MCTA 3203 SYSTEM INTEGRATION LAB

WEEK 2:
DIGITAL LOGIC SYSTEM

SECTION : 1
GROUP NUMBER : 7 (G)

LECTURER'S NAME : DR ZULKIFLI BIN ZAINAL ABIDIN

GROUP MEMBERS :

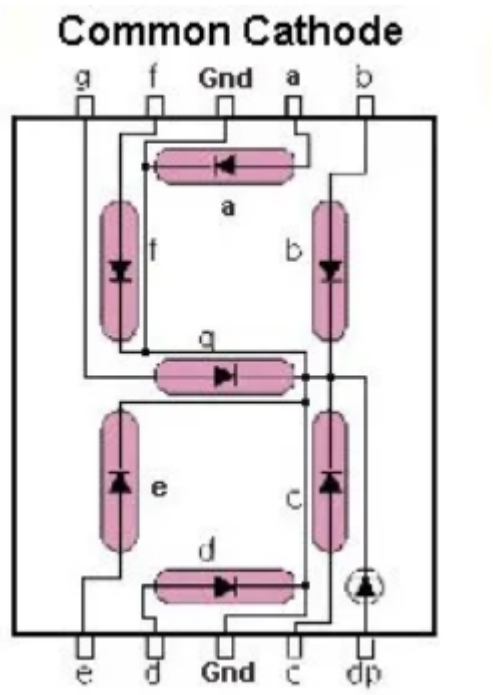
NO :		COURSE	MATRIC
1.	MUHAMMAD NAQIB AIMAN BIN YUSNI	BMCT	2117765
2.	MUHAMAD AZRI BIN ABDUL AZIZ	BMCT	2113537
3.	MUHAMMAD HAFIDZUDDIN HANIF DANIAL BIN NORIZAL	BMCT	2123651
4.	SHAREEN ARAWIE BIN HISHAM	BMCT	2116943

TABLE OF CONTENTS

ABSTRACT

7 Segment Display:

A digital 7-segment display is an electronic component used to visually represent numerical characters. It consists of seven independently programmable segments placed in a particular pattern. By turning on or off individual segments, it allows numbers 0 through 9 to be displayed.Thus, it is frequently found in a variety of electronic equipment, such as microwaves, calculators, and digital clocks. A digital 7-segment display is an easy to understand and straightforward way of transmitting information. The one that we used in this is a common cathode seven segment display.It common path to ground is shared by the segments. Individual segments are lit when their pin receives sufficient positive voltage. Other than that it maximum voltage for it is 5V which it can directly connect to arduino UNO.



Arduino UNO:

The Arduino Uno has a total of 14 digital input/output pins, of which 6 can be used as PWM (Pulse Width Modulation) outputs. Additionally, there are 6 analog input pins. This brings the total number of available input/output pins to 20. These pins can be used for various purposes, including reading digital and analog sensors, controlling LEDs, motors, and other actuators, and communicating with other devices

Push Button:

A momentary push button switch is a type of electrical switch that is designed to make momentary contact when pressed. It is commonly used in electronic circuits to provide temporary electrical connections. When the button is pressed, it completes the circuit, allowing current to flow through.When the button releases, the circuit opens again. This type of switch is often used for functions like power on/off toggles, resetting devices, or triggering specific actions in electronic systems. In this experiment we use Push Button to make the number from the 7 segment display increase and decrease.

Introduction

The experiment is conducted to help us understand how the 7 segment display works by connecting it to the Arduino UNO and controlling it by push button. The arduino UNO will light up the segment from the 7 segment display according to the code that we have set up in it. Other than that, by using two push buttons we can control the number from the display to increase and decrease respectively.

Objective

1) To understand how the 7 segment display work
2) To understand how to connect 7 segment with arduino UNO

Material & Equipments

1. · Arduino Uno board x1
2. · Common cathode 7-segment display x1
3. · 220-ohm resistors x2
4. · Pushbuttons x2
5. · Jumper wires
6. · Breadboard x1

Circuit 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 UNO (e.g., D2 to D8).
2. Connect the pushbuttons to the Arduino UNO as follows:

- Connect both push buttons to separate digital pins on the Arduino UNO D9 and D10 respectively.
-

Experiment Setup

1. Build the circuit according to the circuit setup instructions.
2. Upload the provided 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. The number returns to 0 when the button is being pressed after the number 9.

Circuit Diagram

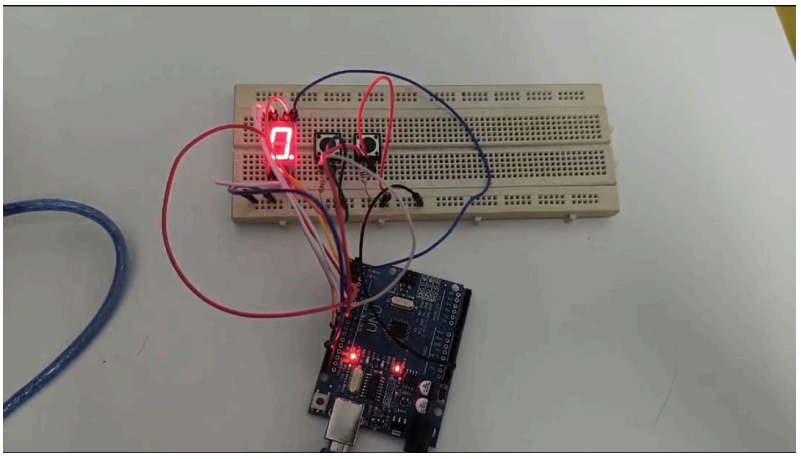


Figure 1: Wire connection on Arduino Uno and 7-segment and push button.

Results

The experiment delivered an operating counter display with a range of 0 to 9 and a button that allows the display to be reset to 0 and increased in steps of 1.

Discussion

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 an arduino, we have to connect the hardware first, after that we have to write the code in Arduino IDE and upload the code to the Arduino board then we can observe the output on the LCD.

The coding principle behind interfacing an I2C LCD with Arduino involves sending commands and data over the I2C bus to control the LCD display.

Compared to a 7-segment display, an I2C LCD provides more flexibility in displaying characters, symbols, and even graphics. The coding for a 7-segment display typically involves directly controlling each segment or using a library for convenience.

On the other hand, a matrix LED display involves multiplexing rows and columns of LEDs to display characters or images. The coding for a matrix LED display often requires more complex logic to manage the multiplexing and control individual LEDs, compared to the relatively straightforward commands used with an I2C LCD.

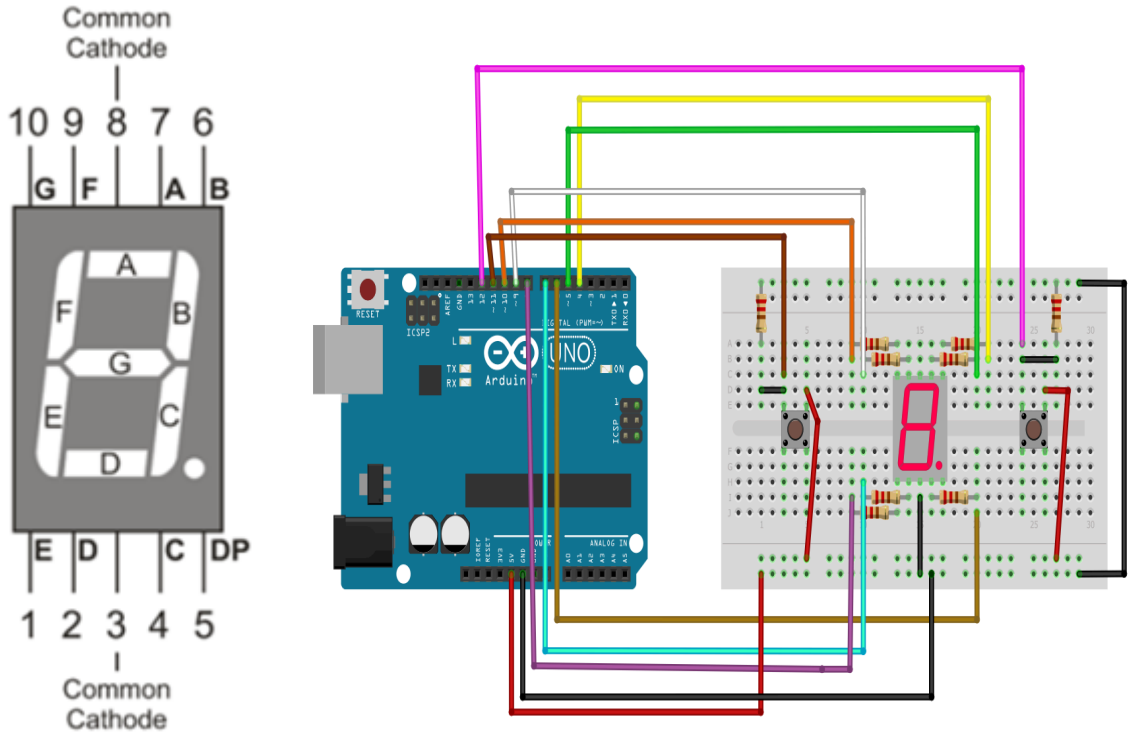
Conclusion

In conclusion, the experiment involving the integration of a seven-segment display with two push buttons for count up and count down functions has provided valuable insights into the practical application of digital displays and user interface design. The implementation of these buttons allowed for user interaction, making it possible to increment the displayed count and reset it to zero, demonstrating the versatility and utility of the seven-segment display in various real-world scenarios.

Through this experiment, we have experienced firsthand how a simple arrangement of 7-segments can be harnessed to communicate numerical information effectively. The

count-up button showcases the display's capacity to convey increasing values, while the reset button highlights its responsiveness to user input for control and interaction.

Appendices



Appendix A: Connection of Arduino Uno to 7- segment display

```
const int segmentA = 2;
const int segmentB = 3;
const int segmentC = 4;
const int segmentD = 5;
const int segmentE = 6;
const int segmentF = 7;
const int segmentG = 8;
const int switchbuttondown = 9;
const int switchbuttonup = 10;
const int DP = 11;

int count = 0;
int buttonUpState = 0;
int iniButtonUpState = 0;
int buttonDownState = 0;
int iniButtonDownState = 0;
```

```
void setup() {
  pinMode(segmentA, OUTPUT);
  pinMode(segmentB, OUTPUT);
  pinMode(segmentC, OUTPUT);
  pinMode(segmentD, OUTPUT);
  pinMode(segmentE, OUTPUT);
  pinMode(segmentF, OUTPUT);
  pinMode(segmentG, OUTPUT);
  pinMode(DP, OUTPUT);
  digitalWrite(DP ,HIGH);
}

void loop() {
  buttonUpState = digitalRead(switchbuttonup);
  buttonDownState = digitalRead(switchbuttondown);
  if (buttonUpState != iniButtonUpState) {
    if (buttonUpState == HIGH) {
      count++;
      Serial.println(count);
      changeNumber(count);
      if(count == 9) {
        count = 9;
      }
      delay(200);
    }
    delay(10);
  }
  if (buttonDownState != iniButtonDownState) {
    if (buttonDownState == HIGH) {
      count--;
      Serial.println(count);
      changeNumber(count);
      if(count == 0) {
        count = 0;
      }
      delay(200);
```

```
    } else {
      Serial.println("OFF");
    }
    delay(10);
  }
  changeNumber(count);
}

void changeNumber(int buttonPress) {
  switch (buttonPress) {
    case 0:
    digitalWrite(segmentA, HIGH);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, HIGH);
    digitalWrite(segmentE, HIGH);
    digitalWrite(segmentF, HIGH);
    digitalWrite(segmentG, LOW);
    break;
    case 1:
    digitalWrite(segmentA, LOW);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, LOW);
    digitalWrite(segmentE, LOW);
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, LOW);
    break;
    case 2:
    digitalWrite(segmentA, HIGH);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, LOW);
    digitalWrite(segmentD, HIGH);
    digitalWrite(segmentE, HIGH);
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, HIGH);
    break;
```

```
case 3:
digitalWrite(segmentA, HIGH);
digitalWrite(segmentB, HIGH);
digitalWrite(segmentC, HIGH);
digitalWrite(segmentD, HIGH);
digitalWrite(segmentE, LOW);
digitalWrite(segmentF, LOW);
digitalWrite(segmentG, HIGH);
break;
case 4:
digitalWrite(segmentA, LOW);
digitalWrite(segmentB, HIGH);
digitalWrite(segmentC, HIGH);
digitalWrite(segmentD, LOW);
digitalWrite(segmentE, LOW);
digitalWrite(segmentF, HIGH);
digitalWrite(segmentG, HIGH);
break;
case 5:
digitalWrite(segmentA, HIGH);
digitalWrite(segmentB, LOW);
digitalWrite(segmentC, HIGH);
digitalWrite(segmentD, HIGH);
digitalWrite(segmentE, LOW);
digitalWrite(segmentF, HIGH);
digitalWrite(segmentG, HIGH);
break;
case 6:
digitalWrite(segmentA, HIGH);
digitalWrite(segmentB, LOW);
digitalWrite(segmentC, HIGH);
digitalWrite(segmentD, HIGH);
digitalWrite(segmentE, HIGH);
digitalWrite(segmentF, HIGH);
digitalWrite(segmentG, HIGH);
break;
case 7:
```

```
    digitalWrite(segmentA, HIGH);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, LOW);
    digitalWrite(segmentE, LOW);
    digitalWrite(segmentF, LOW);
    digitalWrite(segmentG, LOW);
    break;
    case 8:
    digitalWrite(segmentA, HIGH);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, HIGH);
    digitalWrite(segmentE, HIGH);
    digitalWrite(segmentF, HIGH);
    digitalWrite(segmentG, HIGH);
    break;
    case 9:
    digitalWrite(segmentA, HIGH);
    digitalWrite(segmentB, HIGH);
    digitalWrite(segmentC, HIGH);
    digitalWrite(segmentD, HIGH);
    digitalWrite(segmentE, LOW);
    digitalWrite(segmentF, HIGH);
    digitalWrite(segmentG, HIGH);
    break;
  }
}
```

Appendix B: Arduino Coding (Refer to Github for .ino file)

https://github.com/HafidzuddinNorizal/MCTA3203_GROUP7

Appendix C: Video demonstration of System (Github)

Student's Declaration

Declaration:
We certify that this project/assignment is entirely our own work, except where we have given fully documented references to the work of others, and that the material in this assignment has not previously been submitted for assessment in any formal course of study.

azri

_____
Name: Muhamad Azri bin Abdul Aziz

hafidzuddin

_____
Name: Muhammad Hafidzuddin Hanif Danial bin Norizal

arawie

_____
Name: Shareen Arawie bin Hisham

naqib

_____
Name: Muhammad Naqib Aiman bin Yusni

Date: 18/3/2024