SEM 1 25/26

SECTION 1

COURSE CODE: MCTA 3202

LECTURER NAME:
ZULKIFLI BIN ZAINAL ABIDIN

WEEK 2: DIGITAL LOGIC SYSTEM BASIC LOGIC GATES, ELECTRONIC CIRCUIT INTERFACING, BASIC ALU, 7 SEGMENT DISPLAYS, ICs BASED INTERFACING APPLICATION

PREPARED BY:
GROUP 1

NAME	MATRIC NUMBER
AMIR MUIZZUDDIN BIN NORAIDIL RAZMAN	2129579
MUHAMMAD HAIKAL HANIF BIN ABDUL RAZAK	2213297
AHMAD ILHAM BIN ABDUL AZIZ	2112109

INTRODUCTION

The task for the second week is to learn how to make a digital counter from 1 to 5 and then a task where two buttons are required to increase the number from 1 to 9 and reset the counter to 0. The purpose of this experiment is to help us understand how to interface a 7-segment display with an Arduino UNO and how to control it manually using buttons. We used the void loop argument for the button so it can increase by 1 and reset it back to 0. By the end of this task, the expected outcomes are we have successfully fulfilled two of the tasks which are a counter and a 2 push button. It also helps us to elevate our abilities in programming

MATERIALS

- Arduino UNO Board
- Common cathode 7-segment display
- 7 220-ohm resistors
- 2 pushbuttons
- Jumper wires
- Breadboard

EXPERIMENTAL SETUP

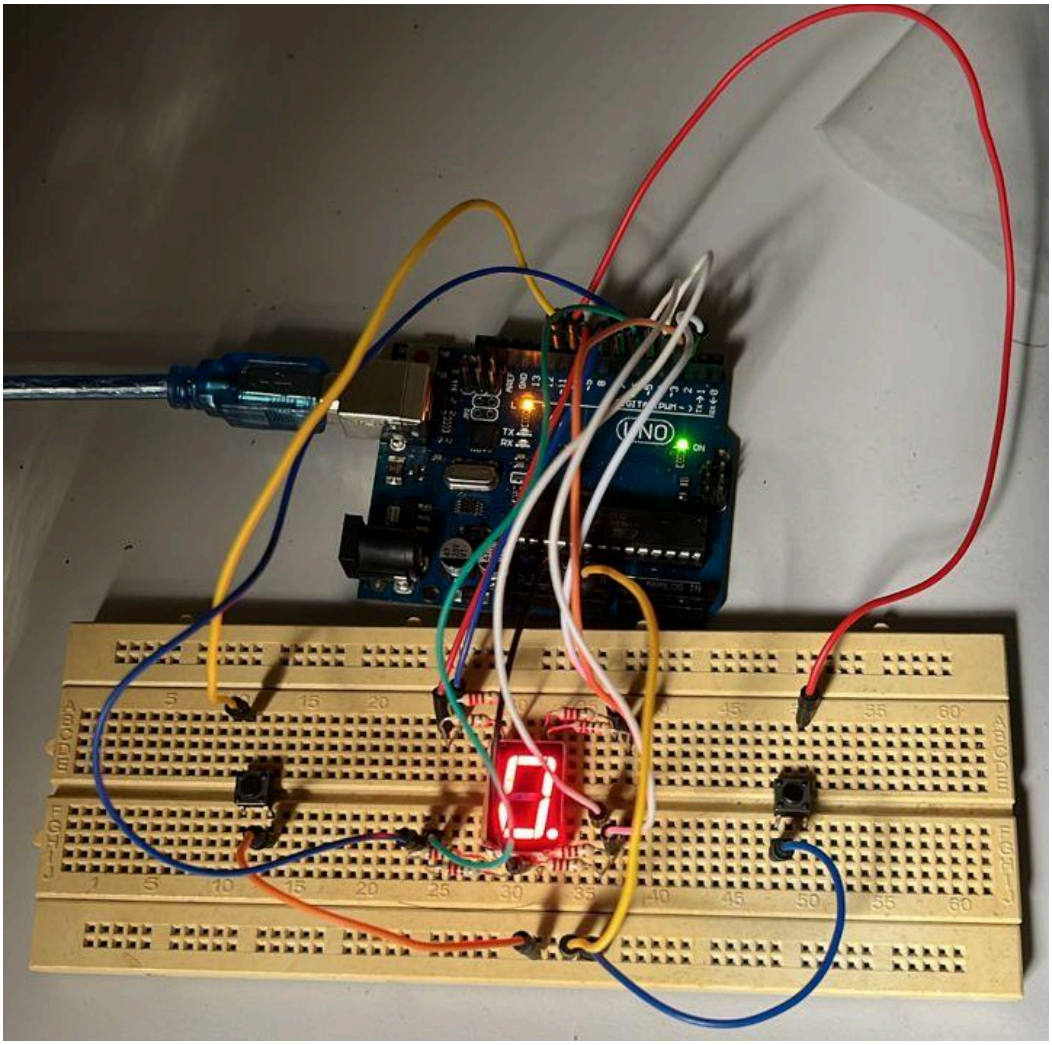
1. Build the circuit according to the circuit setup instructions.
2. Upload the provided Arduino code to your Arduino Uno.
3. Open the Serial Monitor in the Arduino IDE.
4. Press the increment button to increase the count. The 7-segment display should show the numbers from 0 to 9 sequentially.
5. Press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

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 (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.

RESULTS



Discussion

The experiment successfully demonstrated how a 7-segment display can be controlled using an Arduino. Each segment corresponds to an individual digital output, and by controlling the logic HIGH/LOW states, different numbers can be formed. The inclusion of push buttons allowed user interaction for incrementing and resetting the displayed values, showing how input and output devices can be integrated within one microcontroller system.

During the experiment, two Arduino pins (D7 and D8) were found to be faulty, requiring reassignment to D11 and D12. This modification highlighted the importance of flexible coding and pin management in hardware troubleshooting.

Overall, the experiment provided hands-on experience in:

- Digital logic control and pin configuration.
- Interfacing hardware with software logic.
- Debugging and improving circuit functionality.

This exercise built a foundational understanding of how digital logic, electronics, and programming combine in mechatronic system design.

Questions : How can you interface an I2C LCD with arduino? Explain the coding principle behind compared to a 7-segment display and a matrix LED.

Answer :

Interfacing I2C LCD with Arduino :

An I2C LCD (for example, 16×2) uses only two wires for communication:

- SDA (A4) → sends data
- SCL (A5) → sends clock signal

Connection:

- VCC → 5V
- GND → GND
- SDA → A4
- SCL → A5

You need the LiquidCrystal_I2C library to control the display easily.

Example Code:

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>


LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
```

```
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Hello, World!");
}
void loop() {}
```

Summary :

- I2C LCD: Easy to use, needs only 2 pins, shows text.
- 7-Segment: Manual control of each LED segment, shows only digits.
- Matrix LED: More complex, used for scrolling text or images.

So, the I2C LCD is more efficient and simpler to code compared to the 7-segment and matrix LED displays.