MCTA 3203 (MECHATRONICS SYSTEM INTEGRATION)
WEEK 2: DIGITAL LOGIC SYSTEM

TITLE:

Covering basic logic gates, electronic circuit interfacing, basic Arithmetic Logic Unit (ALU), 7-segment display, and applications involving IC-based interfacing

SEMESTER 1, 24/25

SECTION 1 – GROUP 9

LECTURER:
DR. WAHYU

no	Name	Matric number
1	AZMI BIN BASHARUDIN	2211387
2	HAMZAH FAISAL	2219537
3	KHAIRULDDIN BIN ZULKIFLEI	2210527
4	MUHAMAD IZZUDIN BIN MUHAMAD	2219735
5	MUHAMMAD AIMAN BIN MOHD RODZI	2118813

DATE OF EXPERIMENT:

DATE OF SUBMISSION:

ABSTRACT

This lab experiment uses an Arduino microcontroller to display an ascending number sequence on a 7-segment display. The 7-segment display displayed values ranging from 0 to 9. The experiment aimed to teach basic digital logic, binary-coded decimal, and multiplexing ideas using microcontroller-controlled displays. This interface provided exact control over each display section. This enabled direct display of numeric data and alphanumeric characters. This experiment advances our understanding of microcontroller integration with display technology and has practical implications for digital electronics.

TABLE OF CONTENT

1. INTRODUCTION

The purpose of this experiment is to gain practical knowledge in handling 7-segment displays to visually represent the numerical digits (0-9) and encourage problem-solving skills by allowing students to troubleshoot and debug any issues that arise during experimentation, improving their ability to identify and solve electronic problems. A 7-segment display uses seven LED segments arranged in a precise configuration to represent numbers 0-9 and alphabetic letters. There are two main types of seven-segment displays: Common Cathode (CC) and Common Anode. The categorization depends on which LED pin is connected to the common pin. Common anode displays connect LEDs at their anodes. Common Cathode displays occur when multiple Cathodes are coupled together.

The Arduino is a popular open-source microcontroller platform that enables experts to develop various electronic projects. Arduino supports 7-segment displays with either common cathode or common anode configurations. The common cathode connects all negative terminals, whereas the common anode connects all positive terminals.By connecting a 7-segment display to an Arduino, we can program it to display numeric digits (0-9) and other characters. In experiments, two push buttons are typically used as input devices to elicit specific activities. For example, one push button can cycle through a shown number while another can reset the display or perform a certain function.The idea suggests that push buttons will enable user engagement with the 7-segment display. Pressing the push button is supposed to change the displayed number or trigger an action visible on the display.

Objectives

1. To demonstrate ascending number sequence using Arduino and 7-segment display.
2. To design and implement the working of a 7-segment display using Arduino.
3. To understand the interface between Arduino and 7-segment display.

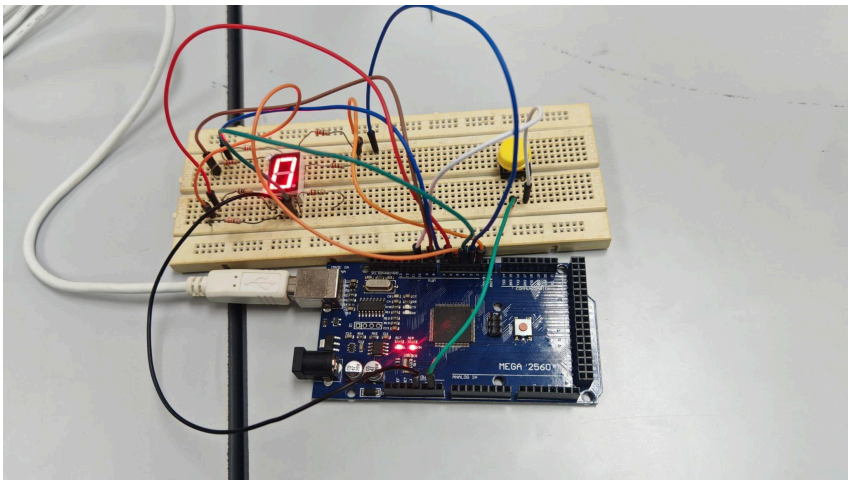
2. MATERIALS AND EQUIPMENT

i. Arduino Mega 2560 board
ii. Common Cathode 7-segment display
iii. 8 220-ohm resistors
iv. Pushbutton
v. Jumper wires
vi. Breadboard

3. EXPERIMENT SETUP

i. Put the seven segment display onto the breadboard.
ii. Connect each of the connectors on the 7-segment display to the digital pins on the Arduino Mega.
iii. Attach a 220-ohm resistor on each connection between the 7-segment display and the Arduino pins to limit the current to the segments.
iv. Connect the common cathode pin of the 7-segment display to any GND pin on the Arduino
v. Connect the push button legs to pin 9 and 10 of the Arduino.
vi. Connect the Arduino to a laptop with Arduino IDE already installed.

Our work focused on the 7-segment display as a common cathode type. The common cathode pin, with its shorter leg, was attached to an Arduino 5V pin. Each segment of the 7-segment display (a, b, c, d, e, f, g) was linked to its own digital output pin on the Arduino. Each 7-segment pin will be linked to a 220 ohm resistor to limit current flow. Our circuit schematic guided the selection of pins, which were then controlled by Arduino code to illuminate certain regions for displaying numbers or characters.Two push buttons were added as input devices for user interaction. Each push button terminal was linked to a different digital input pin on the Arduino board. To assure correct operation, we used pull-up resistors with a 10k ohm value between the digital input pin and VCC. In this experiment, we use two push buttons: one to manually increase the number sequence, and another to return it to zero. Jumper wires were utilised to connect components in accordance with the specified circuit schematic. A breadboard was used to organise and simplify wiring, improving the overall configuration. To apply the experiment's functionality, we uploaded the Arduino code to the board after checking all connections were properly made. Please find attached our schematic diagram and circuit view created with Tinkercad.



4. METHODOLOGY

i. Open Arduino IDE on your laptop.
ii. Insert the code provided.
iii. Upload the code to the Arduino Mega.
iv. Press the pushbutton and record the number changes on the 7-segment display.

This project teaches how to control a 7-segment display with an Arduino Uno. To ensure a productive workspace, gather all necessary supplies and components. link the 7-segment display to the breadboard first, then link each segment pin (a,b,c,d,e,f,g, and dp) to the corresponding digital pins on the Arduino using jumpers. To use a common cathode display, connect the common cathode pin to 5V and segments to digital pins. To control the 7-segment display, the algorithms were written using an Arduino sketch. The Arduino sketch's tools menu allows you to select the right board and COM port for code uploading.After compiling the codes and ensuring no errors, upload them to the Arduino. Tinkercad was used to simulate the circuit code before analysing the results.To achieve the required effects, modify the 7-segment display, which can display numbers from 0 to 9 in sequential, reset, and incremental sequences upon push button touch. The observations are recorded. To prevent short circuits and component damage, connections are made securely using resistors. To make adjustments to the circuit, disconnect the power supply first.

```
#define BUTTON_PIN 10

// Define segment pins
const int segmentPins[7] = {2, 3, 4, 5, 6, 7, 8};

// Segment patterns for numbers 0-9
const int segmentPatterns[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
};

int currentNumber = 0; // start dengan 0

void setup() {
        // Set up segment pins as output
```

```
        for (int i = 0; i < 7; i++) {
        pinMode(segmentPins[i], OUTPUT);
        }

        // Set up button pin
        pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
        displayNumber(currentNumber);

        if (digitalRead(BUTTON_PIN) == LOW) {
        while (digitalRead(BUTTON_PIN) == LOW) {

        }
        incrementNumber();
        }

        delay(100);
}


void displayNumber(int number) {

        if (number < 0) number = 0;
        if (number > 9) number = 9;

        for (int j = 0; j < 7; j++) {
        digitalWrite(segmentPins[j], segmentPatterns[number][j]);
        }
}


void incrementNumber() {
        currentNumber++;
        if (currentNumber == 10) {
        currentNumber = 0;
        }
}
```

6. DATA ANALYSIS

The 7-segment display is controlled by two push buttons: one to advance the count and another to reset it to zero. In the circuit design and tinkercad code, 'pushbutton_reset' (connected to pin 0) resets the count to zero and 'pushbutton_inc' (connected to pin 1) increments it.\When the 'pushbutton_reset' is pressed, the count is reset to zero and shown on the 7-segment display. When 'pushbutton_inc' is pressed, the count is incremented by 1-10 times and shown using the 'displayNumber' function.

7. RESULT

Overall, the experiment succeeded in meeting its theoretical aims.
Using Tinkercad to simulate the circuit helped achieve the desired result. Using an Arduino microcontroller and a 7-segment display, we demonstrated an ascending number sequence and a reset function that resets the display to zero. We gained a thorough understanding of the interface between Arduino and the 7-segment display. Nevertheless, certain objectives may not be fully fulfilled in the practical circuit. This experience emphasises the need for more refining and optimization in future studies, such as adjusting the circuit architecture or exploring new variables. The exercise provided a valuable learning opportunity by bridging the gap between theory and practical application.

8. DISCUSSION

The code identifies the pins for each segment (D0 to D6) of the 7-segment display. It also specifies pins for two push buttons used to control the display. The code includes numerous routines for manipulating display segments and monitoring push button inputs. The functions are as follows:

- setup(): In this function, digital pins are initialised as OUTPUTs, allowing for control over the segments and reading input from the push buttons. 11

- loop(): This function operates to verify the state of the push buttons (state1 and state2) to check whether they are pressed (LOW) or not. Upon detecting a button press, the code updates the count, showing a sequential display of numbers from 0 to 9. The code employs a delay(1000) to ensure each number is displayed for a second before transitioning to the next one. The second button, when pressed, sets the count to 0.

- displayNumber(int num): This function is used in controlling the individual segments of the 7-segment display based on the number to be displayed. Each case within the switch statement corresponds to a specific number (0-9) and dictates the state of each segment, allowing for the accurate representation of the number on the display.

9. CONCLUSION

This experiment successfully demonstrated the 7-segment display's ability to visually represent numerical digits in sequence. It was conceived, manufactured, and tested thoroughly. To control and show numerical digits on a 7-segment display, we connected an Arduino microcontroller and added push buttons as input devices. Tinkercad enabled us to visualise and validate desired outcomes before building a physical circuit connection using Arduino. The push button is essential for interfacing with the 7-segment display, as we

anticipated. Thus, our hypothesis was confirmed.During the experiment, several issues impacted the system's performance and functionality. We would gain practical experience troubleshooting and debugging electronic circuits, which would enhance our skills. 12 insights on theoretical theories. Our experience enables us to integrate Arduino and 7-segment displays into future inventions. Improving our understanding and skills in this area is crucial for successfully integrating components into future experiments and projects.

10. RECOMMENDATION

To improve future experiments, focus on error management in the code. The exercise should teach students how to handle unexpected user inputs. Students should be taught how to handle non-numeric characters on a 7-segment display intended for numerical values. To do this, the following suggestions should be considered.

1.Future experiments should prioritise error management at their core. This involves teaching students about error identification and repair in electronic projects.

2.Encourage pupils to identify potential mistakes in their work. It's important to consider the risk of encountering erroneous or inappropriate input values.

3.To incorporate validation checks in their code, students should follow the experiment instructions. These checks ensure that the system handles erroneous data or input appropriately.

Integrating mistake management into activities enhances students' practical skills and understanding of error prevention. These skills are valuable and can be applied immediately in future engineering and electrical projects.
This experiment taught pupils to approach problem-solving challenges. Electronics projects generally require debugging and problem-solving skills, which are essential in engineering. This experiment helps students develop good breadboarding and wiring skills.

11. REFERENCES

1. Zulkifli. (2014). Mechatronics Interfacing Lab Manual, (Rev. ed.). Unpublished Class Materials.

2. Interfacing Seven Segment Display with Arduino (Joseph, 2022)
https://circuitdigest.com/microcontroller-projects/interfacing-seven-segment-displaywith-Arduino

14. STUDENTS 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 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		
NAME	Azmi bin basharudin	**Read** ✔
MATRIC NUMBER		**Understand** ✔
CONTRIBUTION		**Agreed** ✔

SIGNATURE		
NAME	Hamzah faizal	**Read** ✔
MATRIC NUMBER		**Understand** ✔
CONTRIBUTION		**Agreed** ✔

SIGNATURE		
NAME	Khairulddin bin zulkiflei	**Read** ✔
MATRIC NUMBER		**Understand** ✔
CONTRIBUTION		**Agreed** ✔

SIGNATURE		
NAME	Muhamad izzudin bin muhamad	**Read** ✔

MATRIC NUMBER		**Understand** ✔
CONTRIBUTION		**Agreed** ✔

SIGNATURE		
NAME	Muhammad aiman bin mohd rodzi	**Read** ✔
MATRIC NUMBER	2118813	**Understand** ✔
CONTRIBUTION		**Agreed** ✔