KULLIYYAH OF ENGINEERING (KOE)

MECHATRONICS SYSTEM INTEGRATION (MCTA3203)

SEMESTER 1, 24/25

SECTION 1

PROJECT REPORT WEEK 2

TITLE : DIGITAL LOGIC SYSTEM

PREPARED BY :

NO	NAME	MATRIC NO
1	ARIF EMRULLAH BIN TAJUL ARIFFIN	2215359
2	AZLIYANA SYAHIRAH BINTI AZAHARI	2210620
3	DAMIA MAISARAH BINTI ZAWAWI	2217830
4	HUDA BINTI AB RAHMAN AL-QARI	2226676
5	IKMAL HAKIM BIN ZAKI	2125625

DATE OF SUBMISSION : 23RD OCTOBER 2024

ABSTRACT

In this experiment, we connected an Arduino Mega 2560 board to a common cathode 7-segment display. The circuit configuration entailed attaching the seven display segments to different digital pins on the Arduino and using the proper current-limiting resistors. The Arduino was also wired with push buttons so that the count on the 7-segment display could be manually increased and reset to zero.

We were able to see the consecutive display of numbers from 0 to 9 on the 7-segment display by pushing the increment button after uploading the provided Arduino code. Push-button control of the display is also shown by hitting the reset button, which resets the count to zero.

Through this experiment, we gained practical insights into interfacing electronic components with Arduino, understanding the principles of controlling a 7-segment display, and utilising push buttons for manual input. This foundational knowledge serves as a basis for more complex projects involving Arduino-based digital logic systems and interfacing applications.

TABLE OF CONTENTS

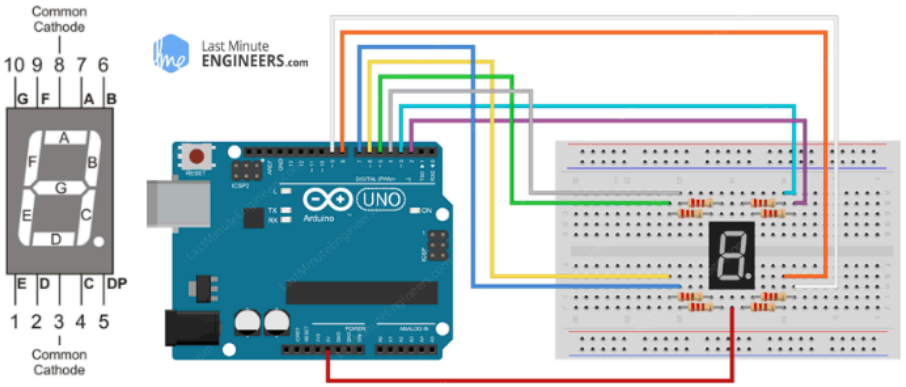
INTRODUCTION

This experiment was conducted to interface a 7-segment display using a pushbuttons that is connected to Arduino Mega 2560, resulting in creation of a digital counter 0 - 9. The purpose is to understand the control mechanisms of basic logic systems through manual input using pushbuttons. By implementing fundamental concepts of digital logic systems, such as logic gates and binary counting, this setup will help in visualising the numeric outputs on the display. The expected outcome is that the display will count from 0 to 9 and reset to 0 sequentially. This serves as a foundational step for more complex digital interfacing systems.

MATERIALS AND EQUIPMENTS

1. Arduino Mega 2560 board
2. Common cathode 7-segment display
3. Resistors
4. Pushbutton
5. Jumper wires
6. Breadboard

EXPERIMENT SETUP



METHODOLOGY

1. The common cathode 7-segment display was connected to Arduino Mega 2560 in following setup:
    - Each of the 7- segments display (a ,b , c, d, e, f, g) was connected to individual digital pins on the Arduino D2 to D8 respectively.
    - The common cathode pin was connected to 5V pin on the Arduino
    - Eight 290 ohm resistors were connected to each segment pin
2. The pushbuttons was connected to Arduino and 7-segment display as follows:
    - One end of the pin was connected to the D10 pin while the other end was connected to the GND pin.

CODING

```
#define BUTTON_PIN 10

int segmentA = 2;

int segmentB = 3;

int segmentC = 4;

int segmentD = 5;

int segmentE = 6;

int segmentF = 7;

int segmentG = 8;


int i = 0;

int j = 0;


int Arduino_Pins[7] = {segmentA, segmentB, segmentC, segmentD,
segmentE, segmentF, segmentG};

int Segment_Display[10][7] = {{0, 0, 0, 0, 0, 0, 1}, //display number 0
                              {1, 0, 0, 1, 1, 1, 1}, //display number 1
                              {0, 0, 1, 0, 0, 1, 0}, //display number 2
                              {0, 0, 0, 0, 1, 1, 0}, //display number 3
```

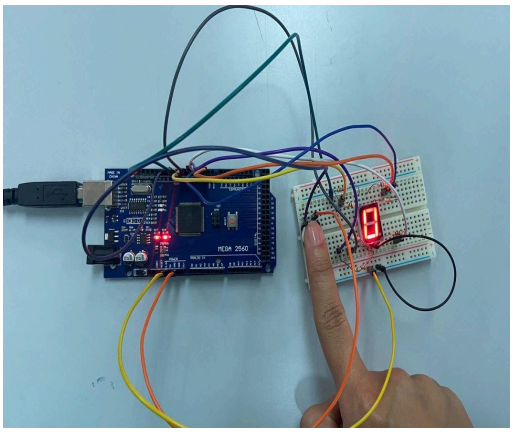
```
                                {1, 0, 0, 1, 1, 0, 0}, //display number 4
                                {0, 1, 0, 0, 1, 0, 0}, //display number 5
                                {0, 1, 0, 0, 0, 0, 0}, //display number 6
                                {0, 0, 0, 1, 1, 1, 1}, //display number 7
                                {0, 0, 0, 0, 0, 0, 0}, //display number 8
                                {0, 0, 0, 0, 1, 0, 0}, //display number 9
};

void setup() {
  // put your setup code here, to run once:
  pinMode(segmentA, OUTPUT);
  pinMode(segmentB, OUTPUT);
  pinMode(segmentC, OUTPUT);
  pinMode(segmentD, OUTPUT);
  pinMode(segmentE, OUTPUT);
  pinMode(segmentF, OUTPUT);
  pinMode(segmentG, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  // put your main code here, to run repeatedly:
  //to display number according to segment pin
  for (j = 0; j<7; j++)
  {
    digitalWrite(Arduino_Pins[j], Segment_Display[i][j]);
  }
  //if button is pushed, number increase
  if(digitalRead(BUTTON_PIN)==0)
  {
    while(digitalRead(BUTTON_PIN)==0)
    {

    }
    i++; //increase counter with each push button
```

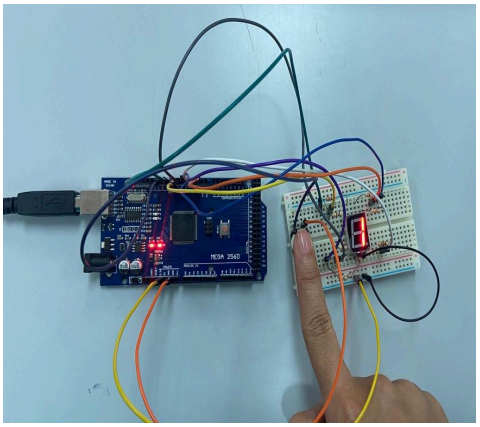
```
  }
  //start count back to 0
  if(i == 10)
  {
    i = 0;
  }
  delay(100);
}
```

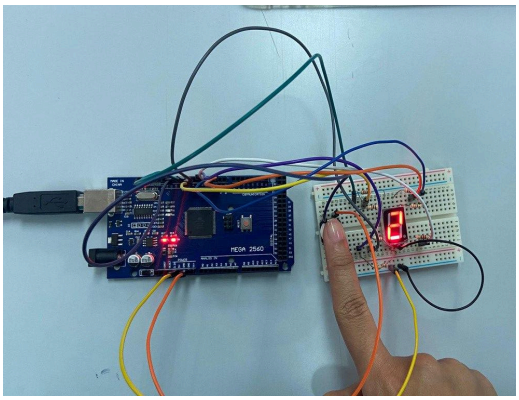
PROCEDURES

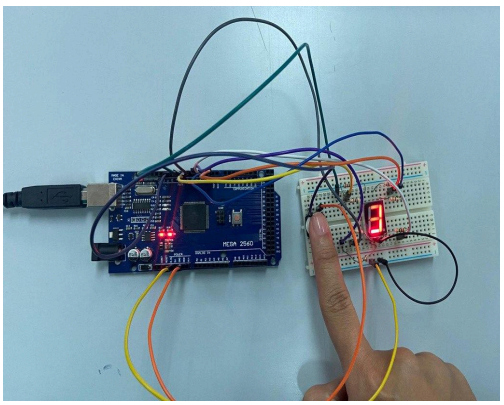
1. Build the circuit according to the circuit setup instructions.
2. Upload the provided Arduino code to Arduino Mega 2560.
3. Open the Serial Monitor in the Arduino IDE.
4. Press the increment button to increase the count. The 7-segment display should show the numbers from 0 to 9 sequentially.
5. Press the reset button to reset the count to 0.

RESULTS









As shown in the photos, the experiment demonstrated how to control a 7-segment display using an Arduino Mega 2560 and pushbuttons. When the push buttons were pressed incrementally, the display sequentially showed numbers from 0 to 9. This proves the correct functionality of each segment and the control logic.

DISCUSSIONS

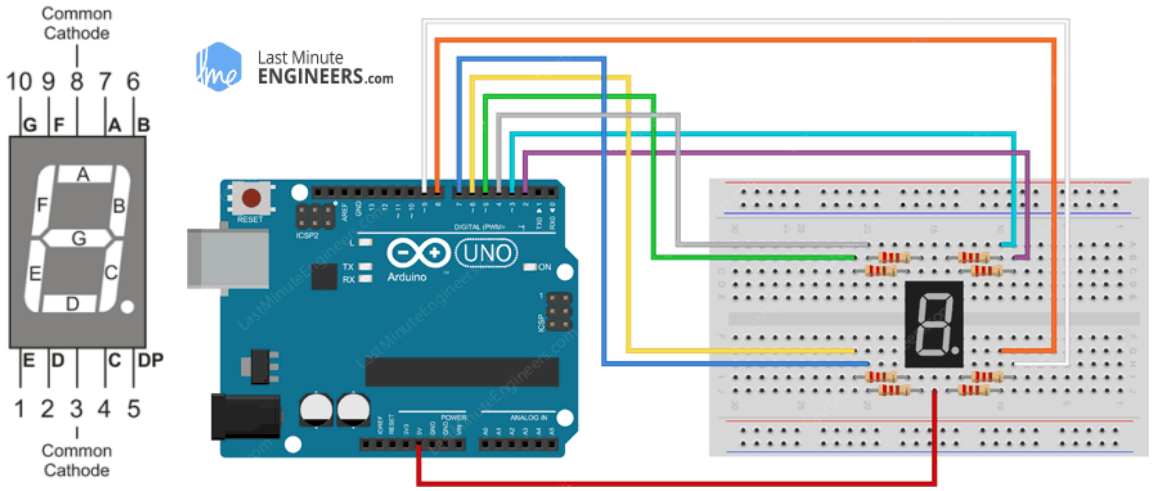
- **Software**

Code is configured to increment the displayed number on a 7-segment display each time a pushbutton is pressed, then roll back to 0 when it reaches 9. The button is attached to BUTTON_PIN (configured to INPUT_PULLUP), so it reads HIGH when not pressed and LOW when pressed.

The number is displayed by looping over the segment pins and setting them based on the current value of ‘i’. Each time the button is pressed, ‘i’ increases. As ‘i’ changes, the number displayed varies because the LED segments are switched on or off based on the Segment_Display array values. If i = 10, the count is reset to zero.

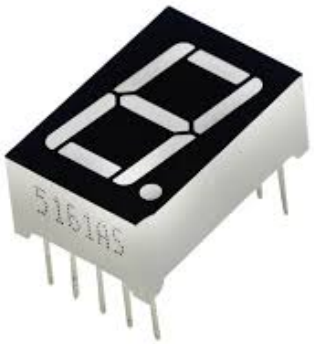
A delay(100) provides a 100 millisecond pause to slow down the loop and keep the code from executing too quickly. It provides smoother operation and prevents quick changes that may be difficult to perceive on the display or in button detection.

· **Electrical**



A 7-segment display was interfaced with Arduino Mega 2560 to create a digital counter that displays numbers 0 to 9, controlled by pushbuttons. The 7-segment display is an electrical device consisting of seven LEDs arranged in a pattern to display numbers. Each LEDs can be arranged to numbers 0 to 9. Arduino Mega 2560 is a microcontroller used to control the 7-segment display. Resistors are used to regulate current flowing through the LED segments to prevent damage. The pushbuttons was used to allow manual input to change the displayed number by incrementing the counter. Lastly, breadboard and wires were used to connect components and facilitate signal transmission.

· **Hardware**

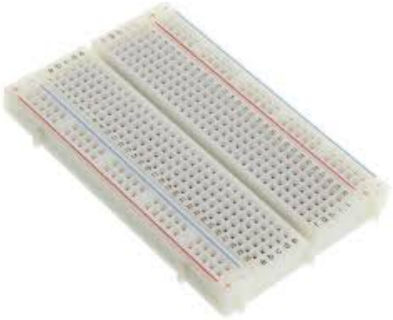


A 7-segment display is an electrical display device that shows decimal numerals which is made up of 7 LEDs arranged in a rectangular pattern to form the numbers 0-9. Each of the LEDs, known as a "segment," can be switched on or off separately to show the correct numeral. For example to show 0, segments a, b, c, d, e, and f are activated. To display 8, all seven segments (a, b, c, d, e, f, and g) are activated.

The microcontroller used for the experiment is the Arduino Mega 2560.

The Pushbutton in the setup is to modify the displayed number. Every time the button is pressed, the state changes (from HIGH to LOW) which can increase a variable that maintains

track of the number to be displayed on the 7-segment display.



Breadboards are primarily used for temporary electronic circuits by connecting components like resistors and LEDs.

Male-to-female jumper wires are used in connecting components, the Arduino Mega and the breadboard. The female end of the jumper wire plugs into the Arduino Mega's pin headers. The male end connects to the breadboard, which allows to send signals or power between the two.

Resistors are for managing the current that flows through the display's LEDs. Resistors prevent excessive electricity from passing through each segment of the 7-segment display, which can potentially destroy the LEDs or the Arduino Mega 2560 pins that power them.

Questions:

To interface the I2C LCD with the Arduino Uno, connect the SDA (data line) and SCL (clock line) of the I2C LCD to the SDA pin (A4) and SCL pin (A5) on the Arduino, respectively. Next, connect VCC and GND to the 5V and GND pins on the Arduino, respectively. Install the LiquidCrystal_I2C library in the Arduino IDE using the Library Manager.

	I2C LCD	7 Segment Display	Matrix LED
Communication	Control via I2C protocol	Direct control of via GPIO pins	Direct control via GPIO pins or dedicated driver
Coding	• Communication is handled by library. • LCD is controlled by sending commands and data to display characters.	• Segment patterns for each digit (0-9, A-F for hexadecimal is defined. • State of each LED segment (on/off) is	• Each LED can be manipulated individually or in groups. • State of each LED (on/off) is managed using more complex

		controlled manually.	**logic.**
Control Level	**High**	**Low**	**Low**

Table 1: The coding principles of the I2C LCD, 7-segment display and matrix LED.

CONCLUSION

In this experiment, we successfully interfaced a 7-segment display with an Arduino Mega to construct a digital counter from 0 to 9 using pushbuttons . This setup provided a demonstration to digital logic systems, electronic circuit interfacing and basic control structures. The results align with the expected outcome, reinforcing the understanding of how binary counting and segment control work in simple display systems.

RECOMMENDATIONS

1. Emphasise the real-world applications of digital logic systems throughout the experiment. Discuss how digital logic concepts are applied in a variety of technologies and businesses, including telecommunications, and digital signal processing

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Wahju Sediono, Dr. Ali Sophian, Dr. Zulkifli Bin Zainal Abidin, for providing the necessary resources and facilities to conduct this and support throughout the duration of the project. Additionally, we extend our appreciation to all individuals who contributed to the success of this lab report through their valuable insights and feedback.

STUDENT'S DECLARATION

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 that no further improvement on the reports is needed from any of the individual contributors to the report.

Signature: 	**Read**	/
Name: Arif Emrullah Bin Tajul Arifin	**Understand**	/
Matric No: 2215359	**Agree**	/

Signature: 	**Read**	/
Name: Azliyana Syahirah Binti Azahari	**Understand**	/
Matric No: 2210620	**Agree**	/

Signature: 	**Read**	/
Name: Damia Maisarah Binti Zawawi	**Understand**	/
Matric No: 2217830	**Agree**	/

Signature:	**Read**	/

Name: Ikmal Hakim Bin Zaki	**Understand**	/
Matric No: 2125625	**Agree**	/

Signature: 	**Read**	/
Name: Huda binti Ab Rahman Al-Qari	**Understand**	/
Matric No: 2226676	**Agree**	/