board with wires connected to it

AI-generated content may be incorrect.A circuit board with wires and a red light

AI-generated content may be incorrect.

A circuit board with wires and a red light

AI-generated content may be incorrect.A circuit board with wires and a finger pointing at it

AI-generated content may be incorrect.A circuit board with wires and a red light

AI-generated content may be incorrect.

A circuit board with wires

AI-generated content may be incorrect.

# Question

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

To interface an I2C LCD with Arduino, we need to connect the LCD module via the I2C communication protocol which utilizes a specific library to simplify the programming. Comparing the coding principles between the LCDs, I2C LCDs require initialization with the module’s address and display size which utilize library functions to control it. Meanwhile, the 7-segment display uses a more direct segment control and the Matric LED employ managing rows and columns with bit manipulations. Despite all the differences, I2C LCDs offer a more user-friendly approach to text-based information thanks to its higher-level commands.

# Discussion

1. **Hardware**

The setup for this project requires a 7-segment display with an Arduino board connected to it.

The display is the main component for this project which requires a connection to the ground and several digital pins onto the Arduino. Each segment is also connected with a resistor to limit its current preventing excessive current draw that could damage the LEDs. A pushbutton is used to increment the counter display and act as reset button. This simple project allows for it to be assembled on a basic breadboard which helps the students make modifications and troubleshoot problems**.**

1. **Software**

Arduino IDE is used to assemble the code for this project as it provides a robust environment for writing and uploading code to the Arduino. The program is structured with a clear setup where

pins are defined for each segment and the main loop handles the logic of incrementing numbers for the display. The code is designed for easy scalability and modification allowing for adjustments. We found that there are better and easy ways to code the Arduino with 7-segment display which is the library to be downloaded in the software that makes coding progress easier. The library that been downloaded is SevSeg by Dean Reading. This library allows the coding to be simpler and easy to apply with 7-segment display.

# Conclusion

The project aims to demonstrate the simplicity of a simple electronic component, in this case, a 7-segment display and how it can be interfaced with a microcontroller to create a functional counter. In the project, we get to know the importance of electrical characteristics and the basic requirements of electric components. Meanwhile, on the software side, we learn the need for effective programming techniques to create a responsive interface. This labwork lays the

groundwork for more advanced projects and serves as a practical introduction to electronics and programming.  
**Recommendations**

A few recommendations in mind are that we can include a simple buzzer that emits sound with each button press to enhance user interaction which provides the user auditory feedback. We can also add a functionality where the user can decrement the display by adding one more button. This is to make it easier for users to cycle through the numbers.

# Appendix

# Circuit Diagram

# A circuit board with wires and a button

**Coding for Week 2 Experiment**

//download SevSeg library from library manager

#include "SevSeg.h" //use library that easy to use with seven segment display

SevSeg S; //Instantiate a seven segment controller object

byte CommonPins[] = {}; // common pin numbers for multi-digit display(array)

byte SegPins[] = {2,3,4,5,11,7,8}; // 7-segment display pins in the order,{a,b,c,d,e,f,g,dp}(array)

int button1=9; //button for increment

int button2=10; //button for reset

int num=0;

void setup()

{

// begin(COMMON\_CATHODE, NumberOfDigits, CommonPins[], SevenSegPins[], resistorUsed);

S.begin(COMMON\_CATHODE, 1, CommonPins, SegPins, 1);

pinMode(9, INPUT);

pinMode(10, INPUT);

}

void loop()

{

int incre = digitalRead(9);

int reset = digitalRead(10);

//increment

if((incre == HIGH) && (num < 9))

{

delay(100);

num++;

}

//reset

if((reset == HIGH))

{

delay(100);

num=0;

}

//Logic to print digit/character on 7 segment display

S.setNumber(num);

S.refreshDisplay();

delay(100);

}

# Acknowledgement

We would like to express my sincere gratitude to the lab technician for their invaluable guidance, support, and encouragement throughout this project. Their expertise and insights have been instrumental in shaping the direction of this work. We would also like to extend our thanks to our fellow peers for their assistance and collaboration, which greatly contributed to the successful completion of this project.

# Declaration

We hereby declare that the work presented in this report is entirely my own, except where

otherwise acknowledged. We affirm that we have adhered to the principles of academic integrity and have not engaged in any form of plagiarism or unethical conduct in the completion of this project. All sources of information and assistance used in this work have been properly cited and acknowledged.