`

| TAYLOR'S | SCHOOL OF ENGINEERING |
|---|---|

# ELECTRICAL AND ELECTRONIC ENGINEERING PROGRAMME

## PROGRAMMING TECHNIQUES
## (EEE 60104)

| | | | |
|---|---|---|---|
| **Group No.** | **1** | **Total Marks** | |
| **Lab No.** | **2** | | |
| **Title of Lab** | **7 Segment Display Control using DIP Switch** | | |

| | | |
|---|---|---|
| **Lecturer** | Dr. Chew Wei Jen | |
| **Date** | **Lab Date: October 4th, 2023** | **Submitted Date: October 17th, 2023** |

| **Lab Report Item/total marks** | **Marks** |
|---|---|
| Format/10 | |
| Abstract and Introduction/15 | |
| Materials and Method/15 | |
| General Results and Discussions/50 | |
| Conclusion and References/Appendix /10 | |
| Total /100 | |



| **No.** | **Members ID** | **Name of Group Members** |
|---|---|---|
| 1 | 0360231 | Muhammad Rafif Edie Wijaya |
| 2 | 0364144 | Law Shing Yi |
| 3 | 0361508 | Wong Kai Tian |
| 4 | 0349907 | Syazani Zikry Zahari |
| 5 | 0353217 | Surryaraj A/L Poobalan |

`

| TAYLOR'S | SCHOOL OF ENGINEERING |
|---|---|

**Table of Contents:**

## 1. Abstract

In this lab experiment, our aim was to create a C program that would use an Arduino Uno and a DIP switch to operate a 7-segment display and show the numbers 0 to 9 on it.The 7-segment display is a commonly used output device in electronics and digital design, and it plays a crucial role in displaying numeric data in a visually accessible format.

We used a variety of electronic parts, including a DIP switch, 10k and 270 ohm resistors, to complete this.This lab report describes how to program the Arduino Uno to communicate with the 7-segment display so that we can change the numbers shown on the 7-segjment using the DIP switch.

## 2. Introduction

In this lab, the Arduino Uno will be used to interpret a DIP switch that serves as a binary input to display specific digits on the 7-segment display. This lab aims to build a circuit that will show sequences ranging from 0 to 9 on a 7-segment display, with the pattern determined by the input value of a DIP switch, and to develop a C program for an Arduino Uno to accomplish this functionality.

A DIP switch is a compact mechanical switch device that is often used for configuration and control in electrical circuits. It is made up of several tiny individual switches that are aligned in a dual inline arrangement, with commonly 2, 4, 8, or more switch locations. DIP switches are suitable for binary input because you can generate binary patterns to represent different values or states by changing the locations of the individual switches. In this lab, we will utilize a DIP switch to control our 7-segment display using binary values.

A 7-segment display is a type of visual output device that often represents numbers, letters, and special characters. It is made up of seven separate LED segments that may be lighted in various combinations to create diverse patterns. Each segment represents a portion of a digit, and we may show numerals from 0 to 9 as well as other characters by manipulating which parts are lighted. The 7-segment display will act as our output device

in this lab, and we will programme the Arduino Uno to control which segments light up to produce the required number pattern.

## 3. Methodology

a. Materials:

i. Arduino Uno
ii. 7 segment display
iii. 270Ω resistors
iv. 10kΩ resistors
v. DIP switch

b. Methods:

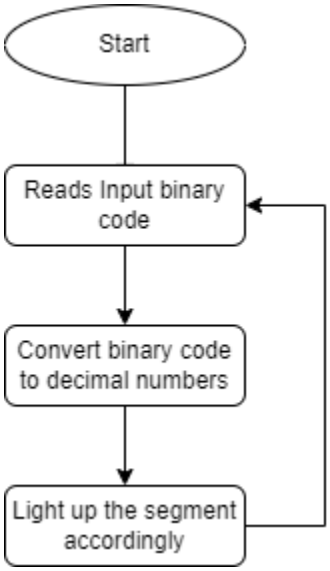


Figure 1. Flowchart of the program.

c. Procedure:

i. The a-g pins of the 7-segment display are each connected to a 270Ω resistor and to the digital pin port 2-8 of the Arduino respectively.
ii. The DIP switch is connected with the switch on config using a 10kΩ resistor into the pins 9-12 of the Arduino.

iii. A C program is written with Arduino IDE to display numbers from 0 to 9 based on the input binary value of the DIP switch.

## 4. Results

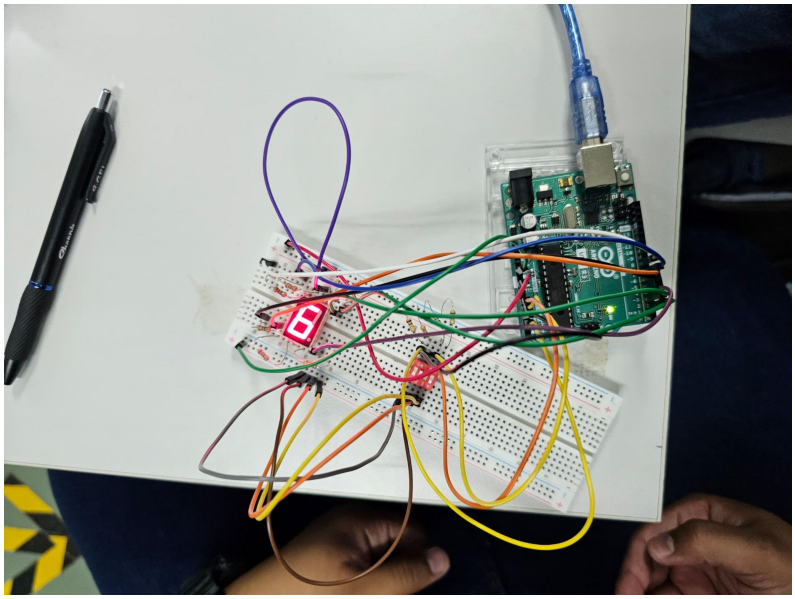

Figure 2. The digit 6 displayed on the 7-segment display

## 5. Discussion

a. For this lab, our objective was to light up a 7-segment display with the binary input of a 4-bit DIP switch. The code starts with setting up the Arduino pins for inputs and outputs. For this lab, 3 port locations were used. In DDRB, DDRD, and DDRC, the pins were set up to be used as inputs or outputs, where 0 represents the input pins and 1 as the output pins.

b. The whole operations of this lab were put in a *while(1)* loop to make sure that the program kept on running. At the beginning of the while loop, an integer variable, number, is set to 0, to restart the value of the number every time the loop runs. The next series of *if* conditions is to check the input from the DIP switch. The DIP switch is connected so that the input resembles a 4-digit binary number. When the

binary matches a number from 0 to 9, the variable number will be changed to that number.

c. The next part of the lab is to light up the 7-segment display to match the value of the variable number. The LEDs for the 7-segment display were connected to 2 ports of the Arduino, B, and D. Therefore, to display the number using the 7-segment display, both PORTB and PORTD commands were needed. Another thing to take note of was the usage of *else if* to turn on the 7-segment display as opposed to *if*. This was done to make sure that only one number was displayed at a time.

## 6. Conclusion

In conclusion, this lab experiment was successful in achieving its goal of using an Arduino Uno and a C software to operate a 7-segment display to show the numbers 0 through 9. We successfully interacted with the 7-segment display to create numbers using the technique of C programming and the abilities of the Arduino platform.

Understanding this lab shows the practical use of microcontrollers in embedded systems as well as improves our understanding of C programming. This practical training is beneficial in a variety of real life scenarios where numerical displays are required, such as digital counters, timers, and clocks.

## 7. Reference

[1] Arduino. "Port Manipulation," [Online], Available: https://docs.arduino.cc/hacking/software/PortManipulation, Accessed: September 19, 2023.

[2] Arduino. "Pin Mapping for Arduino Uno," [Online], Available: https://docs.arduino.cc/hacking/hardware/PinMapping168, Accessed: September 19, 2023.

[3] N. Neumann, "Seven-Segment LED Displays," [Online], Available: https://www.egr.msu.edu/classes/ece480/capstone/fall10/group02/doc/Application_Note_-_Nick.pdf, Accessed: October 16, 2023.

## 8. Appendix

```
#include <avr/io.h> // header file for input output pins
#include <util/delay.h> // header file for delay
int main(void)
{
 DDRB = B00000011;
 DDRD = B11111100;
 DDRC = B00000000;
 int number;
 while(1)
 {
  number = 0;
  if (PINC & 0b00000001)
  {
   number = number + 1;
  }
  if (PINC & 0b00000010)
  {
   number = number + 2;
  }
  if (PINC & 0b00000100)
  {
   number = number + 4;
  }
  if (PINC & 0b00001000)
  {
   number = number + 8;
  }
  if (number == 1)//number 1
  {
   PORTD = 0b00011000;
```

```
  PORTB = 0b00000000;
}
else if(number == 2)//number 2
{
  PORTD = 0b01101100;
  PORTB = 0b00000001;
}
else if(number == 3)//number 3
{
  PORTD = 0b00111100;
  PORTB = 0b00000001;;
}
else if(number == 4)//number 4
{
  PORTD = 0b10011000;
  PORTB = 0b00000001;
}
else if(number == 5)//number 5
{
  PORTD = 0b10110100;
  PORTB = 0b00000001;
}
else if(number == 6)//number 6
{
  PORTD = 0b11110100;
  PORTB = 0b00000001;
}

else if(number == 7)//number 7
{
  PORTD = 0b00011100;
  PORTB = 0b00000000;
}

else if(number == 8)//number 8
{
```

`



```
      PORTD = 0b11111100;
      PORTB = 0b00000001;
    }
    else if(number == 9)//number 9
    {
      PORTD = 0b10011100;
      PORTB = 0b00000001;
    }
    else
    {
      PORTD = 0b11111100;
      PORTB = 0b00000000;
    }
  }
}
```