

Student Name
Naomi Thing

Student Number
Np03cs4a220491

Portfolio Introduction

Workshop Activities 50% Weighting

Mini Project 50% Weighting

**This completed portfolio will need submitting to Canvas by the
due date.**

Questions please email

Dr Sarah Slater

s.i.slater@wlv.ac.uk

Portfolio

Contents

Workbook 1	4
Activity 1.1: Actual voltage across 5V breadboard pins.....	4
Activity 1.2: Actual voltage across 3.3V breadboard pins.	4
Activity 1.3: Potential Divider Calculations	
4	
Activity 1.4: 3V Calculations from either the 5V supply or 3.3V supply	5

Activity 1.5: Voltage Divider circuit readings from Breadboard circuit.	7
Activity 1.6: LED Circuits.....	8
Activity 1.7: Current Measurement	9
Activity 1.8: Fritzing for 4 switches & LEDS	10
Activity 1.9: Fritzing for Number 0-7	11
Workbook 2	13
Activity 2.1: LED Flashing to show decimal number 63 as binary.	13
Activity 2.2: 4 LED's for counting up in binary from 0 to 15.	15
Activity 2.3: Traffic Lights	18
Workbook 3	21
Activity 3.1: Circuit Diagram of Button & LED	21
Activity 3.2: 3 Switches & Led	22
Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS)	24
Workbook 4	26
Activity 4.1: Serial Port	26
Activity 4.2: Serial Port binary to decimal	29
Activity 4.3: Calibrating Analogue Information.....	32
Activity 4.4: Temperature Sensor & Serial Port	34
Workbook 5	36
Activity 5.1: RGB Led and switches	36
Activity 5.2: Distance Sensor	38

Activity 5.3: 1602 LCD Display	40
Workbook 6	43
Activity 6.1: PWM	
43 Workbook 7	44
Activity 7.1: 2 Arduinos – using Digital Pins	44
Activity 7.2: 2 Arduinos – using Serial I/O	47
Workbook 8	49
Activity 8.1: Stepper Motor Circuit Diagram	
49	
Activity 8.2: 2 Stepper Motors	
51	
Workbook 9	52
Activity 9.1: Windscreen Wiper Code using Servos & Temperature Sensor	52
Individual Project (50%)	54
Rationale	
54	
Timescales	
54	
Equipment	
54	
The Project	
54	
Step 1 produce a detailed description of your project.	
54	
Step 2 Circuit Diagram&Fritzing Schematic	
54	
Step 3 A Program	
55	
Step 4 Testing	
55	

Step 5 Conclusions	55
Layout	55
Marking	55
All sections carry equal marks.	55

If you prefer, you may use Tinkercad to show a component layout, rather than a circuit Diagram in Fritzing or other circuit design software, though a circuit diagram is more useful as this is what you would most likely see if you were working on embedded systems.

Workbook 1

Activity 1.1: Actual voltage across 5V breadboard pins.

Voltage: 5.12V

Activity 1.2: Actual voltage across 3.3V breadboard pins.

Voltage: 3.29

Explain in around 100 words why you think the value read by a multi meter on a circuit, may be different to a simulator value such as TinkerCad.

When measuring values on a circuit using a multi-meter, the readings may vary or be imprecise due to external factors such as sound energy, weather, and light energy that can affect the hardware. However, simulators like TinkerCad are not affected by these environmental disturbances, allowing the multi-meter to provide accurate readings in a simulated environment. Therefore, using a real multi-meter may yield values that are close to the exact value, but not necessarily the exact value, while in TinkerCad, the multi-meter can provide the exact value.

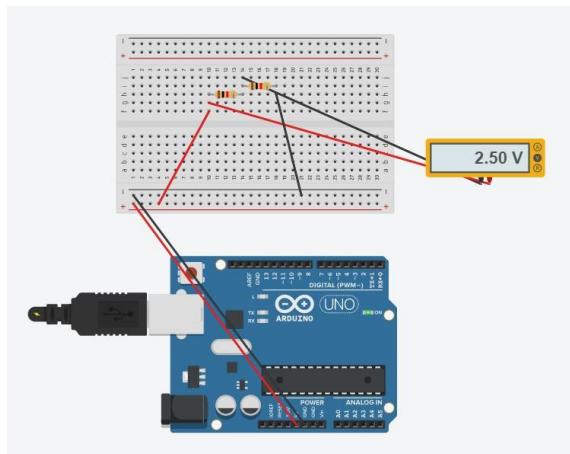
If the read value is 4.84V on a 5V supply, what would be a sensible tolerance to quote, explain your answer.

If the read value is 4.84 on a 5V supply, the tolerance will be:

$$\{(5-4.84)/5\} \times 100\% \text{ or } -0.16$$

Activity 1.3: Potential Divider Calculations

Show the working on how you achieved 2.5V



Calculation:

Here,

$$R_1 = R_2 = ?$$

$$\text{Voltage(V)} = 5$$

$$RT = R_1 + R_2 = 2R_1$$

$$\text{Voltage Drop} = (R_1/RT) * V$$

$$= (R_1/2R_1) * V$$

$$= 0.5 * 5$$

$$= 2.5$$

Thus, Voltage Drop is 2.5V

Activity 1.4: 3V Calculations from either the 5V supply or 3.3V supply

Answer:

Calculation:

Here,

$$R_1 = 220 \text{ ohm}$$

$$R_2 = ?$$

Voltage(V)=5 Voltage

Drop (VD1)=3V Voltage

Drop (VD2)=?

$$VD1 = (R1/R2) * V$$

$$3 = 3/220 + R2 * 5$$

$$600 + 3R2 = 220 * 5$$

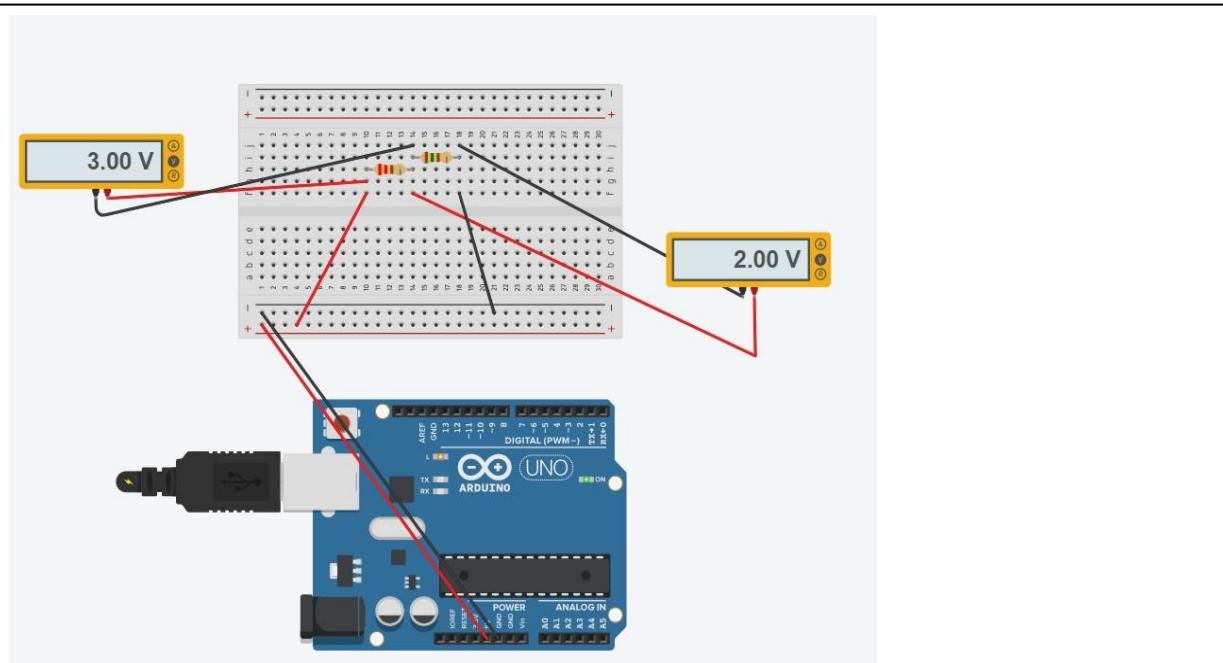
$$R2 = (1100 - 600) / 3$$

- $R2 = 146.6 \text{ OR } 147 \text{ ohm}$ $VD2 = R2 / RT * V$

$$= 147 / (200 + 147) * 5$$

- $VD2 = 2V$

Activity 1.5: Voltage Divider circuit readings from Breadboard circuit.



Calculation:

Here,

$$R_1 = 220 \text{ ohm}$$

$$R_2 = ?$$

$$\text{Voltage(V)} = 5$$

$$\text{Voltage Drop of } R_1(\text{VD1}) = 3\text{v}$$

$$\text{Voltage Drop pf } R_2(\text{VD2}) = ?$$

$$\text{VD1} = (R_1/R_2) * V$$

$$3 = 3/220 + R_2 * 5$$

$$600 + 3R_2 = 220 * 5$$

$$R_2 = (1100 - 600) / 3$$

- R₂=146.6 or 147 ohm

$$\text{VD2} = R_2 / R_T * V$$

Activity 1.6: LED Circuits

Each resistor Value

2150 ohm

2150 ohm

Total resistance Calculation

Calculation:

Here,

$R1=2.15 \text{ Kohm}$, $R2=2.15 \text{ Kohm}$ (Parallel Connection)

$1/\text{Total Resistance}=1/R1+1/R2$

$1/\text{Total Resistance}=1/2.15+1/2.15$

$\text{Total Resistance}=2.15/2$

$\text{Total Resistance}=1075 \text{ ohm}$

Measured Resistance

$R1=R2=2160 \text{ ohm}$

If measured resistance is not the same, why not? If you simulated this, why might the real value be different.

The measured resistance value on a multimeter may differ from the labeled number due to the influence of several factors such as humidity, sound energy, light energy, weather conditions, among others. Therefore, the multi-meter does not provide an accurate value that matches the labeled value.

Activity 1.7: Current Measurement

Calculation of current flowing into LED

Here,

V=5V

I=?

R=1000 ohm

$I=V/R=5/1000$

I=0.004A

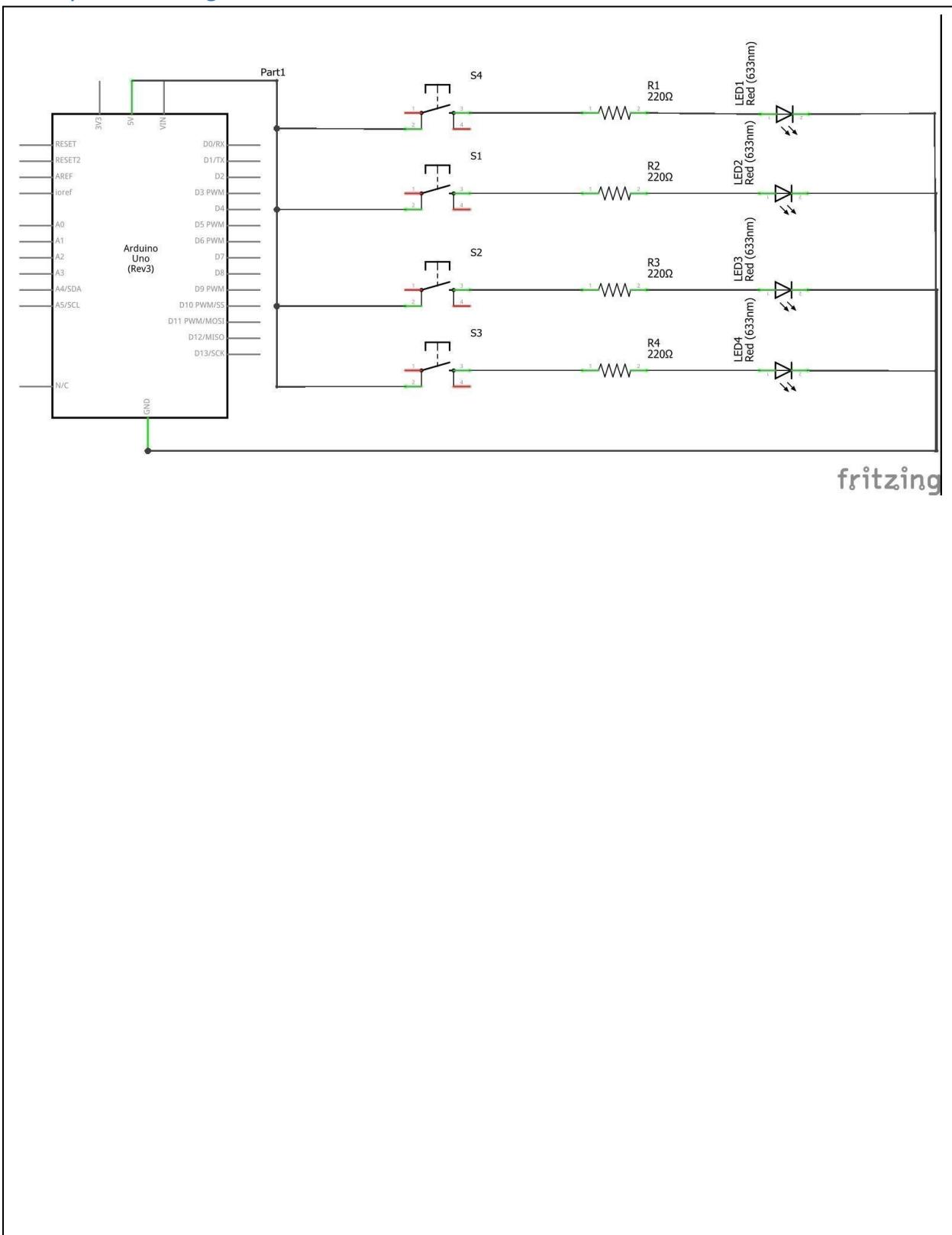
Actual measured value of current

0.003A

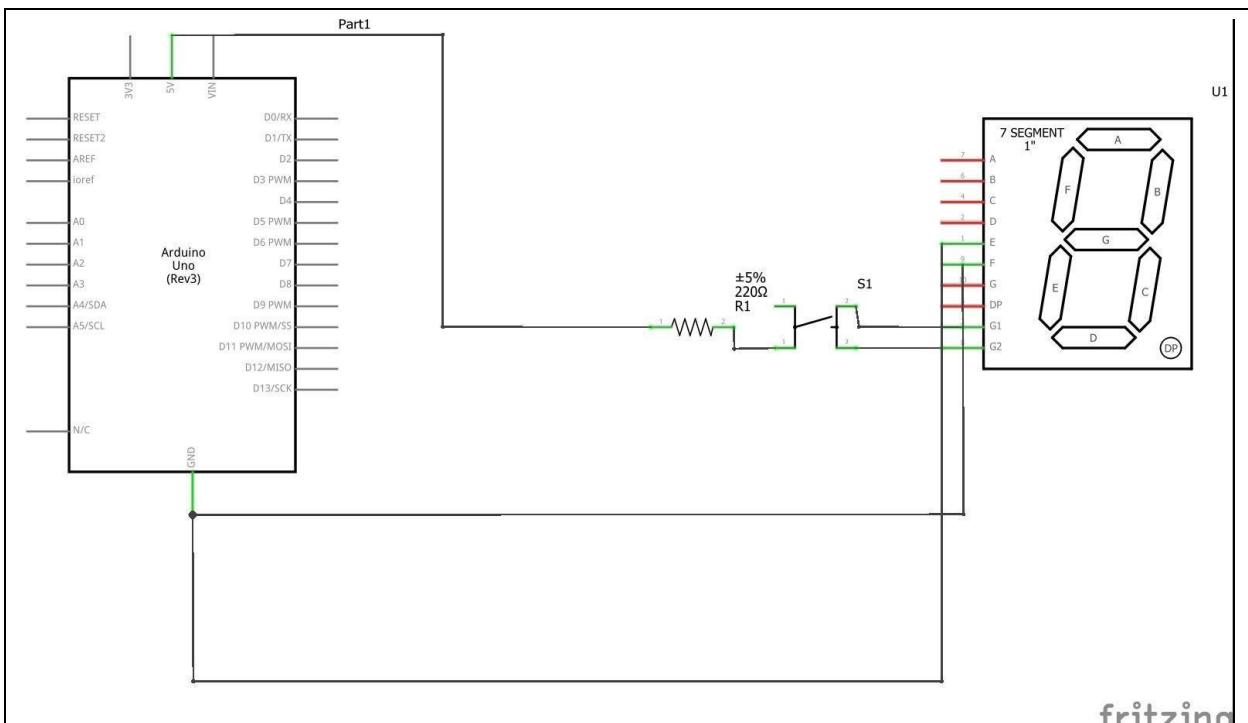
Why might they be different?

Environmental factors such as humidity, weather conditions, static electricity, light energy, sound energy, and others can influence the current flow and multi-meter, causing a difference between the calculated and actual values.

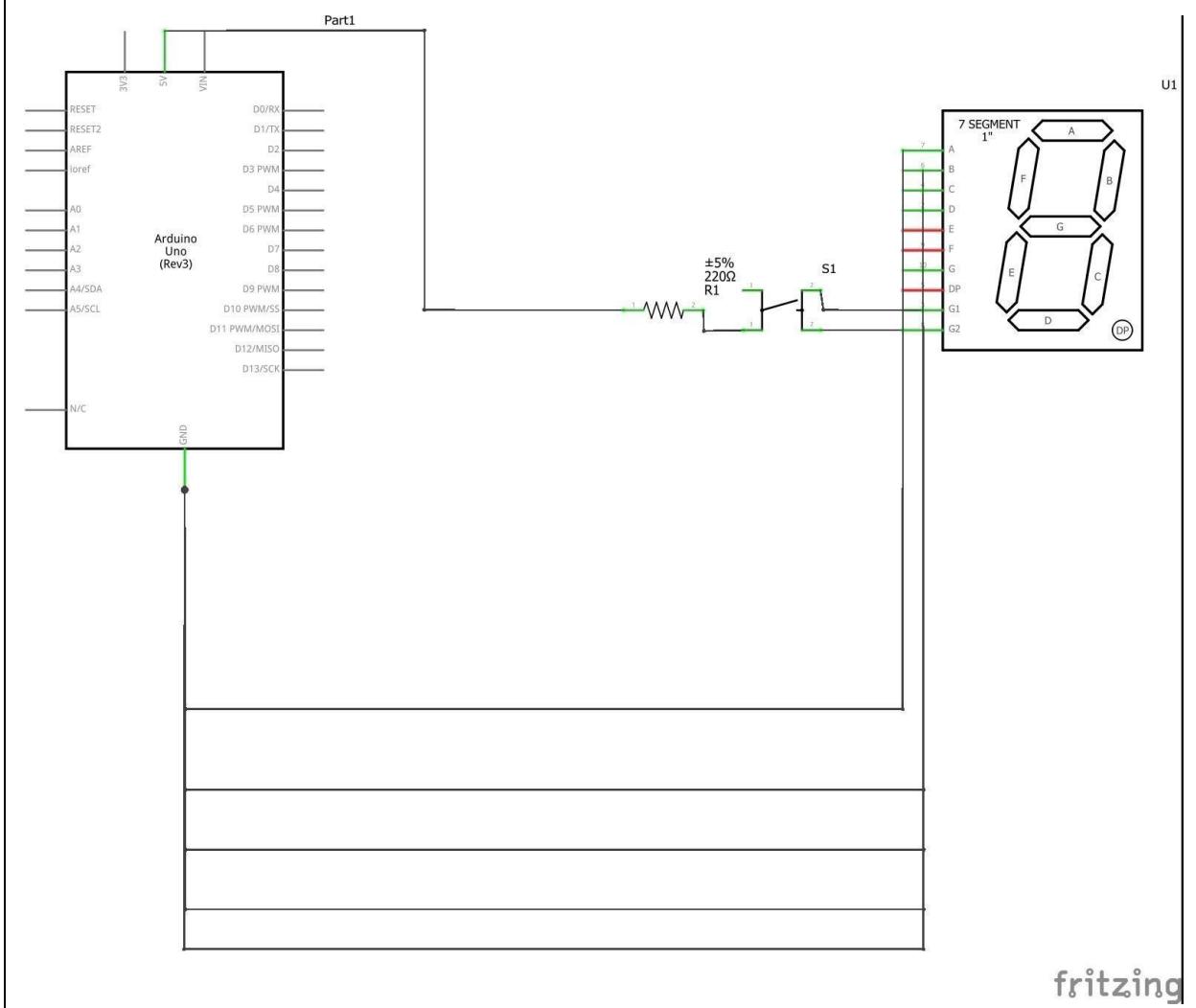
Activity 1.8: Fritzing for 4 switches & LEDs



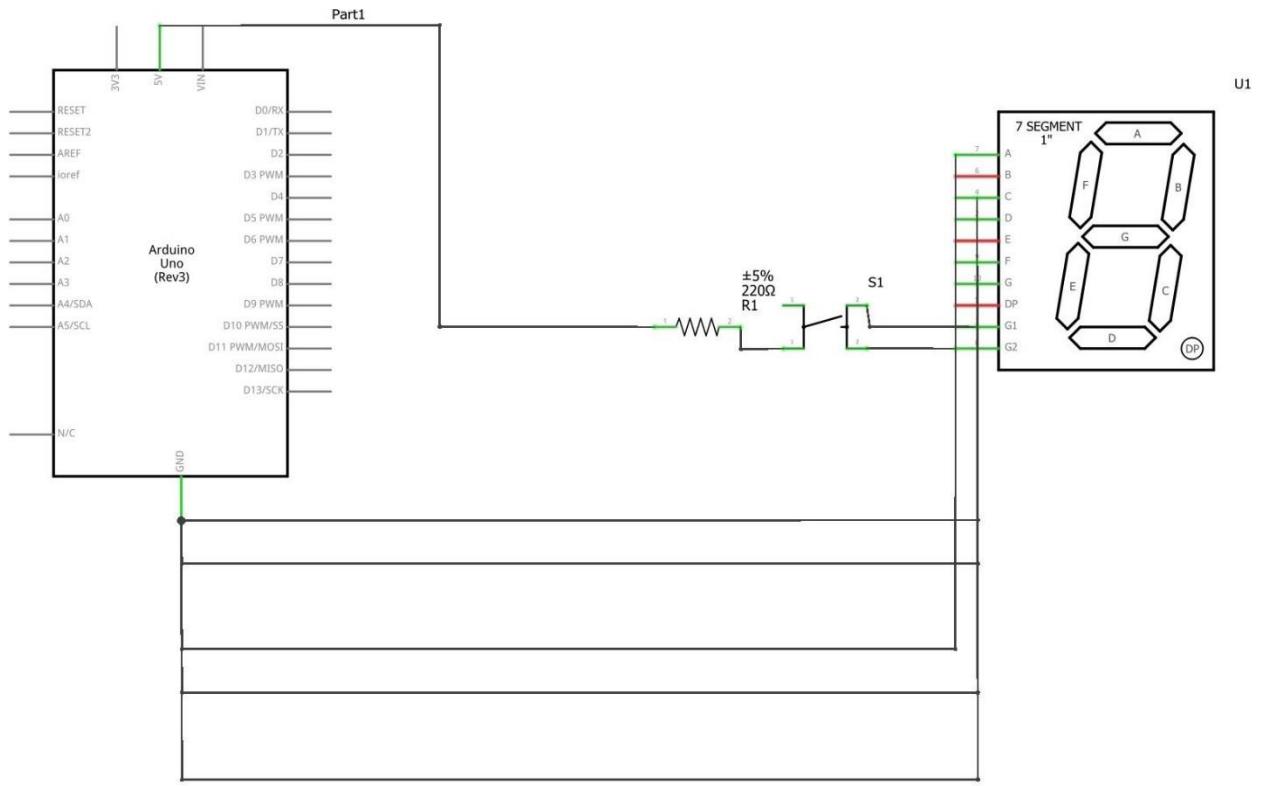
Activity 1.9: Fritzing for Number 0-7



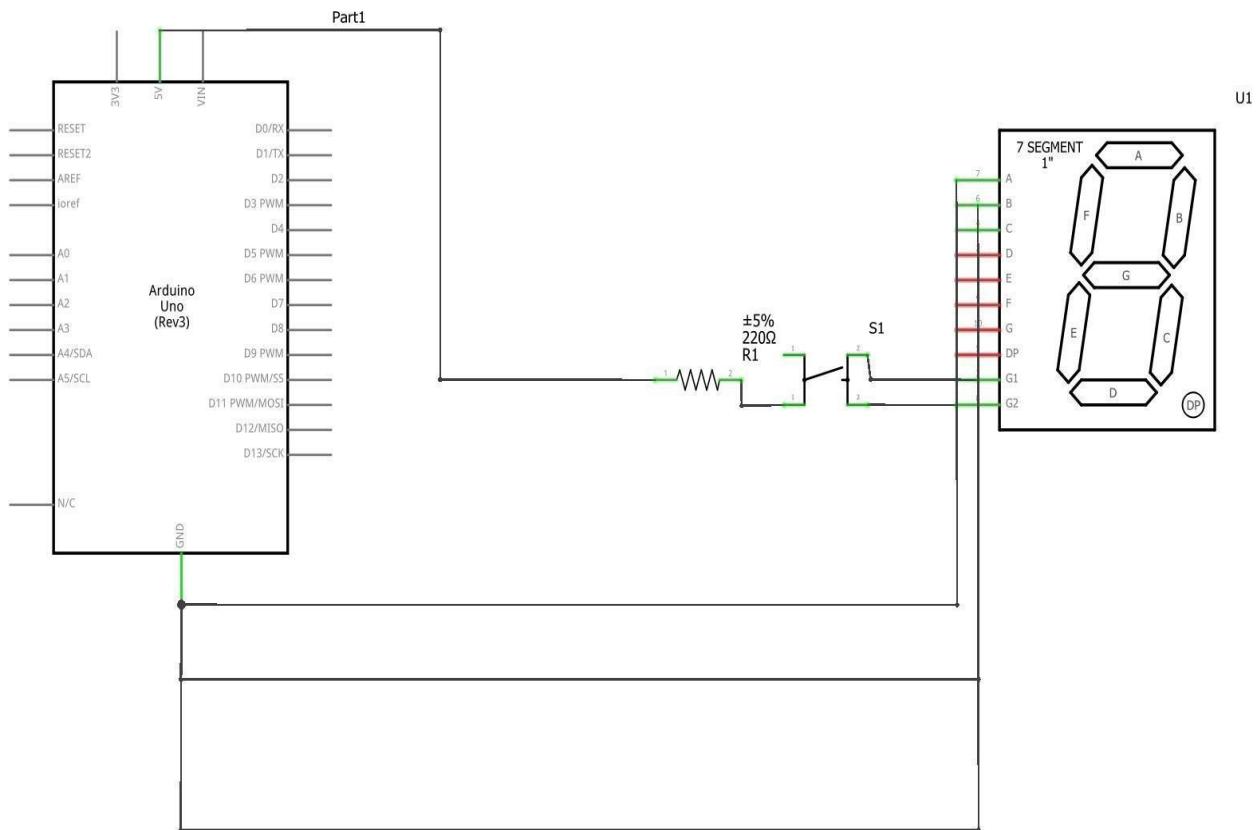
fritzing



fritzing



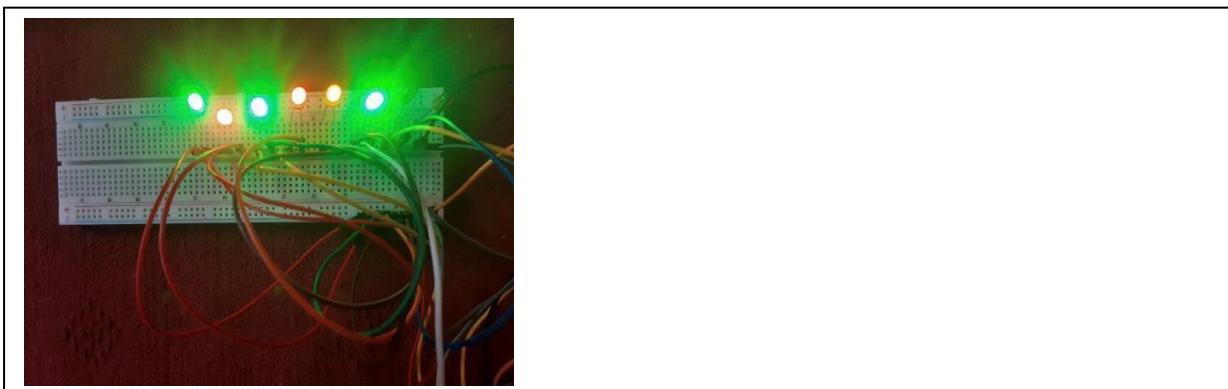
fritzing



fritzing

Workbook 2

Activity 2.1: LED Flashing to show decimal number 63 as binary.



63 as binary, including working

$$(63)_{10} = (111111)_2$$

Copy & post your code with a suitable comment at the top of code with your name & student number

□

```
//name: Naomi Thing
//student ID: 2332244
int led1=13; //assigning 13 number to led1 for easier access of 13 number pin
int led2=12; //assigning 12 number to led1 for easier access of 12 number pin
int led3=11; //assigning 11 number to led1 for easier access of 11 number pin
int led4=10; //assigning 10 number to led1 for easier access of 10 number pin
int led5=9; //assigning 9 number to led1 for easier access of 9 number pin
int led6=8; //assigning 8 number to led1 for easier access of 8 number pin
int time=1000; //assigning 1000 to time variable for controlling delay time
void setup()
{ //selecting pinmode for leds
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
pinMode(led5, OUTPUT);
pinMode(led6, OUTPUT);
}

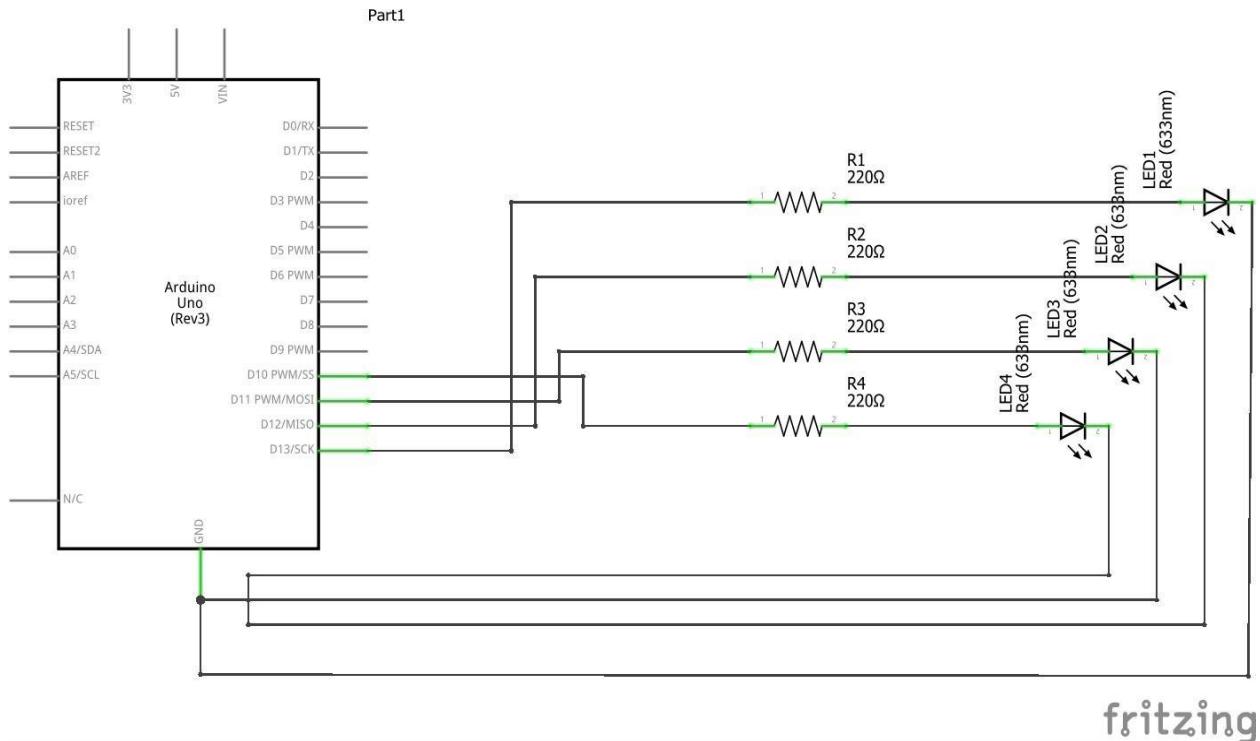
void loop()
{ // HIGH command given to light up leds inserted in the repective digitalpins
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);

digitalWrite(led5, HIGH);
digitalWrite(led6, HIGH);
delay(time); //1000ms or 1 second delay kept for holding the lights on for 1 second
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);

digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
digitalWrite(led5, LOW);
digitalWrite(led6, LOW);
delay(time);
}
```

Activity 2.2: 4 LED's for counting up in binary from 0 to 15.

Fritzing Circuit diagram for Step 4 i.e. 4 LEDs



Arduino Program for Step 4 i.e. 4 LEDs

```
//name: Naomi Thing
//student ID: 2332244
int led1=13; //initializing led1, led2, led3 and led4 as the written numbers for accessing the pins easily
int led2=12;
int led3=11;
int led4=10;
int time=5000;//initializing time for time to be kept in delay
void setup()
{
pinMode(led1, OUTPUT); //pinmode of four leds selected as output as light is to be emitted
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
}

void loop()
[
//1.
digitalWrite(led4, HIGH); //only the fourth led lightened as binary equivalent of dec no. 1 is 1
delay(time);
digitalWrite(led4, LOW); //led turned off for avoiding overlapping the codes stated below
//2.
digitalWrite(led3, HIGH); // third led lightened
delay(time);
digitalWrite(led3, LOW);// led turned off for avoiding overlapping the codes stated below
```

```
//2.
digitalWrite(led3, HIGH); // third led lightened
delay(time);
digitalWrite(led3, LOW); // led turned off for avoiding overlapping the codes stated below
//Similarly, all the leds are turned on equivalent to binary equivalency of dec numbers
//3.
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
delay(time);
digitalWrite(led3, LOW);

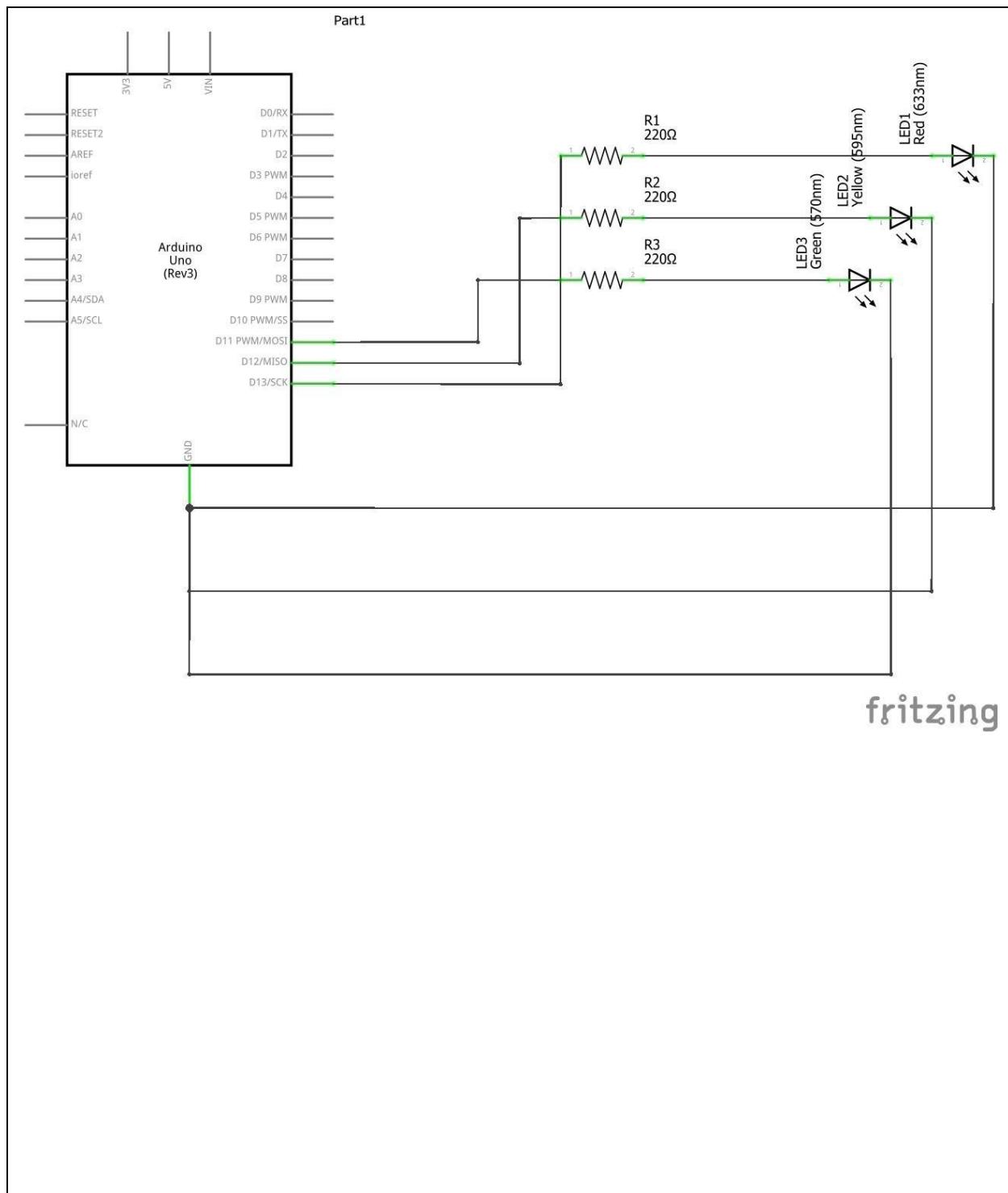
digitalWrite(led4, LOW);
//4.
digitalWrite(led2, HIGH);
delay(time);
digitalWrite(led2, LOW);
//5.
digitalWrite(led2, HIGH);
digitalWrite(led4, HIGH);
delay(time);
digitalWrite(led2, LOW);
digitalWrite(led4, LOW);
//6.
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
```

```
//7.  
digitalWrite(led2, HIGH);  
digitalWrite(led3, HIGH);  
digitalWrite(led4, HIGH);  
delay(time);  
digitalWrite(led2, LOW);  
digitalWrite(led3, LOW);  
digitalWrite(led4, LOW);  
//8.  
digitalWrite(led1, HIGH);  
delay(time);  
digitalWrite(led1, LOW);  
  
//9.  
digitalWrite(led1, HIGH);  
digitalWrite(led4, HIGH);  
delay(time);  
digitalWrite(led1, LOW);  
digitalWrite(led4, LOW);  
//10.  
digitalWrite(led1, HIGH);  
digitalWrite(led3, HIGH);  
delay(time);  
digitalWrite(led1, LOW);  
digitalWrite(led3, LOW);
```

```
//11.  
digitalWrite(led1, HIGH);  
digitalWrite(led3, HIGH);  
digitalWrite(led4, HIGH);  
delay(time);  
digitalWrite(led1, LOW);  
digitalWrite(led3, LOW);  
digitalWrite(led4, LOW);  
//12.  
digitalWrite(led1, HIGH);  
digitalWrite(led2, HIGH);  
delay(time);  
digitalWrite(led1, LOW);  
digitalWrite(led2, LOW);  
//13.  
digitalWrite(led1, HIGH);  
digitalWrite(led2, HIGH);  
  
digitalWrite(led4, HIGH);  
delay(time);  
digitalWrite(led1, LOW);  
digitalWrite(led2, LOW);  
digitalWrite(led4, LOW);  
//14.  
digitalWrite(led1, HIGH);
```

```
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
delay(time);
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
//15.
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
delay(time);
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
}
```

Activity 2.3: Traffic Lights



Fritzing Circuit diagram for Traffic Lights Arduino Program for Traffic Lights

```
//name: Naomi Thing
//student ID: 2332244
int redLed=13; //Initializing digital pin number of the respective leds
int amberLed=12;
int greenLed=11;
int time=500; //Initializing time to hold the state of led using time variable
void setup()
{
pinMode(redLed, OUTPUT); // Selecting the pinmode for all three leds in traffic light
pinMode(amberLed, OUTPUT);
pinMode(greenLed, OUTPUT);
}

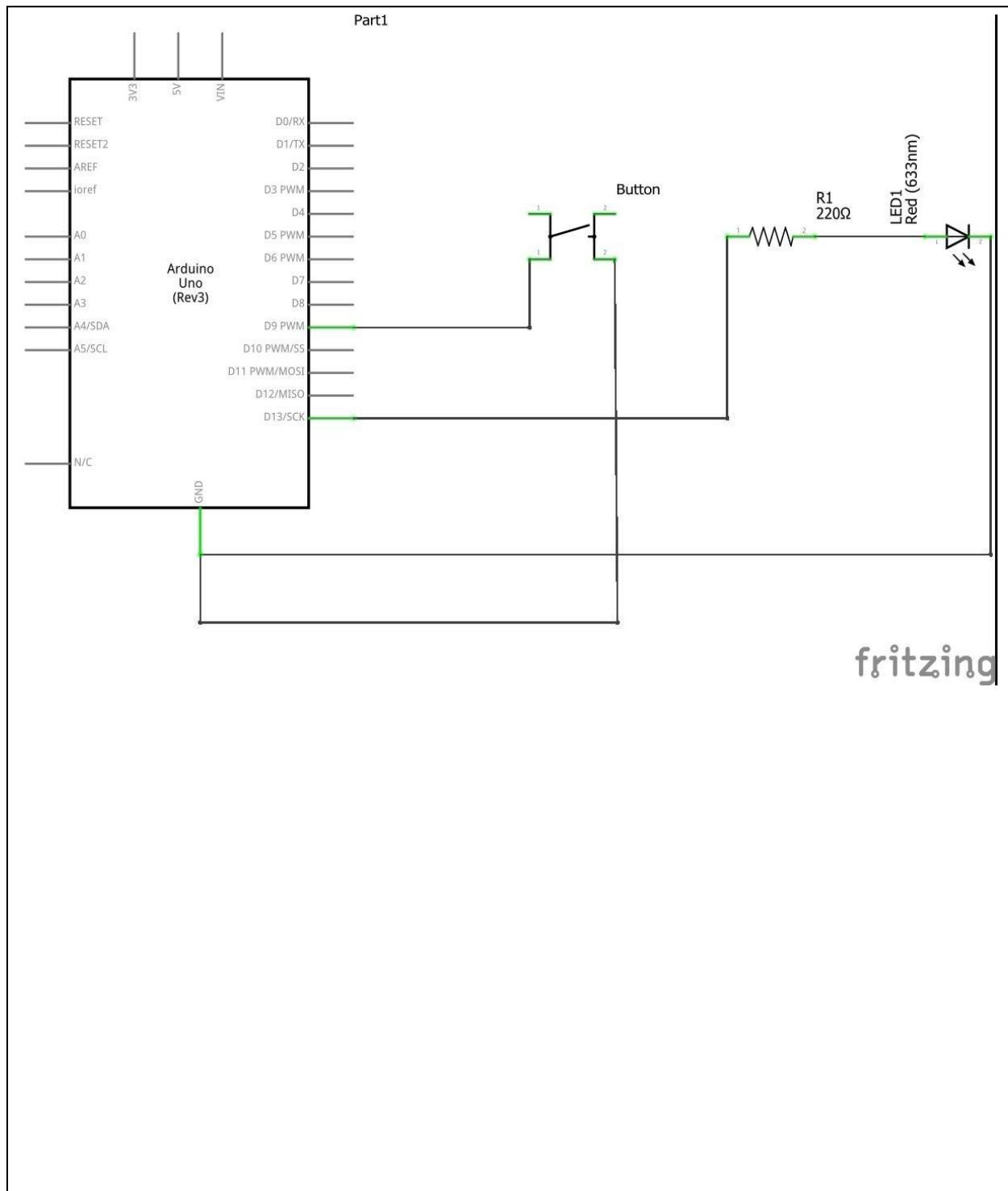
void loop()
{
for(int i=0;i<5;i++){ //Using a for loop for blinking red led 5 times
digitalWrite(redLed, HIGH); //Used for turning the led on
delay(time); //holding the on state of led for 500ms or 0.5 seconds
digitalWrite(redLed, LOW); //Used for turning off the led
delay(time); holding the off state for 500ms or 0.5 seconds
}
for(int i=0;i<5;i++){ //Using for loop for blinking yellow led 5 times
digitalWrite(amberLed, HIGH);
delay(time);
digitalWrite(amberLed, LOW);

delay(time);
}
for(int i=0;i<5;i++){ //Using for loop for blinking green led 5 times
digitalWrite(greenLed, HIGH);

delay(time);
digitalWrite(greenLed, LOW);
delay(time);
}
for(int i=0;i<5;i++){ //Using for loop for blinking yellow/amber led 5 times
digitalWrite(amberLed, HIGH);
delay(time);
digitalWrite(amberLed, LOW);
delay(time);
}
}
```

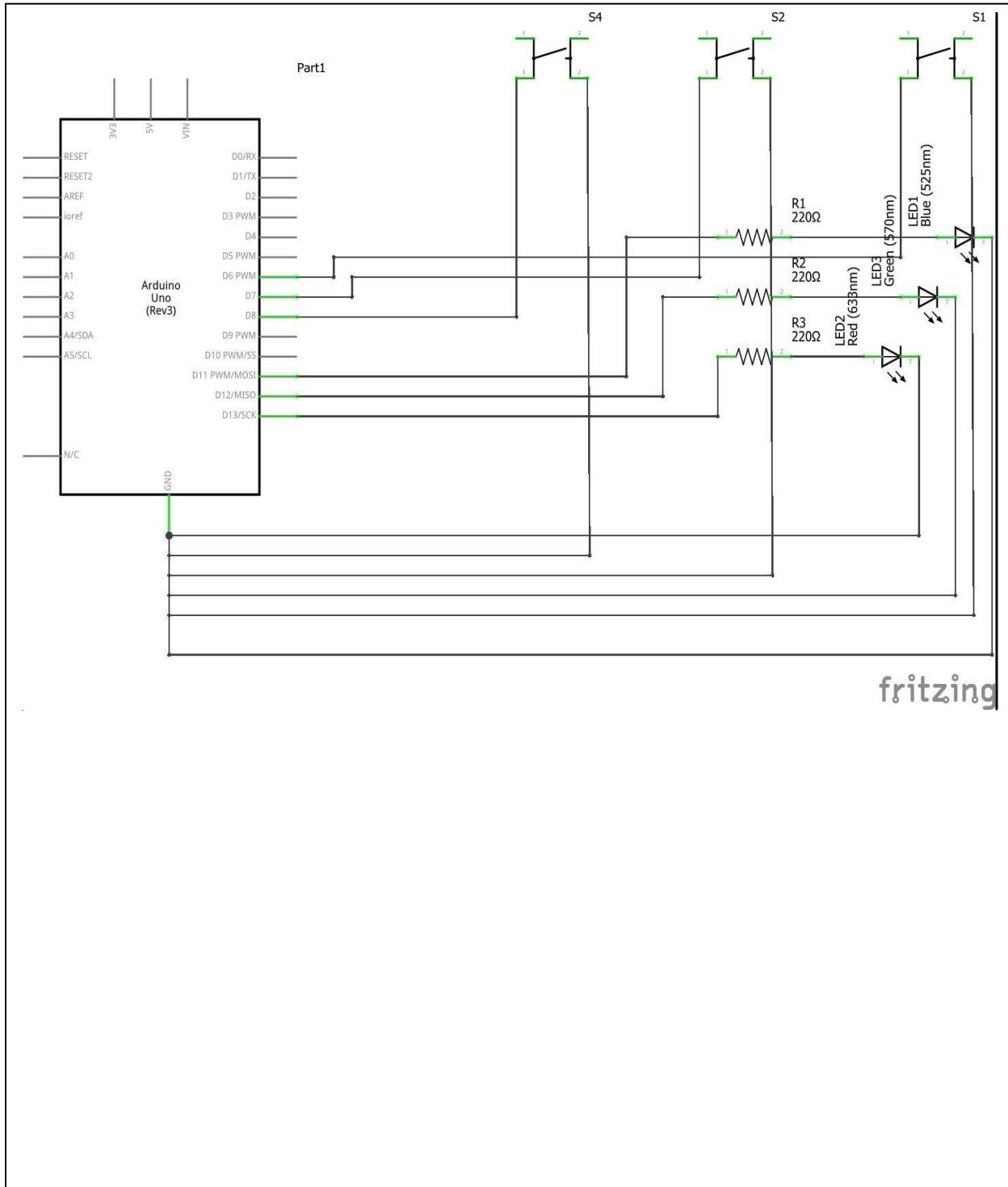
Workbook 3

Activity 3.1: Circuit Diagram of Button & LED



Fritzing

Activity 3.2: 3 Switches & Led



Fritzing Circuit Diagram (floating points handled using external resistors) Arduino Program

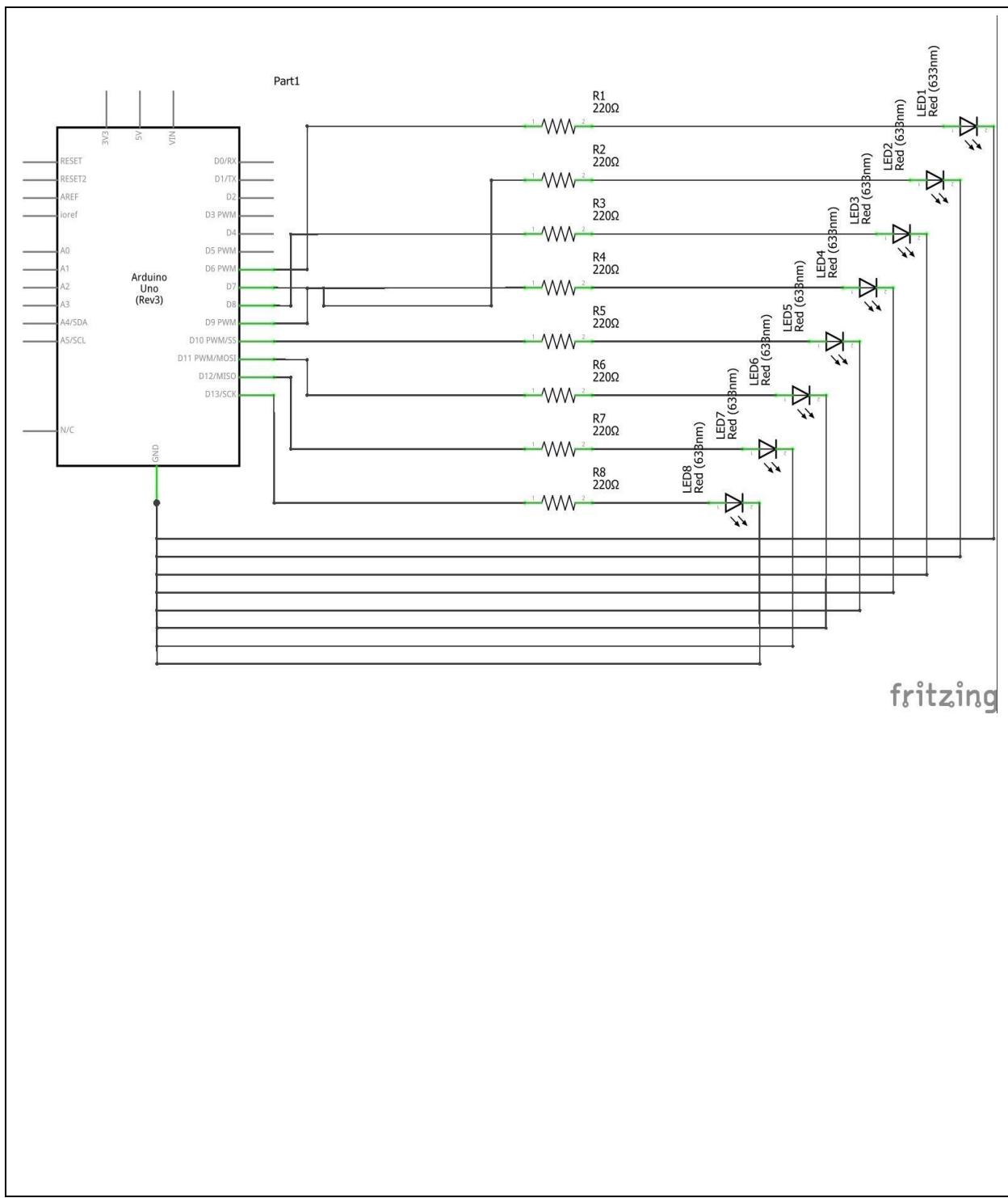
```
//name: Naomi Thing
//student ID: 2332244
int redLed=13; //Initializing three variables with three digital pin numbers for leds
int blueled=12;
int yellowled=11;
int switchInput1=9; //initializing three variables with three digital pin numbers for switch
int switchInput2=8;
int switchInput3=7;
void setup()
{
    pinMode(redLed, OUTPUT); //Selecting the pinmode of leds as output
    pinMode(blueled,OUTPUT);
    pinMode(yellowled,OUTPUT);
    pinMode(switchInput1,INPUT_PULLUP); /*Selecting the pinmode of switchinputs as input_pullup to control the floating values
    coming from the digital pins using internal resistor of the arduino */
    pinMode(switchInput2,INPUT_PULLUP);
    pinMode(switchInput3,INPUT_PULLUP);
}
void loop()
{
    int switchState1=digitalRead(switchInput1); //reading the state of digitalpin i.e. either 0 or 1
    //since we are using input_pullup the default state coming from the digital pin is 1(HIGH)
    int switchState2=digitalRead(switchInput2); //reading the state of digitalpin
    int switchState3=digitalRead(switchInput3); //reading the state of digitalpin
    if(switchState1==LOW) //if switch1 is pressed codes are to be executed

    {
        digitalWrite(redLed,HIGH); //redled is turned on
        delay(1000); //holds the on state for 1000ms or 1 second
    }
    else //if switch1 is not pressed
    {
        digitalWrite(redLed,LOW); //redled remains off
        delay(1000); //holds the off state
    }

    if(switchState2==LOW) //if switch2 is pressed the given codes are to be executed
    {
        digitalWrite(blueled,HIGH); //blueled is turned on
        delay(1000);
    }
    else //if switch2 is not pressed
    {
        digitalWrite(blueled,LOW); //blueled is turned off
        delay(1000);
    }

    if(switchState3==LOW) // switch3 is pressed the given codes are to be executed
    {
        digitalWrite(yellowled,HIGH); //yellow led is turned on
        delay(1000);
    }
    else //if switch3 is not pressed
    {
        digitalWrite(yellowled,LOW); //yellow led is turned off
        delay(1000);
    }
}
```

Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS)



Fritzing

Arduino Program

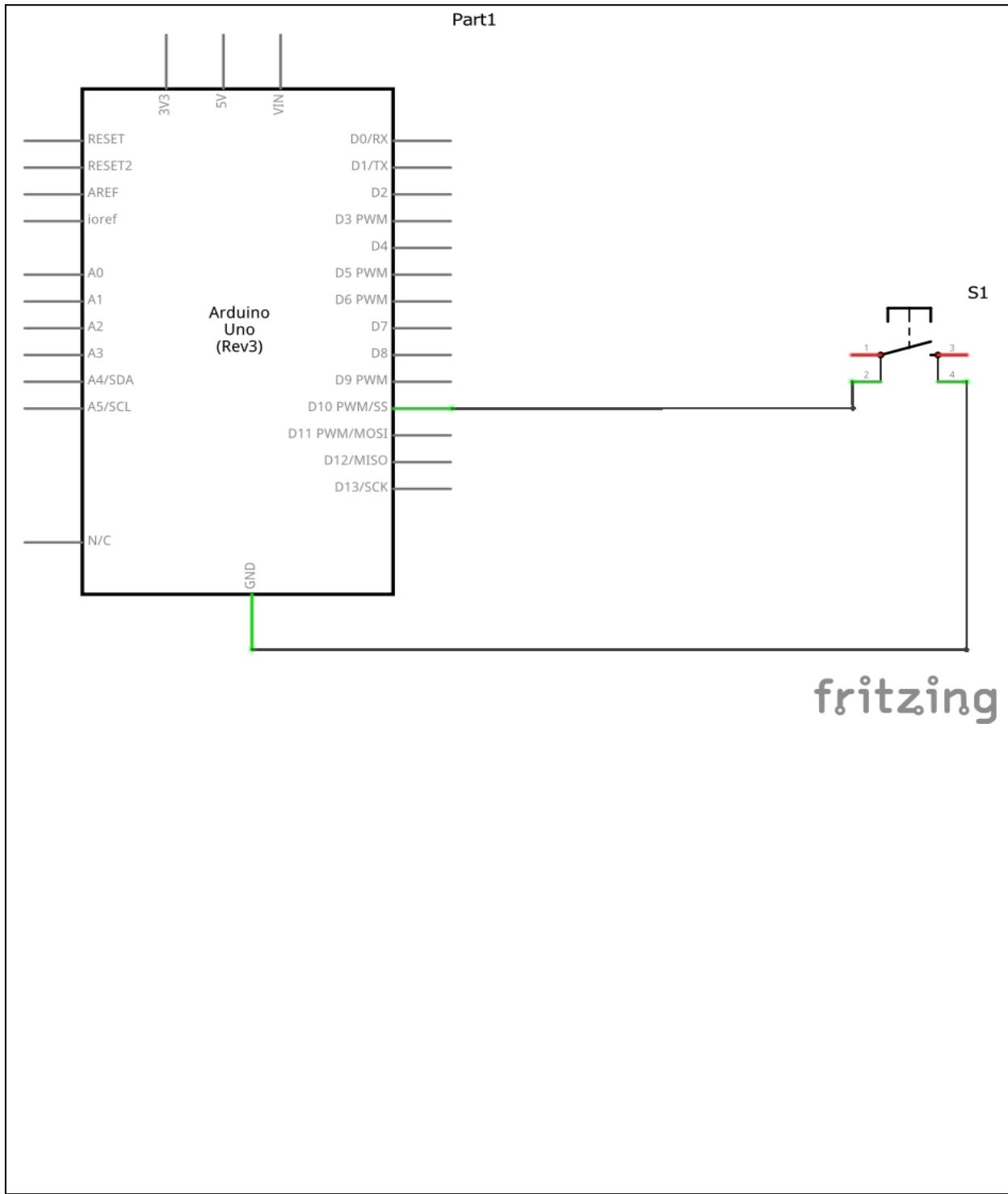
```
//name: Naomi Thing
//student ID: 2332244
int led[8]={6,7,8,9,10,11,12,13}; 8 digital pin numbers stored in array named led
int val=22; //decimal value 22 initialized in val variable for bin conversion
void setup()
{
for(int i=0;i<8;i++){
pinMode(led[i],OUTPUT); //pinmode selected for the digital pins of 'led' array using loop
}
}

void loop()
{
dis(val); //dis(val) function called
}
void dis(int val){ //dis(val) function made for transforming decimal num in 'val' to binary equivalent
for(int i=0;i<8;i++)//used for finding the binary equivalent of val
if(val%2==0){// condition: if val/2 gives remainder 0 then the bin equivalent is 0 thus led is off
digitalWrite(led[i],LOW);
}
else{ if the remainder is 1 then the bin equivalent is 1 thus led is turned on
digitalWrite(led[i],HIGH);
}
val=val/2; //val variable is divided by 2 for correct execution of conversion
}

}
```

Workbook 4

Activity 4.1: Serial Port



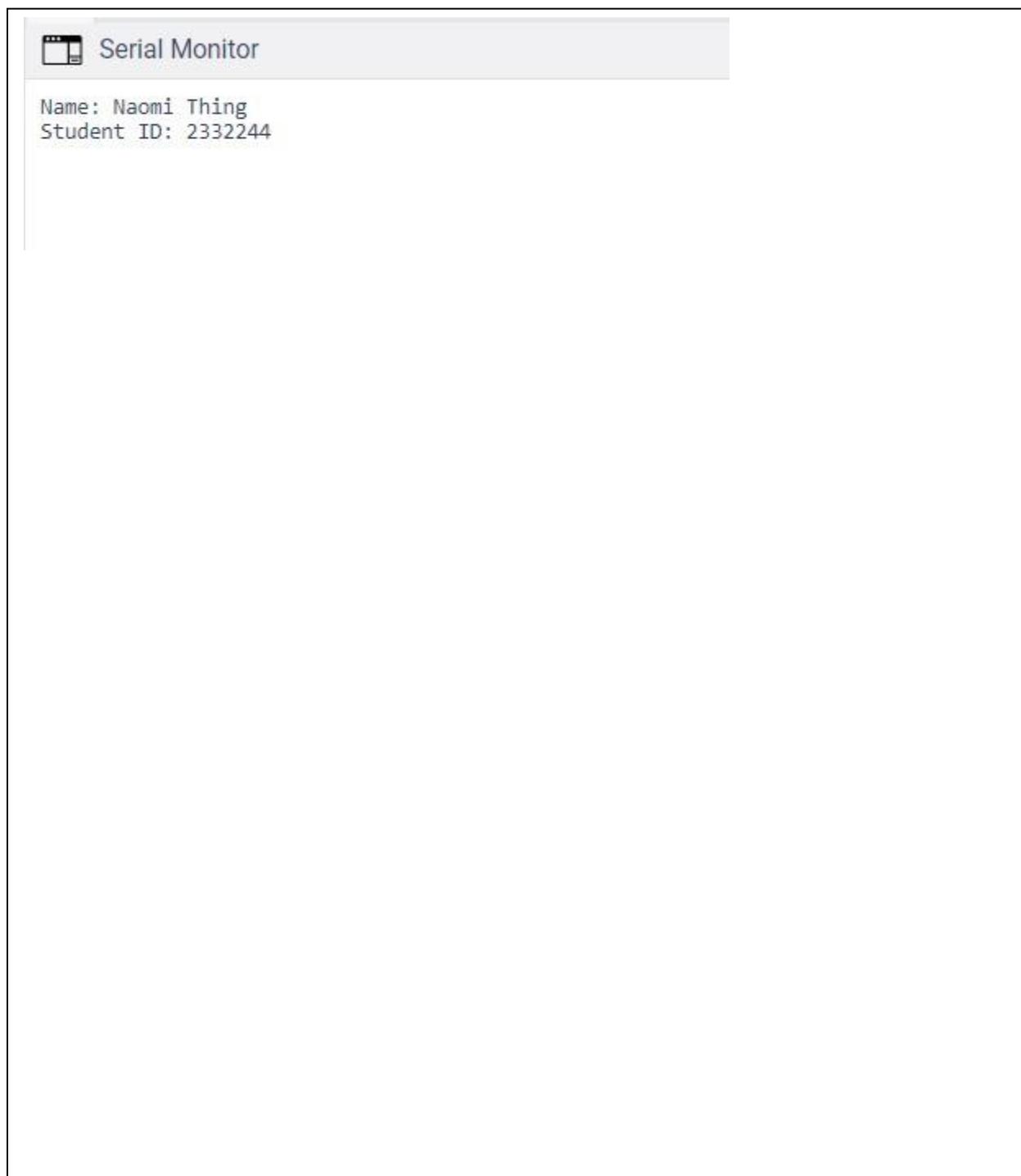
Fritzing

Arduino Program

```
//name: Naomi Thing
//student ID: 2332244
int switchinput=10;
void setup()
{
Serial.begin(9600);
pinMode(switchinput, INPUT_PULLUP);

}
void loop()
{
int switchstate=digitalRead(switchinput);
if(switchstate==LOW){
Serial.println("Name: Naomi Thing");
Serial.println("Student ID: 2332244");
delay(1000);
}
}
```

Screen Shot of Serial Port



Activity 4.2: Serial Port binary to decimal

Code

```

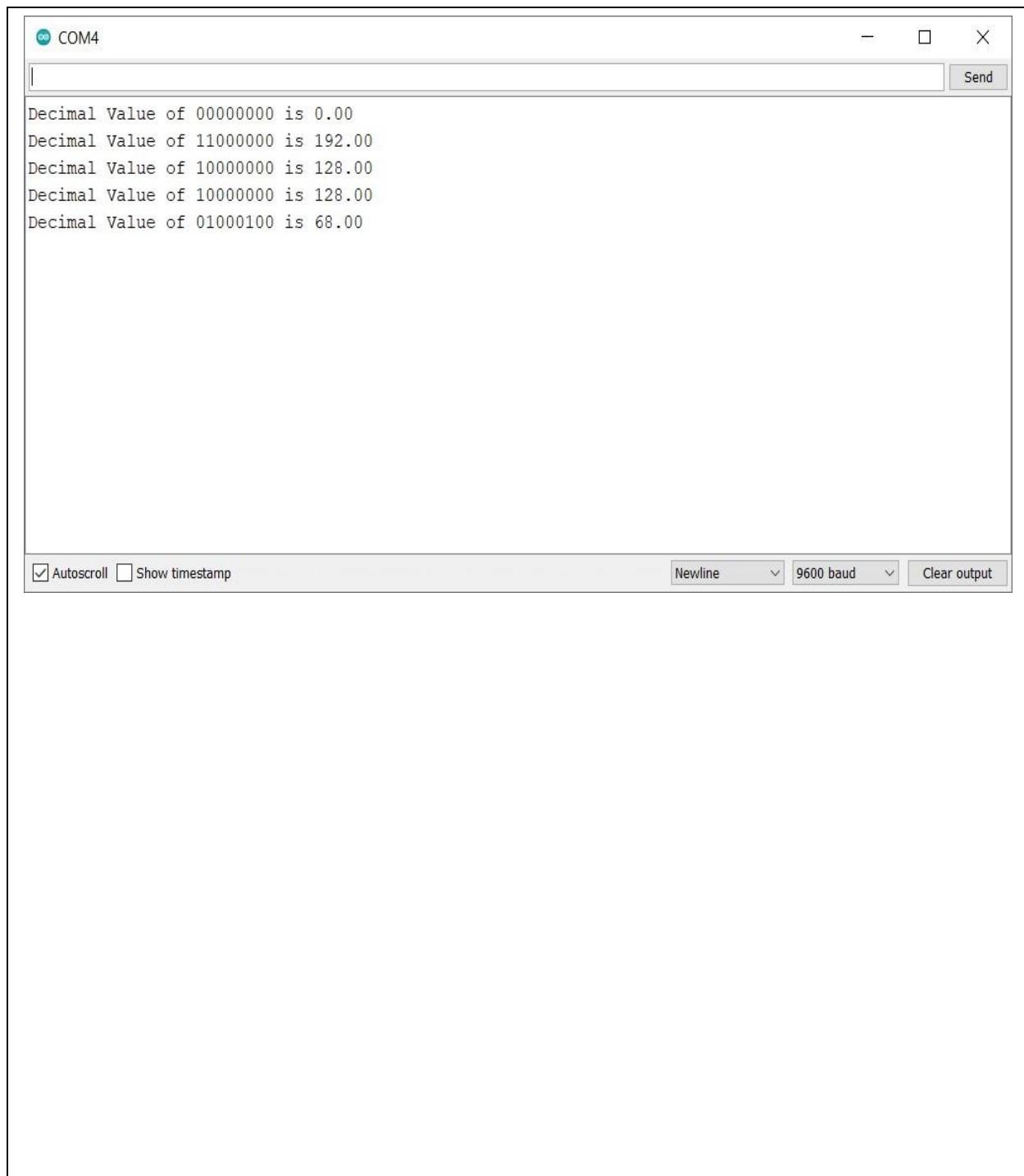
//name: Naomi Thing
//student ID: 2332244
int switchinput[8] = {6, 7, 8, 9, 10, 11, 12, 13}; //initializing 8 digital pins stored in a array named switchinput
int i; // variable 'i' being initialized
void setup()
{
    for (i = 0; i < 8; i++) {
        loop used for selecting the pinmode for each digital pin number stored in the array
        pinMode(switchinput[i], INPUT_PULLUP); //input_pullup mode used to control floating values
    }
    Serial.begin(9600); //serial begin used to set the number of bits to be transferred per second
}

void loop()
{
    int n; //variable n being declared
    int bin[10]; //array being declared of size 10
    float result = 0; //variable result being initialized
    for (i = 0; i < 8; i++) { //loop used to access the each digital pin stored in array switchinput
        int switchstate = digitalRead(switchinput[i]); //variable storing the high or low state of the digital pins
        if (switchstate == LOW) { //if the state of digital pin is low or if the button is pressed
            n = 1; //n will be assigned one
        }
        else {
            n = 0; //if button is not pressed zero will be assigned to n variable
        }
        result = result + n * pow(2, i); //calculation of decimal equivalent being added and stored in result variable
    }
    for (int j = 0; j < 9; j++) { //loop used for storing the binary number in bin array
        if (switchstate == LOW) {
            if the button is pressed
            bin[i] = 1; //the respective array index in bin array stores 1
        }
        else { //if button is not pressed
            bin[i] = 0; //the respective array index in bin array stores 0
        }
    }
}

//for showing meaningful output in serial monitor
Serial.print(" Decimal Value of ");
for (int i = 7; i >= 0; i--) { //accessing the values stored in bin array
    Serial.print(bin[i]); //displaying the values stored in bin array
}
Serial.print(" is ");
Serial.println(result); //displaying the decimal equivalent
delay(1000); //delay used to hold the output for 1000ms or 1 second
}

```

Screen Shot of Serial Port



Activity 4.3: Calibrating Analogue Information

Code

```

//name: Naomi Thing
//student ID: 2332244
int switchinput=10; //initializing variable to store pin number of digital pin 10
void setup()
{
  Serial.begin(9600); //serial begin used to set the number of bits to be transferred per second
  pinMode(switchinput, INPUT_PULLUP); //input_pullup mode used to control floating values
}

void loop()
{
  int switchstate=digitalRead(switchinput); //storing the state of digital pin in switchstate variable
  float value=analogRead(A0); //values of analog pin A0 being stored in value variable
  if(switchstate==LOW){ //if the button is pressed
    Serial.print("Value: ");
    Serial.println(value); //displaying the raw value coming from the analog pin
    Serial.print("Voltage: ");
    float v=value*(5.0/1023.0); //Calculation of voltage from the value coming from analog pin(A0)
    Serial.println(v); //displaying the equivalent voltage
    Serial.print("Resistance: ");
    float r=v*250.0/5.0; //Calculation of resistance from the calculated voltage
    Serial.println(r); //displaying the resistance
    Serial.println();
    delay(1000); //using delay to hold the message for 1000ms or 1 second
  }
}

```

Pot Resistance Clockwise

0 ohm

Pot Resistance Anti-clockwise

1019 ohm

Sample of Values

Pot Resistance against Voltage change

Pot Resistance	Voltage Measured
100.68 ohm	2.10 V
121.21 ohm	2.42 V
112.66 ohm	2.25 V
87.24 ohm	1.74 V
249.02 ohm	4.98 V

Screen Shot of Meaningful Serial Port Output, not just numbers

The screenshot shows a terminal window titled "COM4". The window displays several lines of text representing sensor readings. The text is organized into three distinct sections, each starting with "Value: " followed by a numerical value, then "Voltage: " and "Resistance: " with their respective values. The first section shows values near zero. The second section shows values around 68.00, 0.33, and 16.62. The third section shows values around 1019.00, 4.98, and 249.02. At the bottom of the window, there are several control buttons: "Autoscroll" (checked), "Show timestamp" (unchecked), "Newline" (dropdown menu), "9600 baud" (dropdown menu), and "Clear output".

```
Value: 0.00
Voltage: 0.00
Resistance: 0.00

Value: 68.00
Voltage: 0.33
Resistance: 16.62

Value: 1019.00
Voltage: 4.98
Resistance: 249.02

Value: 1019.00
Voltage: 4.98
Resistance: 249.02
```

Autoscroll Show timestamp Newline 9600 baud Clear output

Activity 4.4: Temperature Sensor & Serial Port

Code - Centigrade to Serial port, but when button Pressed Fahrenheit Displayed Instead

```
//name: Naomi Thing
//student ID: 2332244
#include <dht.h> //importing inbuilt function and methods from dht11 library
int switchinput=10; //initializing variable to store the pin number of digital pin 10
dht DHT;
#define DHT11_PIN 8
void setup(){
Serial.begin(9600); //serial begin used to set the number of bits to be transferred per second
pinMode(switchinput, INPUT_PULLUP); //input_pullup to control floating values in digital pin
}
void loop(){
int swistate=digitalRead(switchinput); //variable storing the state of digital pin i.e. HIGH or LOW
float chk = DHT.read11(DHT11_PIN);
float cel=DHT.temperature; //storing the value coming from dht11 temp sensor in cel variable
if(swistate==LOW){ //if the button is pressed
Serial.print("Temperature in Fahrenheit:");
float fah=(cel*1.8)+32; //Calculation of temperature from Celsius to Fahrenheit
Serial.println(fah); //displaying the temperature in fahrenheit
delay(1000); //holding the message for 1000ms or 1 second
}
else{ //if the button is not pressed
Serial.print("Temperature in Celcius:");
Serial.println(cel); //displaying temperature in celcius
delay(1000); //holding the message for 1000ms or 1 second
}
```

Screen Shot of Serial Port

The screenshot shows a terminal window titled "COM4". The window displays the following text in the main pane:

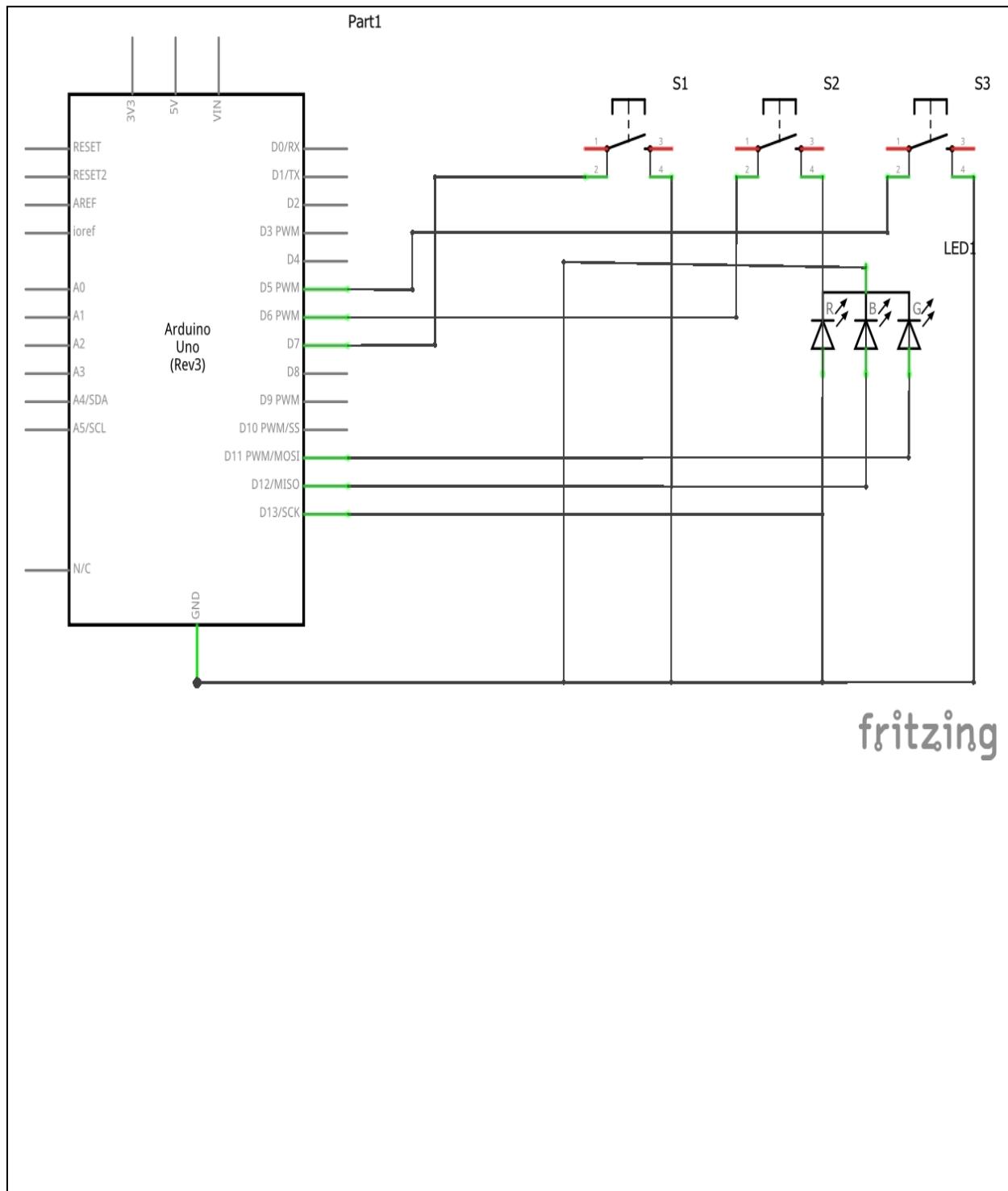
```
Temperature in Celcius:24.00
Temperature in Celcius:24.00
Temperature in Celcius:24.00
Temperature in Fahrenheit:75.20
```

At the bottom of the window, there are several control buttons and settings:

- Autoscroll Show timestamp
- Newline dropdown menu
- 9600 baud dropdown menu
- Clear output button

Workbook 5

Activity 5.1: RGB Led and switches



Fritzing
Arduino Program

```
//name: Naomi Thing
//student ID: 2332244
int redLed=13;
int blueLed=12;
int greenLed=11;
int switchInput1=7;
int switchInput2=6;
int switchInput3=5;
void setup()
{
pinMode(redLed, OUTPUT);
pinMode(blueLed,OUTPUT);
pinMode(greenLed,OUTPUT);
pinMode(switchInput1,INPUT_PULLUP);
pinMode(switchInput2,INPUT_PULLUP);
pinMode(switchInput3,INPUT_PULLUP);
}
void loop()
{
int switchState1=digitalRead(switchInput1);
int switchState2=digitalRead(switchInput2);
int switchState3=digitalRead(switchInput3);
if(switchState1==LOW) {
digitalWrite(redLed,HIGH);
digitalWrite(blueLed,HIGH);

delay(1000);
digitalWrite(redLed,LOW);
digitalWrite(blueLed,LOW);
}
else if(switchState2==LOW) {
digitalWrite(blueLed,HIGH);
digitalWrite(greenLed,HIGH);
delay(1000);
digitalWrite(blueLed,LOW);
digitalWrite(greenLed,LOW);
}
else if(switchState3==LOW) {
digitalWrite(greenLed,HIGH);

digitalWrite(redLed,HIGH);
delay(1000);
digitalWrite(greenLed,LOW);
digitalWrite(redLed,LOW);
}
}
```

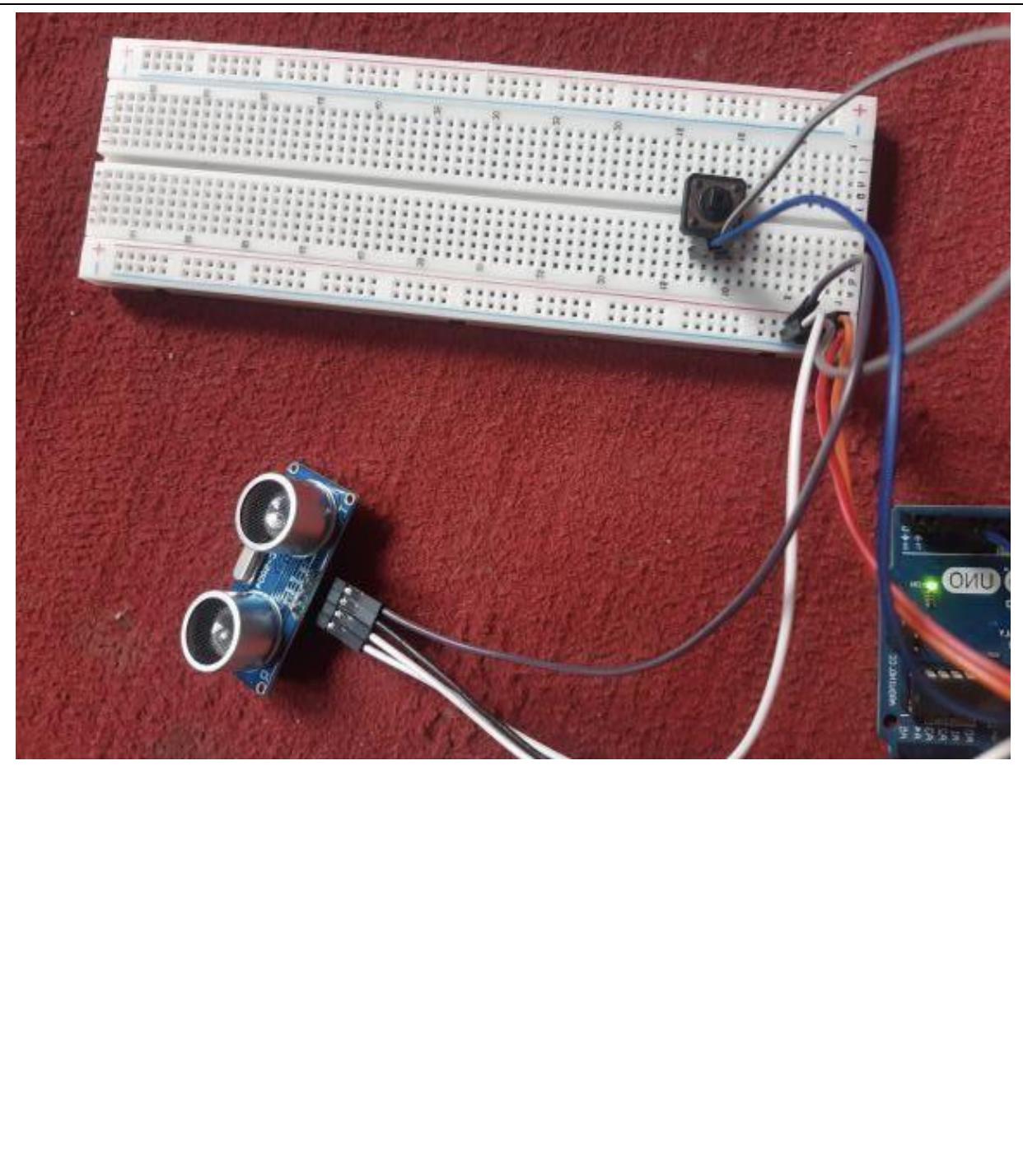
Activity 5.2: Distance Sensor

Arduino Code

```
//name: Naomi Thing
//student ID: 2332244
#define trigPin 9
#define echoPin 8
int switchinput=2;
void setup()
{
    serial.begin(9600);
    pinMode(echoPin, INPUT_PULLUP);
    pinMode(switchinput, INPUT_PULLUP);
}
void loop()
{
    int switchstate=digitalRead(switchinput);
    long duration,inches,cm;
    pinMode(trigPin, OUTPUT);
    digitalWrite(trigPin,LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin,LOW);
    duration=pulseIn(echoPin,HIGH);

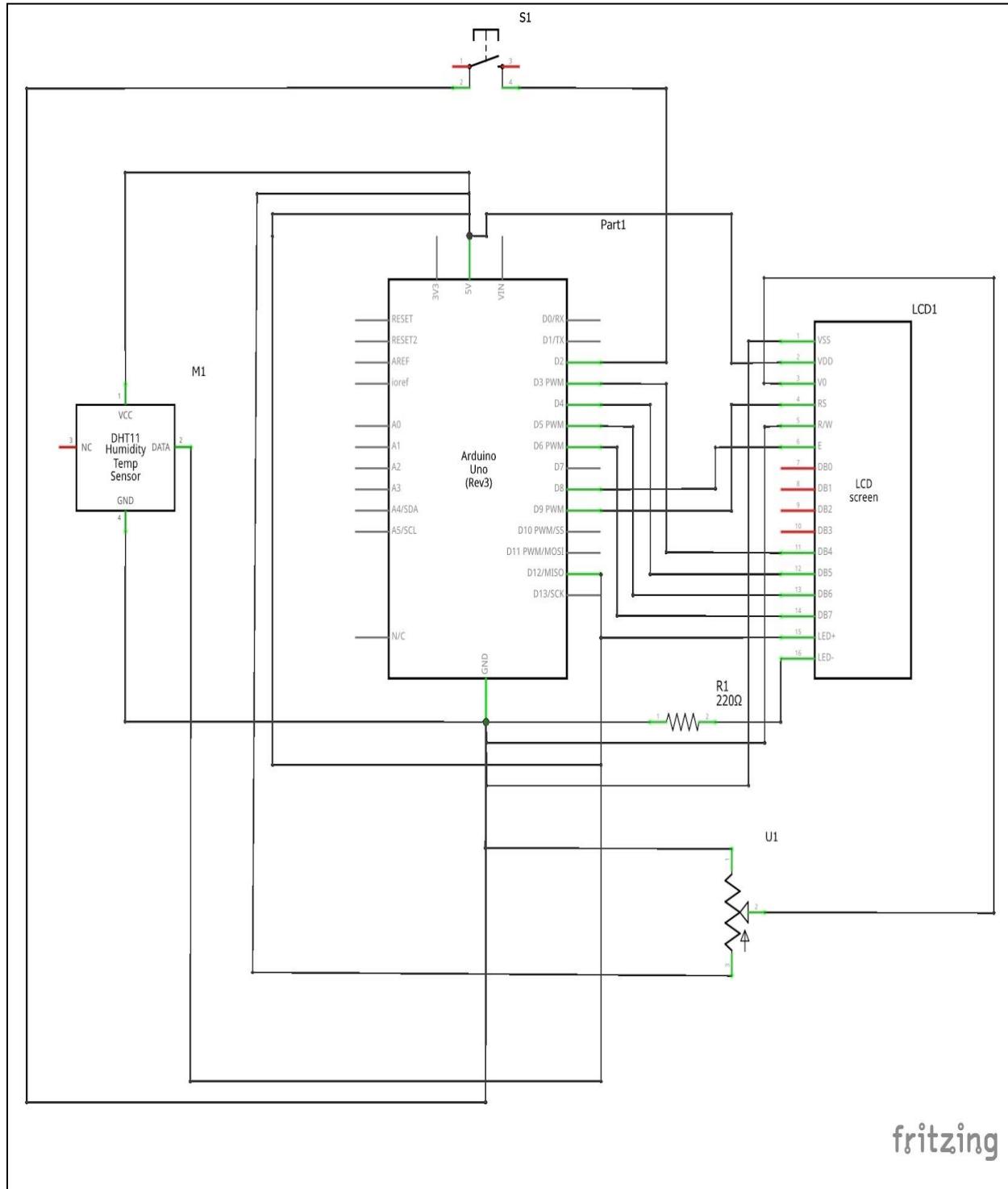
    inches=duration/74/2;
    if(switchstate==LOW){
        serial.print("Distance Measured in Inches: ");
        serial.print(inches);
        serial.println("in");
        serial.print("Distance Measured in Centimeters: ");
        serial.print(cm);
        serial.println("cm");
        delay(1000);
    }
}
```

Take a picture of your distance sensor and include it here, please reduce the size and quality as it will be



too large else □

Activity 5.3: 1602 LCD Display



Fritzing

Arduino Program

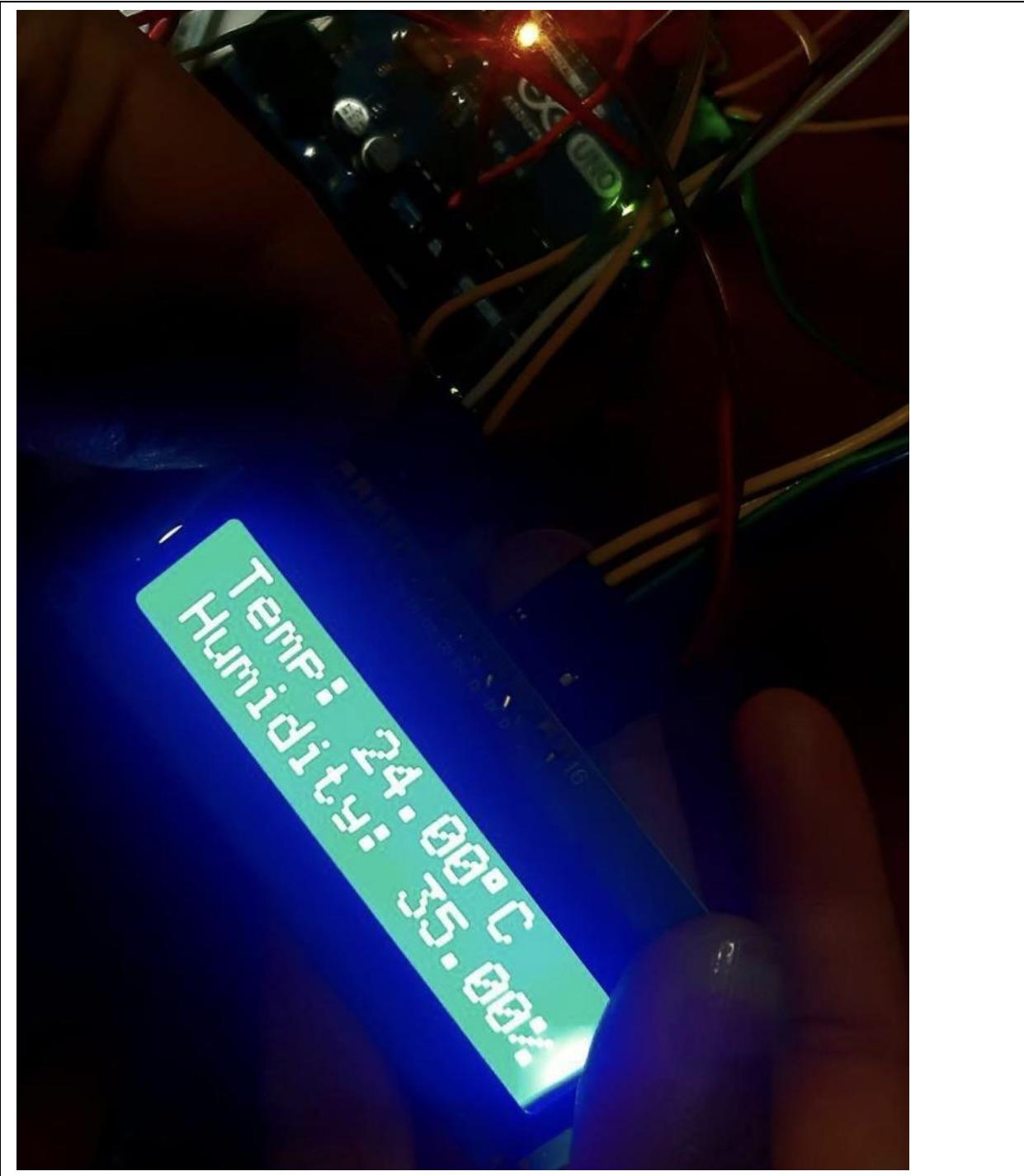
```
//name: Naomi Thing
//student ID: 2332244
#include <LiquidCrystal.h>
#include <dht.h>

int Contrast=75;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
dht DHT;
#define DHT11_PIN 7

void setup()
{
analogWrite(6,Contrast);
lcd.begin(16, 2);
}
void loop()
{
int chk = DHT.read11(DHT11_PIN);
lcd.setCursor(0,0);
lcd.print("Temp: ");
lcd.print(DHT.temperature);
lcd.print((char)223);
lcd.print("C");
lcd.setCursor(0,1);
lcd.print("Humidity: ");

lcd.print(DHT.humidity);
lcd.print("%");
delay(1000);
}
```

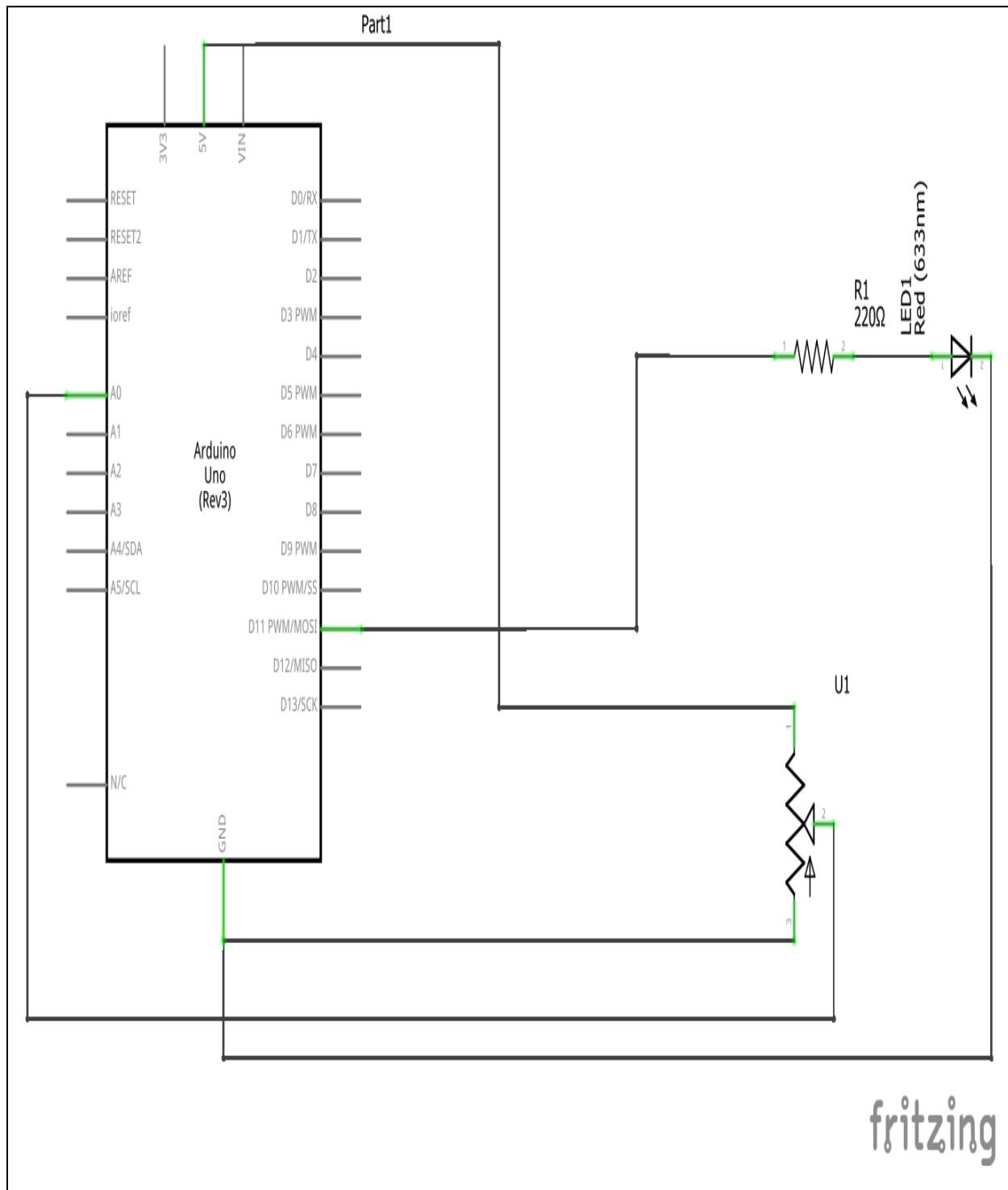
Take a picture of your LCD and include it here, please reduce the size and quality as it will be too large



else □

Workbook 6

Activity 6.1: PWM



Fritzing
Arduino Program

```
//name: Naomi Thing
//student ID: 2332244
int redled=11;

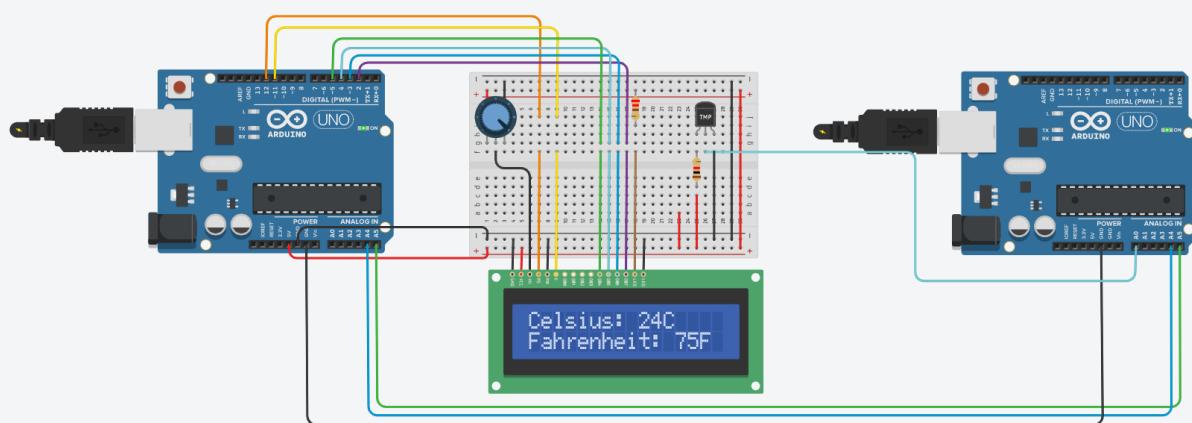
void setup()
{
Serial.begin(9600); //serial begin used to start serial monitor
pinMode(redled, OUTPUT); //pinmode selected as output for led
}

void loop()
{
int analogval=analogRead(A0); //analog value in analog pin 0 stored in variable
float voltage=analogval*5.0/1023.0; //calculation of voltage from the value coming from analog pin
1, controlled by pot
float val=voltage*255.0/5.0; //calculation of value from 0-255 as the scale of analogWrite is 0-255
analogWrite(redled,val); //writing analog value whose range is from 0-255
Serial.println(val); //displaying the value in serial monitor
delay(1000); //holding the message for 1000ms or 1 second
}
```

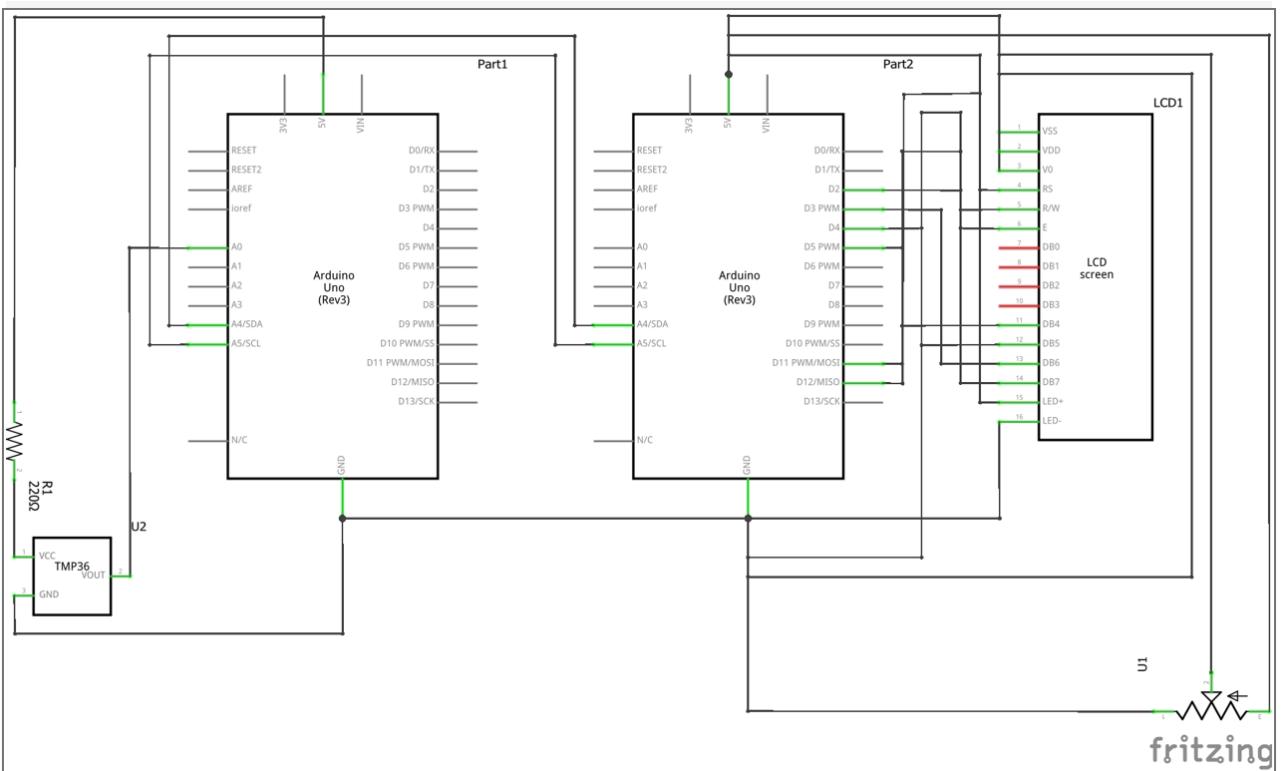
Workbook 7

Activity 7.1: 2 Arduinos – using Digital Pins

circuit diagram



Fritzing



Fritzing

```

/*
//Name: Naomi Thing
// Student I.D: 2332244
#include <Wire.h> //importing the inbuilt functions and methods from Wire.h library
int tempPin = A0; //initializing A0 in the variable tempPin
void setup()
{
    delay(500); //delay set for 500ms or 0.5 second
    Wire.begin(); //begin I2C or TWI
    Serial.begin(9600); // serial begin used to start serial monitor
}
void loop()
{
    float adcVal= analogRead(tempPin); //analog value coming from A0 pin stored in adrVal variable
    float Vout= (adcVal*5)/1023.0; //voltage coming from A0 pin calculated in Vout variable
    int celsius = (Vout - 0.5) * 100; //Temperature in Celsius calculated and stored in 'celcius' variable
    int fahrenheit = celsius * (9.0 / 5.0) + 32.0; // Temperature in Celsius calculated
    float centigrade = (Vout - 0.5) * 100;
    float fahrenheit1 = celsius * (9.0 / 5.0) + 32.0;

    //transmit to device with address 8
    Wire.beginTransmission(8); //establishing connection with the Arduino connected digital pin 8
    Wire.write(celsius); //sending Celsius to the other Arduino
    Wire.write(fahrenheit); //sending Fahrenheit to the other Arduino
    Wire.endTransmission(); //stopping the transmitting connection with the other Arduino
    delay(1000);
}

#define Name Naomi Thing
#define student id 2332244

#include <LiquidCrystal.h> // importing the inbuilt functions and methods from liquidcrystal library
#include <Wire.h> //importing the inbuilt functions and methods from Wire.h library

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //used for creating an object and defining the connection pins

void setup()
{
    lcd.begin(20, 4); //used for initializing the number of rows and columns on the lcd
    Serial.begin(9600); // serial begin used to start serial monitor
    Wire.begin(8); //begin I2C or TWI having address 8
    /*act as slave component(dataReceiver) to display temp reading
    when master component(dataSender) want to write temp data to it*/
    Wire.onReceive(dataReceived); //redirected to the user-defined function after receiving data
    lcd.print("Waiting for data..."); //displaying 'waiting' until data is fetched or received
    lcd.setCursor(0,1); //setting the position of the cursor to column 0 and row 1
    lcd.print("Please wait..."); // displaying 'waiting' until data is fetched or received
    pinMode(8, OUTPUT); //setting pinmode of digital pin 8 to output
}

void loop()
{
}

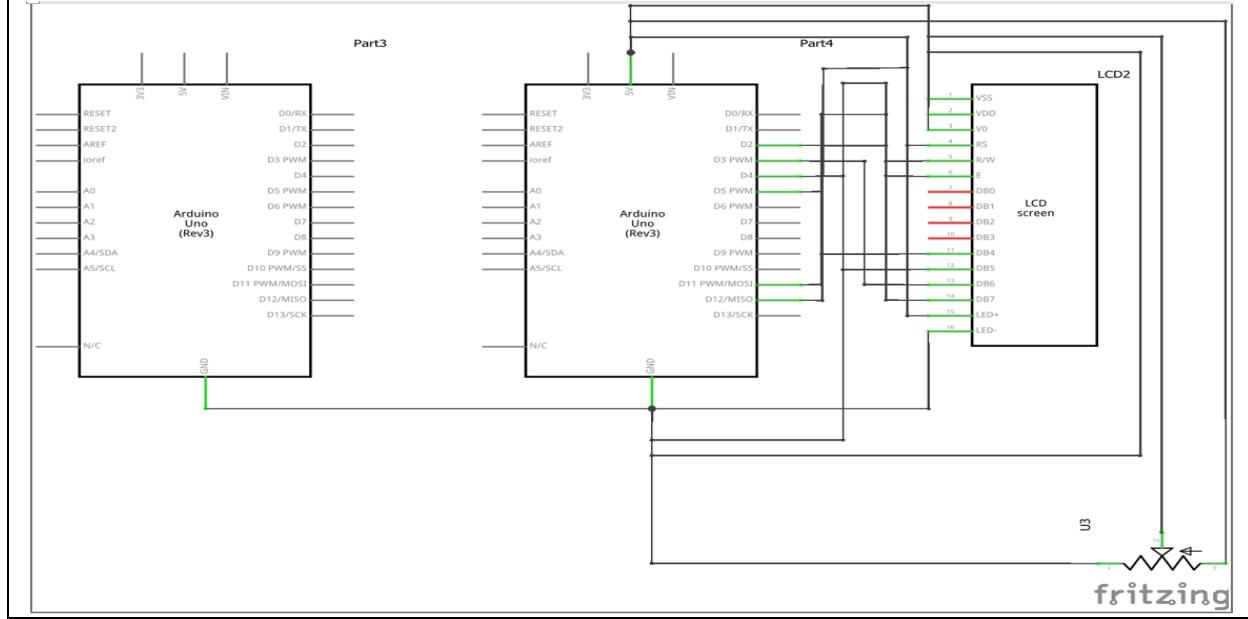
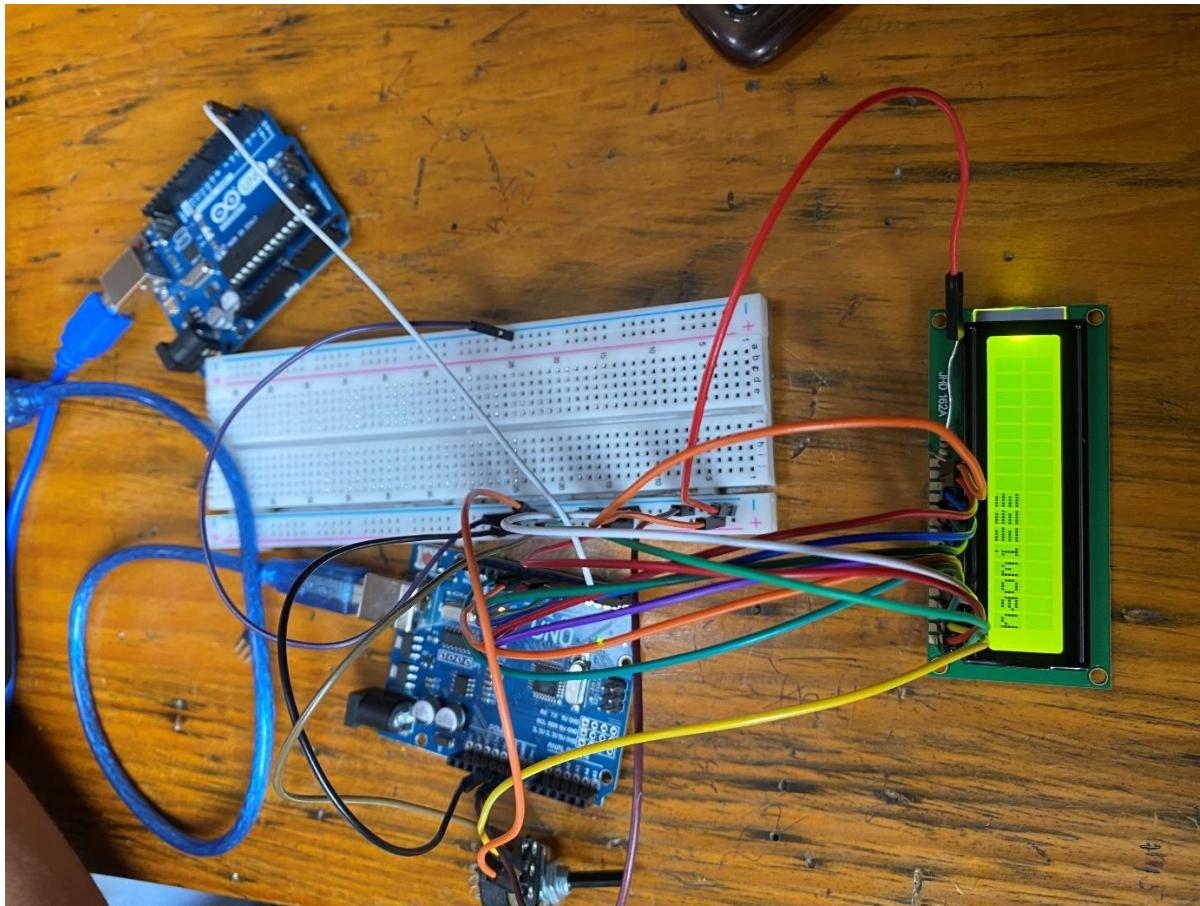
```

```
void dataReceived(int x)//x shows the number of times data is received
{
    int CelsiusVal = Wire.read(); //fetching the data (temp in Celsius) sent from the other Arduino
    Serial.print("Temperature in celsius: ");
    Serial.print(CelsiusVal); //displaying the temperature in Celsius
    Serial.print("degC");
    Serial.println();
    int FahrenheitVal = Wire.read(); // fetching the data (temp in Celsius) sent from the other Arduino
    Serial.print("Temperature in fahrenheit: ");
    Serial.print(FahrenheitVal); //displaying the temperature in Fahrenheit
    Serial.print("degF");
    Serial.println();
    lcd.clear(); //clearing messages on the lcd display

    lcd.setCursor(0, 0); //setting the position of the cursor to column 0 and row 0
    lcd.print("Celsius: ");
    lcd.print(CelsiusVal);//displaying temperature in Celsius
    lcd.print("C");

    lcd.setCursor(0, 1);
    lcd.print("Fahrenheit: ");
    lcd.print(FahrenheitVal);//displaying temperature in Fahrenheit
    lcd.print("F");
}
```

Activity 7.2: 2 Arduinos – using Serial I/O



Fritzing

1st Arduino Program

```
#name= Naomi Thing
#student id= 2332244

char a[10]; //initializing variable a as a string
void setup(){
    Serial.begin(9600); used for starting the serial monitor
}
void loop(){
    while(Serial.available()>0) //when there is any data to read from the serial port
    {
        int i=0; //initializing variable 'i' => 0
        char c=Serial.read(); //reads a single character from the serial port buffer
        if(c!='\n') //if the value stored in c is not newline
        {
            a[i]=c; //variable c will be stored in the 'i' index position of a
            i++; //incrementing 'i' by 1
        }
        Serial.write(a,1); //Used to write the data into the serial port
    }
}
```

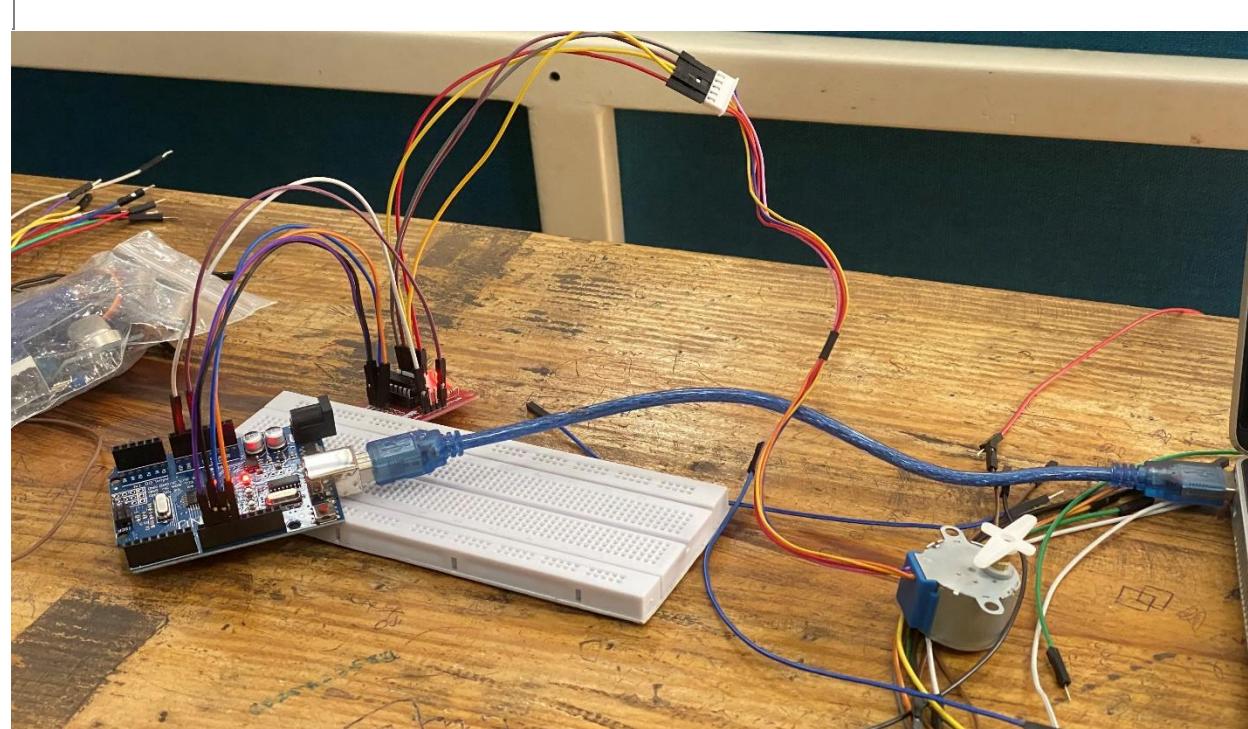
2nd Arduino Connected to Lcd Display program.

```
#name= Naomi Thing
#student id= 2332244

#include <LiquidCrystal.h> // importing the inbuilt functions and methods from liquidcrystal library
char mystr1[10]; //initializing variable mystr1 as string
LiquidCrystal lcd(11,10,4,5,6,7); // used for creating an object and defining the connection pins
void setup() {
    lcd.begin(16, 2); //used for initializing the number of rows and columns on the lcd
    Serial.begin(9600); //used for starting the serial port/monitor
}
void loop() {
    Serial.readBytes(mystr1,10); //reads the data from serial port to serial buffer
    lcd.println(mystr1); //displaying the data entered by the user in the serial port
    delay(1000); //delaying for 1000ms or 1 second
    lcd.clear(); //clearing the data displayed in the lcd display
}
```

Workbook 8

Activity 8.1: Stepper Motor Circuit Diagram



Arduino Program

```
#name= Naomi Thing
#student id= 2332244

#define STEPPER_PIN_1 9 //defining different variables different pin numbers from 9-12
#define STEPPER_PIN_2 10
#define STEPPER_PIN_3 11
#define STEPPER_PIN_4 12
int step_number = 0; //initializing step_number variable 0
void setup() {
pinMode(STEPPER_PIN_1, OUTPUT); //pinmode selected to all the stepper pins as OUTPUT
pinMode(STEPPER_PIN_2, OUTPUT);
pinMode(STEPPER_PIN_3, OUTPUT);
pinMode(STEPPER_PIN_4, OUTPUT);

}

void loop() {
    for (int pos=0;pos<=360;pos+=1){ //used to delay OneStep(true) for 5400ms or 5.4 seconds
        OneStep(true); //user defined function OneStep is send true as argument
        delay(15); //delaying for 15 m/s
    }
    for (int pos=360;pos>=0;pos-=1){ //used to delay OneStep(false) for 5400ms or 5.4 seconds
        OneStep(false); // user defined function OneStep is send true as argument
        delay(15); //delaying for 15 m/s
    }
}

void OneStep(bool dir){ //user-defined function OneStep(bool)
    if(dir){ //if True is sent as argument in OneStep function
switch(step_number){ //switch case used and step_number sent as case
    case 0: //first case 0 is executed as at first step_number is 0
        digitalWrite(STEPPER_PIN_1, HIGH); //digital writing the stepper pins HIGH or LOW
        digitalWrite(STEPPER_PIN_2, LOW);
        digitalWrite(STEPPER_PIN_3, LOW);
        digitalWrite(STEPPER_PIN_4, LOW);
        break;
    case 1: //case 1 is executed when step_number is 1
        digitalWrite(STEPPER_PIN_1, LOW); //digital writing the stepper pins HIGH or LOW
        digitalWrite(STEPPER_PIN_2, HIGH);
        digitalWrite(STEPPER_PIN_3, LOW);
        digitalWrite(STEPPER_PIN_4, LOW);
        break;
    case 2: //case 2 is executed when step_number is 2
        digitalWrite(STEPPER_PIN_1, LOW); //digital writing the stepper pins HIGH or LOW
        digitalWrite(STEPPER_PIN_2, LOW);
        digitalWrite(STEPPER_PIN_3, HIGH);
        digitalWrite(STEPPER_PIN_4, LOW);
        break;
    case 3: //case 3 is executed when step_number is 3
        digitalWrite(STEPPER_PIN_1, LOW); //digital writing the stepper pins HIGH or LOW
        digitalWrite(STEPPER_PIN_2, LOW);
        digitalWrite(STEPPER_PIN_3, LOW);
        break;
}
}
}
```

```
    digitalWrite(STEPPER_PIN_4, HIGH);
    break;
}
}else{ //if false is sent as argument in OneStep function
    switch(step_number){ //switch case used and step_number sent as case
case 0: //first case 0 is executed as at first step_number is 0
    digitalWrite(STEPPER_PIN_1, LOW);
    digitalWrite(STEPPER_PIN_2, LOW);
    digitalWrite(STEPPER_PIN_3, LOW);
    digitalWrite(STEPPER_PIN_4, HIGH);
    break;
case 1: //case 1 is executed when step_number is 1
    digitalWrite(STEPPER_PIN_1, LOW);
    digitalWrite(STEPPER_PIN_2, LOW);
    digitalWrite(STEPPER_PIN_3, HIGH);
    digitalWrite(STEPPER_PIN_4, LOW);
    break;
case 2: //case 2 is executed when step_number is 2
    digitalWrite(STEPPER_PIN_1, LOW);
    digitalWrite(STEPPER_PIN_2, HIGH);
    digitalWrite(STEPPER_PIN_3, LOW);
    digitalWrite(STEPPER_PIN_4, LOW);
    break;
case 3: //case 3 is executed when step_number is 3
    digitalWrite(STEPPER_PIN_1, HIGH);
    digitalWrite(STEPPER_PIN_2, LOW);
    digitalWrite(STEPPER_PIN_3, LOW);
    digitalWrite(STEPPER_PIN_4, LOW);
}
}
step_number++; //incrementing step_number by 1 after each execution
if(step_number > 3){ //if step_number is less than 3
    step_number = 0; //when step_number is 3 or greater than 3 it will be set to 0
}
}
```

Activity 8.2: 2 Stepper Motors

```
#name= Naomi Thing
#student id= 2332244

void setup()
{
    pinMode(8, OUTPUT); //Selecting the pinmodes for digital pins 8-11 as OUTPUT
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(6, INPUT); //Selecting the pinmodes for digital pins 3-6 as INPUT
    pinMode(5, INPUT);
    pinMode(4, INPUT);
    pinMode(3, INPUT);
}
void loop()
{
    int switchInput1=digitalRead(6); //storing the data read from pin 6 in switchinput1
    int switchInput2=digitalRead(5); //storing the data read from pin 5 in switchinput2
    int switchInput3=digitalRead(4); //storing the data read from pin 4 in switchinput3
    int switchInput4=digitalRead(3); //storing the data read from pin 3 in switchinput4
    if(switchInput1==HIGH) if switchInput1 is HIGH or pin 6 is HIGH or pressed
    {
        digitalWrite(8, LOW); //Setting digitalpin 8 LOW
        digitalWrite(9, HIGH); //Setting digitalpin 9 HIGH
        delay(10); //delaying for 10ms
        digitalWrite(10, LOW); //Setting digitalpin 10 LOW
```

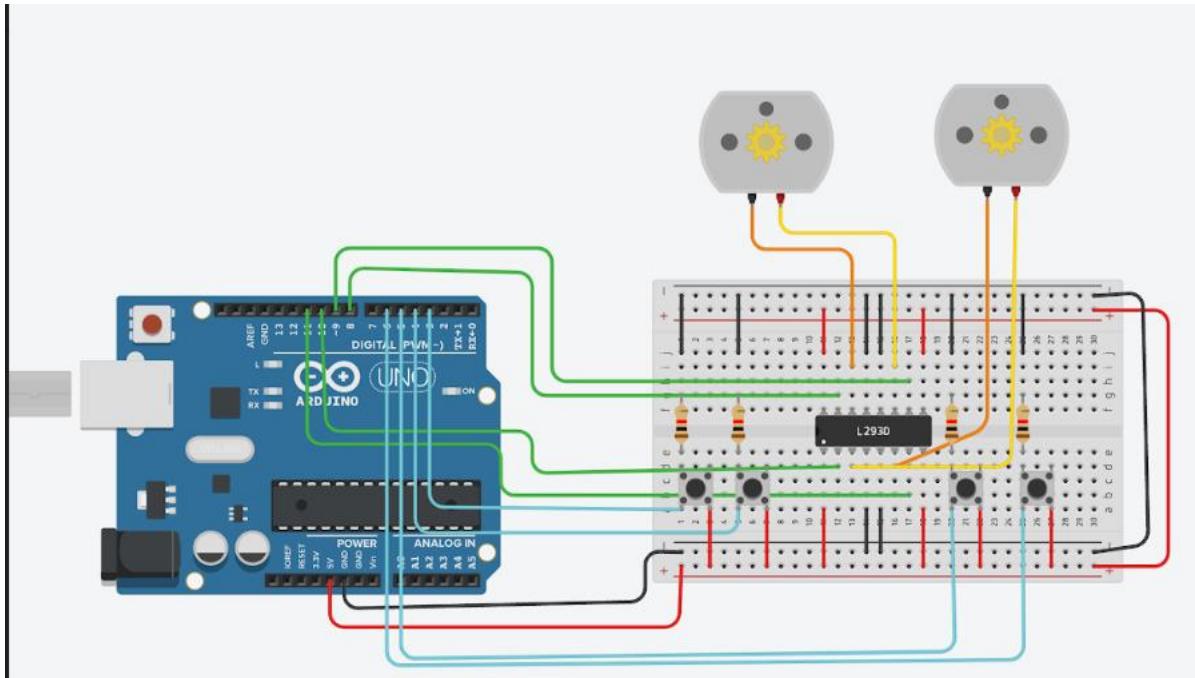
```

digitalWrite(11, HIGH); //Setting digitalpin 11 HIGH
delay(10); //delaying for 10ms
}
if(switchInput2==HIGH)//if switchinput2 is HIGH or pressed
{
    digitalWrite(8, HIGH); //Setting digitalpin 8 HIGH
    digitalWrite(9, LOW); //Setting digitalpin 9 LOW
    delay(10);
    digitalWrite(10, HIGH); //Setting digitalpin 10 HIGH
    digitalWrite(11, LOW); //Setting digitalpin 11 LOW
    delay(10);
}
if(switchInput3==HIGH) //if switchinput3 is HIGH or pressed on
{
    digitalWrite(8, HIGH); //Setting digitalpin 8 HIGH
    digitalWrite(9, LOW); //Setting digitalpin 9 LOW
    delay(10);
    digitalWrite(10, LOW); //Setting digitalpin 10 LOW
    digitalWrite(11, HIGH); //Setting digitalpin 11 HIGH
    delay(10);
}
if(switchInput4==HIGH) if switchinput4 is HIGH or pressed on
{
    digitalWrite(8, LOW); //Setting digitalpin 8 LOW
    digitalWrite(9, HIGH); //Setting digitalpin 9 HIGH
    delay(10);
}
if(switchInput4==HIGH) if switchinput4 is HIGH or pressed on
{
    digitalWrite(8, LOW); //Setting digitalpin 8 LOW
    digitalWrite(9, HIGH); //Setting digitalpin 9 HIGH
    delay(10);
    digitalWrite(10, HIGH); //Setting digitalpin 10HIGH
    digitalWrite(11, LOW); //Setting digitalpin 11 LOW
    delay(10);
}
digitalWrite(8, LOW); //Setting digitalpin 8 LOW
digitalWrite(9, LOW); //Setting digitalpin 9 LOW
delay(10);
digitalWrite(10, LOW); //Setting digitalpin 10 LOW
digitalWrite(11, LOW); //Setting digitalpin 11 LOW
delay(10);
}

```

Arduino Program

Tinkercad Figure



Workbook 9

Activity 9.1: Windscreen Wiper Code using Servos & Temperature Sensor

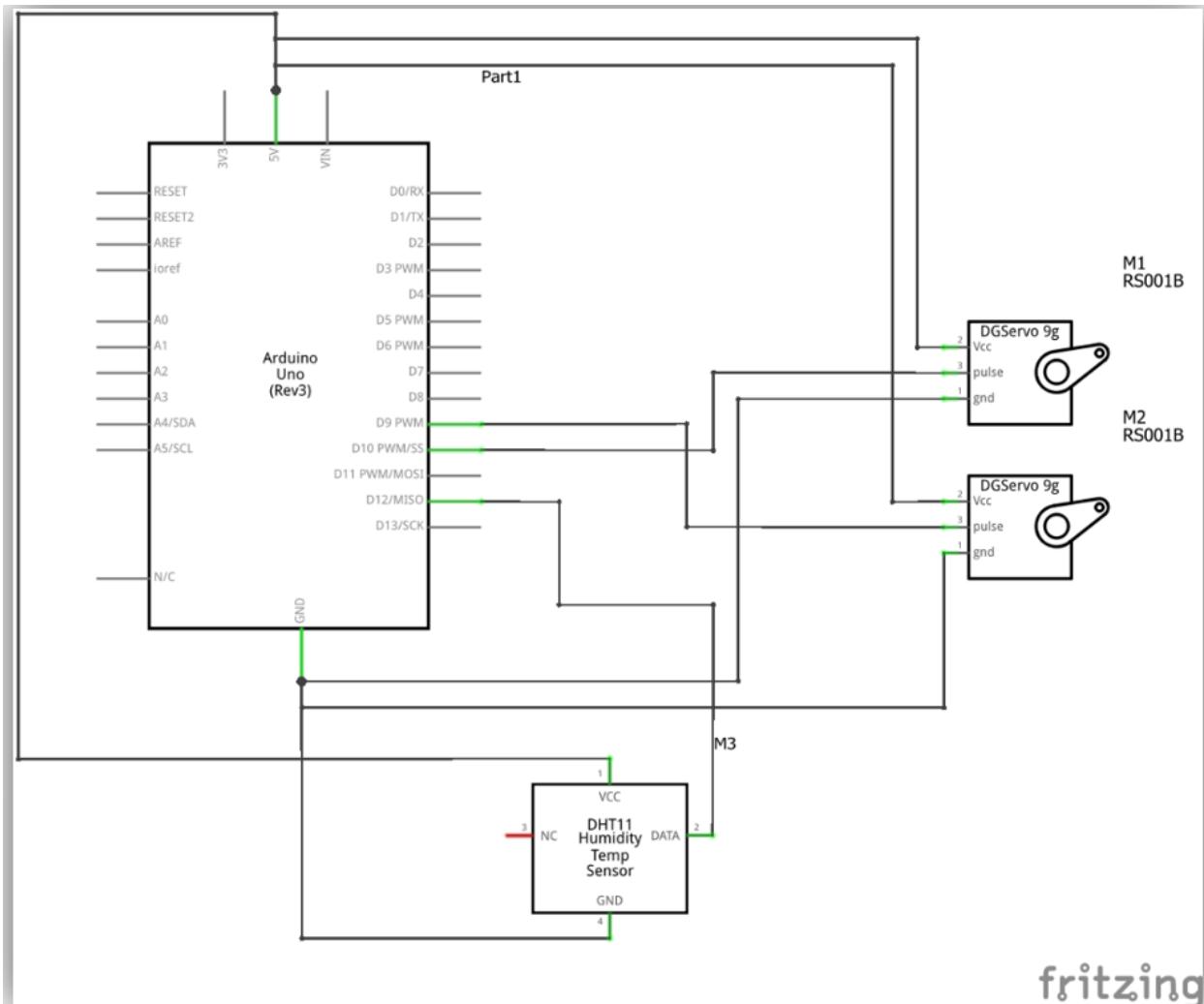
Arduino Code:

```
#name= Naomi Thing
#student id= 2332244

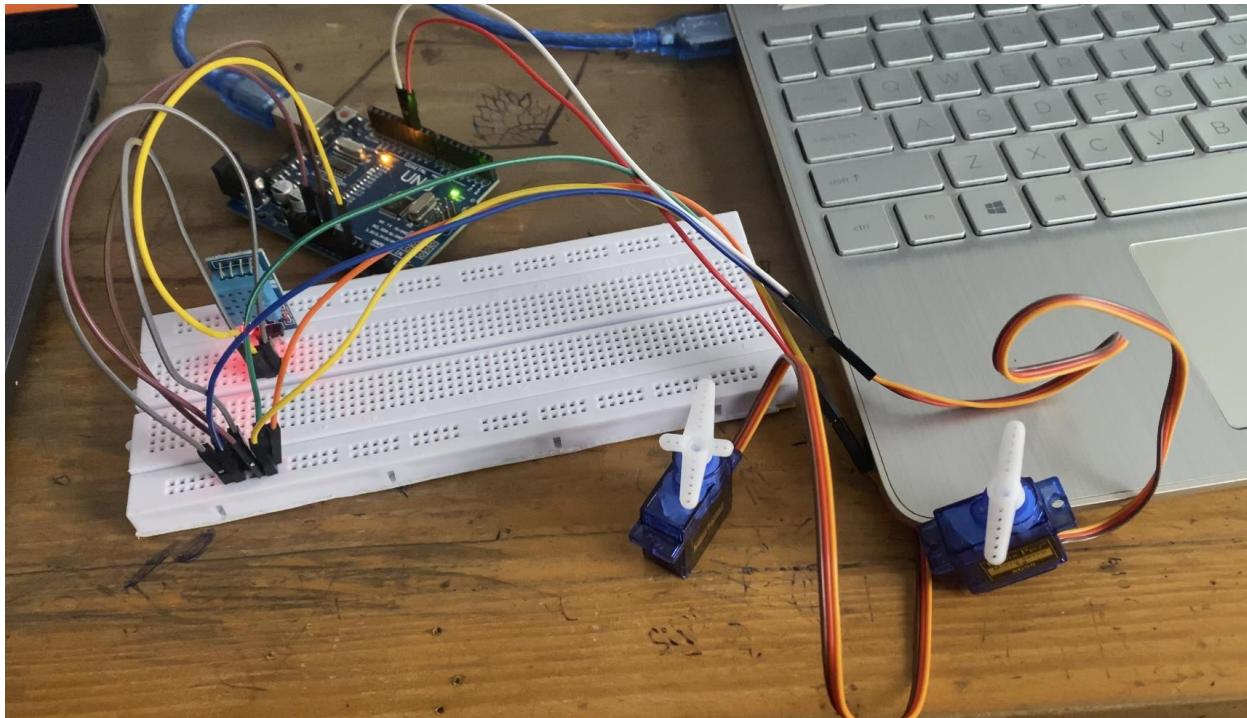
#include <Servo.h> //importing the inbuilt functions and methods from the library Servo.h
#include <dht.h> //importing the inbuilt functions and methods from dht.h library
dht DHT;
#define DHT11_PIN 12 //defining DHT_11PIN the 12 number pin
Servo myservo; //initializing Servo as variable myservo
Servo myservo1; //initializing another Servo as myservo2
int pos=0; //initializing pos variable as 0
void setup()
{
    myservo.attach(9); //establishing connection between servo and digital pin 9
    myservo1.attach(10); //establishing connection between servo and digital pin 10
    Serial.begin(9600); //starting the serial monitor
}

void loop()
{
    float chk = DHT.read11(DHT11_PIN);
    float cel=DHT.temperature; //using the object DHT.temperature to access the temperature readings
    Serial.print("Temperature: ");
    Serial.println(cel); //displaying the temperature in Celsius
    if(cel>20){ //if the temperature is greater than 20
//the two servos move like a wiper of a car
        for(pos=0;pos<=90;pos+=1){
            myservo.write(pos); //writing the value of pos in myservo
            myservo1.write(pos); //writing the value of pos in myservo1
            delay(15);
        }
        for(pos=90;pos>=0;pos-=1){
            myservo.write(pos); //writing the value of pos in myservo
            myservo1.write(pos); //writing the value of pos in myservo1
            delay(15);
        }
    }
}
```

Fritzing figure



Workshop



Individual Project (50%)

Rationale

Throughout the module you have used a range of sensors and actuators with an Arduino to complete weekly tasks. For the mini project we would like you to research and create a small embedded project in an area of your choice, such as:

- Games
- Networking
- IT Security
- Systems Engineering • Smart Technology
- Artificial Intelligence

Previous projects have included a reaction game that gives a score depending on how fast you hit a button, this has buttons to restart the application, and an LCD to show scores, and information.

This project should be your own work, **YOU MUST NOT COPY A PROJECT FROM THE INTERNET.**

Timescales

This project should be started around week 5 and continue until the deadline, when it will be submitted in the Portfolio.

Equipment

You are free to use Tinkercad, or your own kit.

The Project

Step 1 produce a detailed description of your project.

This should clearly describe what you are intending to build and may contain some diagrams of how the sensor/switches input is to be processed by the Arduino. Then what kind of output is intended to be seen or heard by the user. Please mention any tools you intend to use.

Step 2 Circuit Diagram&Fritzing Schematic

You are required to produce a circuit diagram of your work showing any calculations you made, so these might be suitable resistor values for any LED's you use. These calculations are covered on the module. The circuit diagram should not be hand drawn but should follow the format of circuits from the module.

Step 3 A Program

You will need to write some software for this project and a listing of the code with suitable comments will need to be included.

Step 4 Testing

You will be required to produce some suitable test data that you would expect to be able to measure such as voltages, test code.

Once your prototype is complete you will be expected to test your circuit and compare the actual values to your initial test data, and comment on the results.

Step 5 Conclusions

You are required to write a summary of the work along with a short half page reflection on how you found the work.

Layout

The report should be suitably laid out for a report, using headings, references if required in Harvard style, and appendices used for any lengthy code. All diagrams should be produced on a PC, and handwritten work is not acceptable.

Marking

All sections carry equal marks.