ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 351E

MICROCOMPUTER LABORATORY EXPERIMENT REPORT

EXPERIMENT NO	:	4
EXPERIMENT DATE	:	07.12.2020
LAB SESSION	:	MONDAY - 13.30
GROUP NO	:	G10

GROUP MEMBERS:

150180103 : EGEMEN GÜLSERLİLER

150180028 : ABDULKADİR PAZAR

150180716 : AHMET SELÇUK TUNÇER

AUTUMN 2020

Contents

1 INTRODUCTION

In this experiment, we generated various resistance values with the help of potentiometer, read these values from analog inputs and observe the related outputs on RGB LED and read analog inputs and used 7-segment display read inputs.

2 MATERIALS AND METHODS

2.1 Part 1

```
int GreenOPin = 3;
int BlueOPin = 5;
int RedOPin = 6;
int GreenIPin = A2;
int BlueIPin = A1;
int RedIPin = A0;
int Green, Red, Blue;

void setup() {
    pinMode(GreenOPin,OUTPUT);
    pinMode(RedOPin,OUTPUT);
    pinMode(BlueOPin,OUTPUT);
    pinMode(GreenIPin,INPUT);
    pinMode(BlueIPin,INPUT);
    pinMode(RedIPin,INPUT);
}
void loop() {
    Green = analogRead(GreenIPin);
    Blue = analogRead(BlueIPin);
    Red = analogRead(RedIPin);
    analogWrite(GreenOPin,Green/4);
    analogWrite(RedOPin,Red/4);
    analogWrite(BlueOPin,Blue/4);
    delay(100);
}
```

Figure 1: Code for Part 1

In this part we used the code above to display RGB lights with different colors and brightness. Since the range of the analogRead() function is between 0 and 1023 but the range of RGB values are between 0 and 255, we divided the value read by analogRead() by 4 before displaying the light.

2.2 Part 2

```
int analog, analogPin = A0, analog_digit;
int dout[10] = {1,79,18,6,76,36,32,15,0,4};
void setup()
{
    DDRC = 0b000000;
    DDRD = 0b11111111;
    DDRB = 0b111111;
}

void loop()
{
    analog = analogRead(analogPin);
    analog_digit = analog%10;
    PORTD = dout[analog_digit];
    delay(100);
}
```

Figure 2: Code for Part 2

In this part we used the code above to display the ones digit of the value read by analogRead() function. We obtained the values of dout array (the array used to keep values that are fed to seven segment display) by constructing the following table:

Number	A	B	C	D	E	F	G	Total
0	0	0	0	0	0	0	1	1
1	1	0	0	1	1	1	1	79
2	0	0	1	0	0	1	0	18
3	0	0	0	0	1	1	0	6
4	1	0	0	1	1	0	0	76
5	0	1	0	0	1	0	0	36
6	0	1	0	0	0	0	0	32
7	0	0	0	1	1	1	1	15
8	0	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0	4

2.3 Part 3

```
int analog, analogPin = A0, analog_digit;
int dout[10] = {1,79,18,6,76,36,32,15,0,4};
void setup()
{
    DDRC = 0b000000;
    DDRD = 0b11111111;
    DDRB = 0b11111111;
}
void fetch_last_digit()
{
    analog_digit = analog%10;
    analog = analog - analog_digit;
    analog = analog/10;
}
void loop()
{
    analog = analogRead(analogPin);
    PORTB = 14;
    fetch_last_digit();
    PORTD = dout[analog_digit];
    delay(15);
    PORTB = 13;
    fetch_last_digit();
    PORTD = dout[analog_digit];
    delay(15);
    PORTB = 11;
    fetch_last_digit();
    PORTD = dout[analog_digit];
    delay(15);
    PORTB = 7;
    fetch_last_digit();
    PORTD = dout[analog_digit];
    delay(15);
}
```

Figure 3: Code for Part 3

In this part we used the code above to display the value read by analogRead() function. We used the same dout array to keep the values fed to the seven segment display. For the ones digit, PORTB is set to 14 (0b00001110) so that rightmost seven segment display is active. Then fetch_last_digit() function is called to set variable analog_digit to ones digit. Finally member of the dout array corresponding to value of analog_digit is fed to seven segment display. We use a similar logic to display tens, hundreds and thousand digits of the number on four seven segment displays.

3 RESULTS

3.1 Part 1

At the end of the first task, the team is succeeded in displaying RGB lights with different brightness and colors. Implementation of this part is mentioned in detail at the material and methods section. Basically, resistance values are read from the analog input

pins and depending on those inputs RGB lights that have different brightness and colors are displayed.

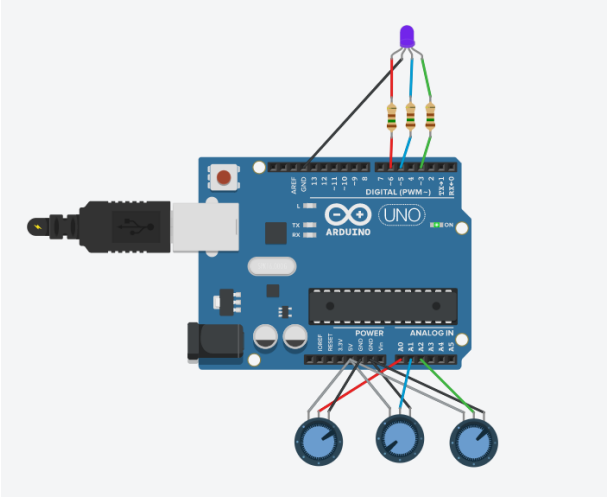

Figure 4: Result of Part 1

3.2 Part 2

In the second part, the team implemented a program that enable us to observe the last digit of analog input value on a 7-segment display. Implementation of this part is mentioned in detail at the material and methods section. Simply, analog input value is read and by calculating mod 10 the number first digit of the number is displayed on the seven segment display.

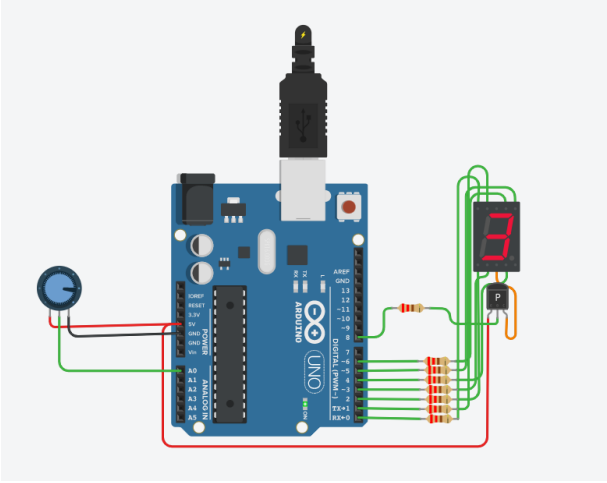

Figure 5: Result of Part 2

3.3 Part 3

In the last part of the experiment, 4 digit 7-segment display which is very similar to part 2 is implemented. Implementation of this part is mentioned in detail at the material and methods section. Similarly, at first analog input value is read. Then each digit is displayed on the 7-segment display by fetching the number and displaying digits one by one with a small delay value.

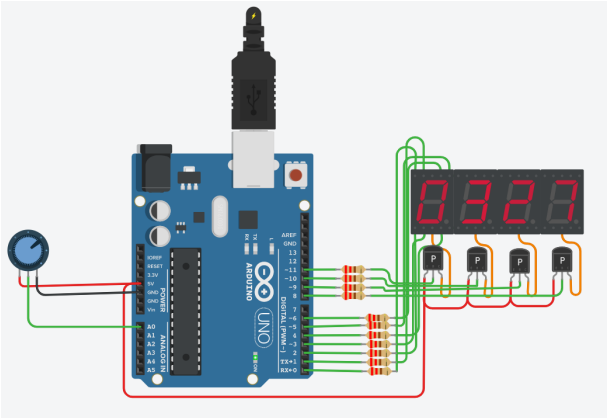

Figure 6: Result of Part 3

4 DISCUSSION

In the first part, pinMode(), analogWrite() and analogRead() functions that are provided by Arduino environment were used and this made our task easy because the methods that are expected to achieve by us was easy to handle compared to other tasks.

In the second part, analog input was read by analogRead() functions but digital output values are assigned by port manipulation. Rx and Tx port caused us to spend short period of time to understand their serial features.This part was a small, easy and instructive task.

In the last part, methodology is very similar to part 2. Two functions; fetch_last_digit() and display() were created in order to reduce complexity. And the lighting up all of the digits at once is possible by showing all digits one after the other rapidly. Which is enough to trick human eye to think all of them are lit at the same time. This is how all of the screens work in today's world. Old televisions used to scan the image and display it on the screen as little parts going from top left to top right then second line and so on.

5 CONCLUSION

In this experiment all together, we have learned how to get and fetch analog inputs and that it is possible to use 3rd, 5th, 6th digital pins as analog output pins. While implementing the second experiment, we wanted to check if our circuit was returning the correct output that we were expecting so we tried to use the serial monitor to print the analog input value. But we couldn't both get the expected output voltage from the pin and the serial output from the serial monitor which led us to searching why and ultimately to finding out that registers at pins D1 and D2 are used for USB to serial conversion therefore they can't be used simultaneously for both[1]. As known, there exists signals that has to be processed in the analog form (such as voice signals via a microphone) and this experiment is a stepping stone in understanding the process of working with analog signals. To sum up we learned how to read analog inputs and how to process them to give desired outputs in this experiment.

REFERENCES

[1] Kushagra Keshari. Why do rx and tx blink on arduino? https://www.quora.com/Why-do-RX-and-TX-blink-on-Arduino, 2019 (accessed December 4, 2020).