|  |  |  |
| --- | --- | --- |
| **Student Name** |  | **Student Number** |
| Alan Shah |  | 2329767 |

**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](#_Toc105489596)

[Activity 1.1: Actual voltage across 5V breadboard pins. 4](#_Toc105489597)

[Activity 1.2: Actual voltage across 3.3V breadboard pins. 4](#_Toc105489598)

[Activity 1.3: Potential Divider Calculations 5](#_Toc105489599)

[Activity 1.4: 3V Calculations from either the 5V supply or 3.3V supply 6](#_Toc105489600)

[Activity 1.5: Voltage Divider circuit readings from Breadboard circuit. 7](#_Toc105489601)

[Activity 1.6: LED Circuits 7](#_Toc105489602)

[Activity 1.7: Current Measurement 8](#_Toc105489603)

[Activity 1.8: Fritzing for 4 switches & LEDS 10](#_Toc105489604)

[Activity 1.9: Fritzing for Number 0-7 11](#_Toc105489605)

[Workbook 2 12](#_Toc105489606)

[Activity 2.1: LED Flashing to show decimal number 63 as binary. 12](#_Toc105489607)

[Activity 2.2: 4 LED’s for counting up in binary from 0 to 15. 13](#_Toc105489608)

[Activity 2.3: Traffic Lights 20](#_Toc105489609)

[Workbook 3 22](#_Toc105489610)

[Activity 3.1: Circuit Diagram of Button & LED 22](#_Toc105489611)

[Activity 3.2: 3 Switches & Led 24](#_Toc105489612)

[Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS) 26](#_Toc105489613)

[Workbook 4 28](#_Toc105489614)

[Activity 4.1: Serial Port 28](#_Toc105489615)

[Activity 4.2: Serial Port binary to decimal 31](#_Toc105489616)

[Activity 4.3: Calibrating Analogue Information 33](#_Toc105489617)

[Activity 4.4: Temperature Sensor & Serial Port 36](#_Toc105489618)

[Workbook 5 38](#_Toc105489619)

[Activity 5.1: RGB Led and switches 38](#_Toc105489620)

[Activity 5.2: Distance Sensor 40](#_Toc105489621)

[Activity 5.3: 1602 LCD Display 42](#_Toc105489622)

[Workbook 6 46](#_Toc105489623)

[Activity 6.1: PWM 46](#_Toc105489624)

[Workbook 7 48](#_Toc105489625)

[Activity 7.1: 2 Arduinos – using Digital Pins 48](#_Toc105489626)

[Activity 7.2: 2 Arduinos – using Serial I/O 50](#_Toc105489627)

[Workbook 8 53](#_Toc105489628)

[Activity 8.1: Stepper Motor Circuit Diagram 53](#_Toc105489629)

[Activity 8.2: 2 Stepper Motors 55](#_Toc105489630)

[Workbook 9 56](#_Toc105489631)

[Activity 9.1: Windscreen Wiper Code using Servos & Temperature Sensor 56](#_Toc105489632)

[Individual Project (50%) 58](#_Toc105489633)

[Rationale 58](#_Toc105489634)

[Timescales 58](#_Toc105489635)

[Equipment 58](#_Toc105489636)

[The Project 58](#_Toc105489637)

[Step 1 produce adetailed description of your project. 58](#_Toc105489638)

[Step 2 Circuit Diagram&Fritzing Schematic 58](#_Toc105489639)

[Step 3 A Program 59](#_Toc105489640)

[Step 4 Testing 59](#_Toc105489641)

[Step 5 Conclusions 59](#_Toc105489642)

[Layout 59](#_Toc105489643)

[Marking 59](#_Toc105489644)

[All sections carry equal marks. 59](#_Toc105489645)

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.

5.08V

## Activity 1.2: Actual voltage across 3.3V breadboard pins.

3.30V

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.

In tinkercad while reading the value with the help of multimeter we get the exact value but while reading the value in real life we do not get the exact value as given in tinkercad. We are provided with two various values cause in simulator everything is done virtually via a computer . while reading the value in a simulator the environmental disturbance is going to be absent but while reading the value in real time there is going to be the presence of environmental disturbance like humidity and temperature. During the reading of multimeter we couldn’t obtain the exact figure since we had to calibrate it to produce exact value. Since temperature helps to charge loss, we couldn’t get the exact number.

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

V' = 4.84V

V = 5V

T= (V - V')/ V \* 100

= (5 - 4.84)/ 5 \* 100

= ± 3.2%

Then,

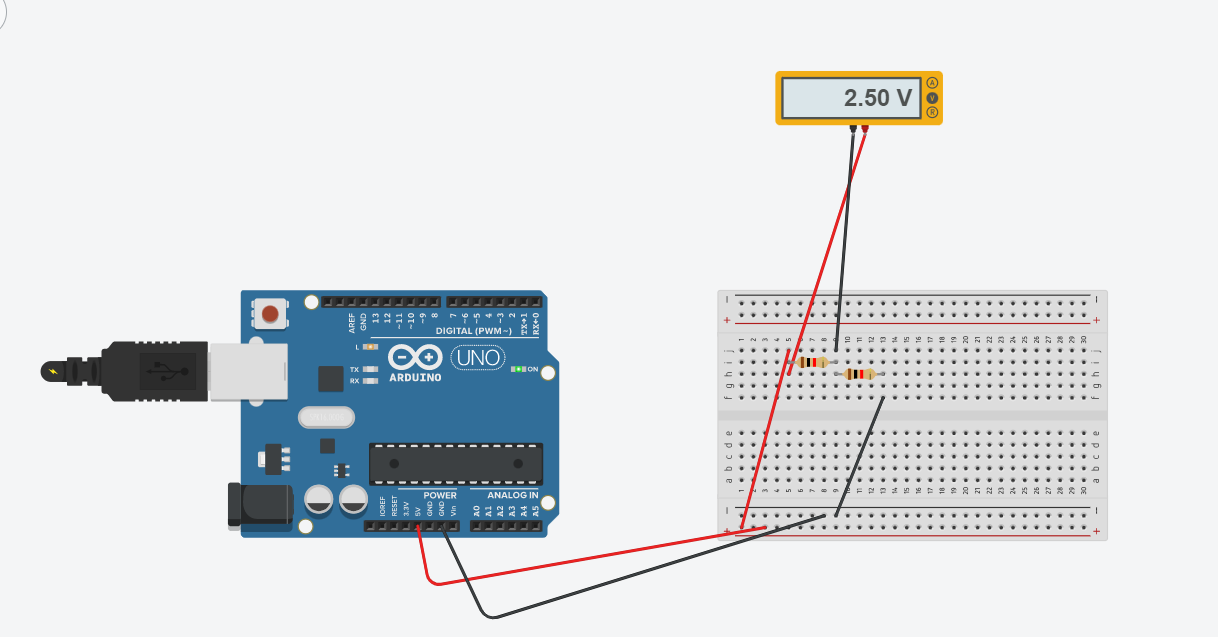
V''= (V \* 103.2)/ 100

= (5 \* 103.2)/ 100

= 5.16V

## Activity 1.3: Potential Divider Calculations

Show the working on how you achieved 2.5V



V=(R/R Total)\*V

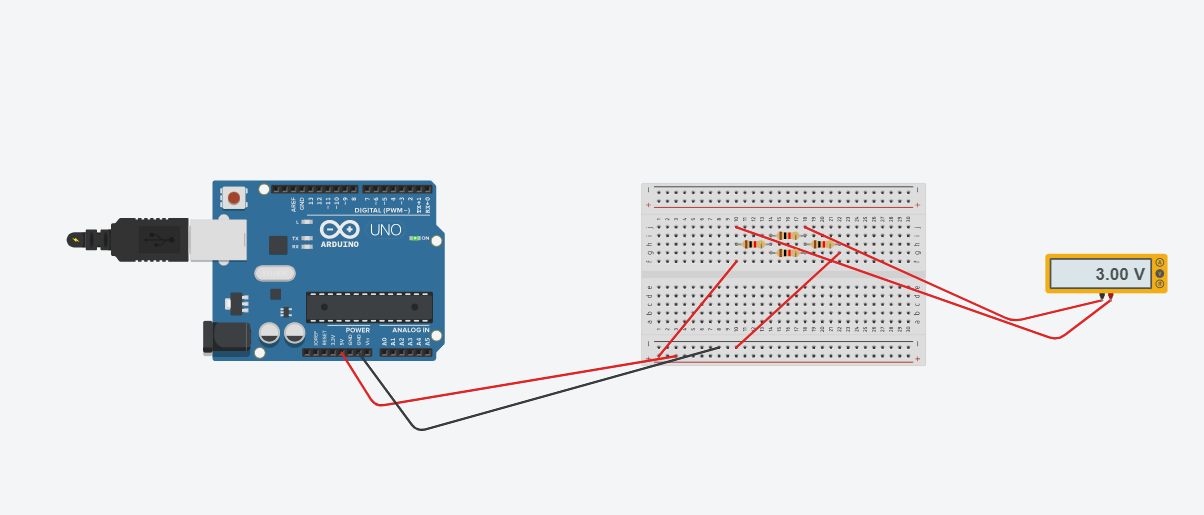
V=(220/(220+220))\*5

V=(220/440)\*5

V=(1/2)\*5

V=2.5v

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



## Activity 1.5: Voltage Divider circuit readings from Breadboard circuit.

Diagram

Description automatically generated

## Activity 1.6: LED Circuits

Each resistor Value

220

220

Total resistance Calculation

R= R1+R2

R=220+220

R=440

Measured Resistance

440

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

Due to temperature having impact on a resistor this causes the resistor to have tolerance. The real value is shown by the simulator due to absence of external or internal disturbance. Henceforth, we are able to conclude that the numbers are different in real life as shown in a simulator .

different.

## Activity 1.7: Current Measurement

Calculation of current flowing into LED

V=I\*R

5 = I \* 110

I = 5/110

I= 0.45A

Hence,

I = 45.45mA

Actual measured value of current

50

Why might they be different?

The discrepancy could be caused by the heating effect, which causes a circuit's current to flow less freely and its resistance to go up a bit.

## Activity 1.8: Fritzing for 4 switches & LEDS

Diagram

Description automatically generated

# Activity 1.9: Fritzing for Number 0-7

Diagram, schematic

Description automatically generated Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated Diagram, schematic

Description automatically generated

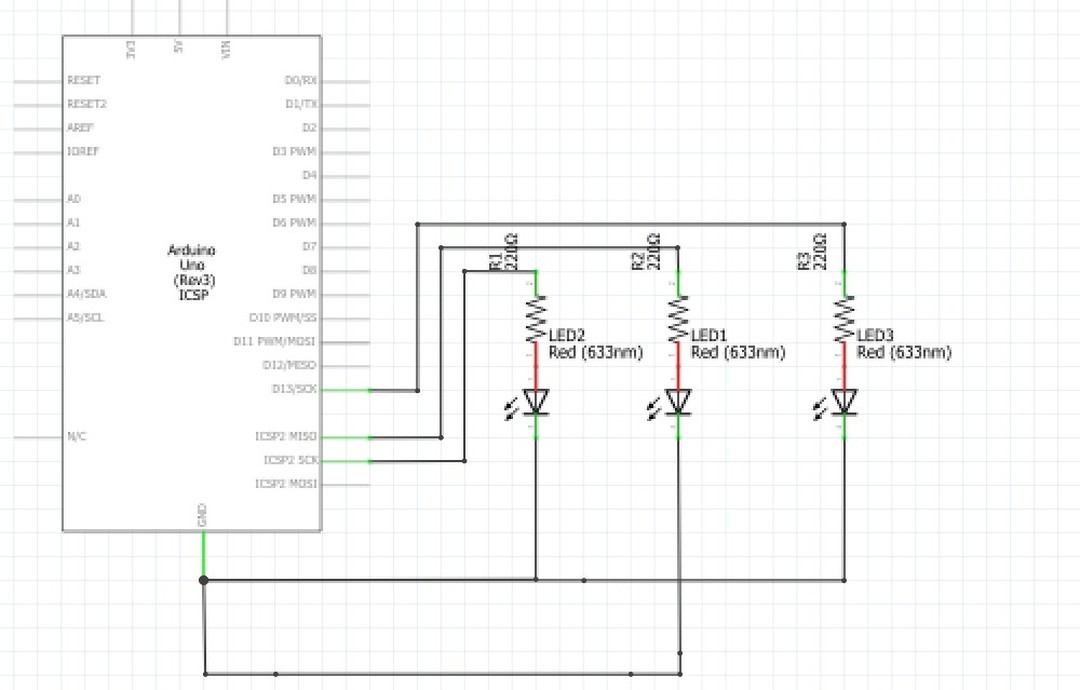
Diagram, schematic

Description automatically generated Diagram, schematic

Description automatically generated

# Workbook 2

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



63 as binary, including working

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

//

Name = Alan Shah

// Student number = 2329767

// we know that

binary value of 63 is 00111111 in 8 digit

// So the led should blink upto 6 times.

void setup()

{

pinMode(13, OUTPUT);

}

void loop()

{

int i;

for(i=0 ; i<6; i++){

digitalWrite(13, HIGH);

delay(1000); // Wait for 1000 millisecond(s)

digitalWrite(13, LOW);

delay(1000); // Wait for 1000 millisecond(s)

}

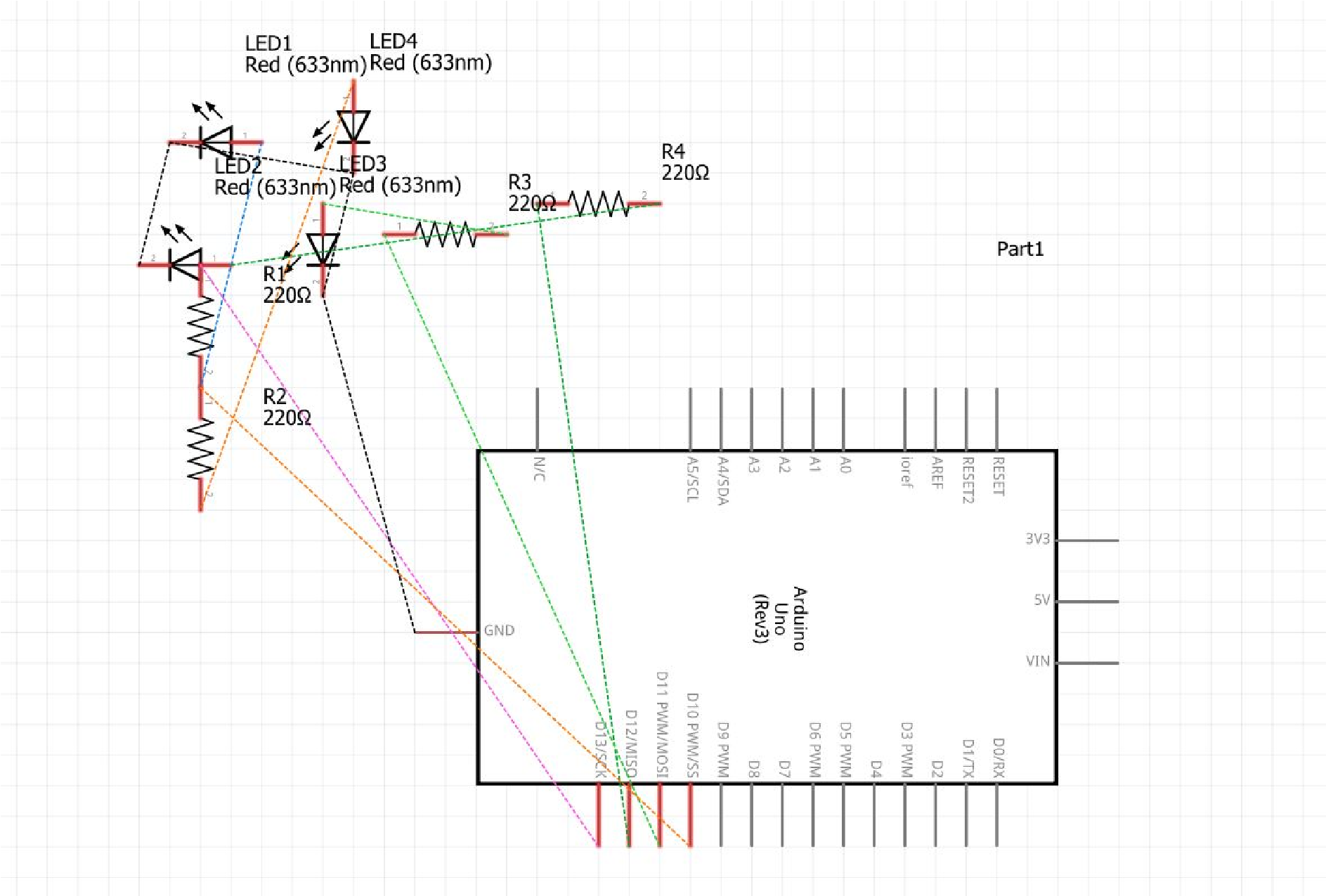
delay(3000);

//3000 millisecond delay

}

## 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

// C++ code

//

Name = Alan Shah

// Student number = 2329767

//

int rled=13;

int gled=12;

int Rled=11;

int Gled=10;

void setup()

{

pinMode(rled,OUTPUT);

pinMode(gled,OUTPUT);

pinMode(Rled,OUTPUT);

pinMode(Gled,OUTPUT);

}

void loop()

{

//Binary 1

digitalWrite(rled, LOW);

digitalWrite(gled, LOW);

digitalWrite(Rled, LOW);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 2

digitalWrite(rled, LOW);

digitalWrite(gled, LOW);

digitalWrite(Rled, LOW);

digitalWrite(Gled, HIGH);

delay(1000);

//Binary 3

digitalWrite(rled, LOW);

digitalWrite(gled, LOW);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 4

digitalWrite(rled, LOW);

digitalWrite(gled, LOW);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, HIGH);

delay(1000);

//Binary 5

digitalWrite(rled, LOW);

digitalWrite(gled, HIGH);

digitalWrite(Rled, LOW);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 6

digitalWrite(rled, LOW);

digitalWrite(gled, HIGH);

digitalWrite(Rled, LOW);

digitalWrite(Gled, HIGH);

delay(1000);

//Binary 7

digitalWrite(rled, LOW);

digitalWrite(gled, HIGH);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 8

digitalWrite(rled, LOW);

digitalWrite(gled, HIGH);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, HIGH);

delay(1000);

//Binary 9

digitalWrite(rled, HIGH);

digitalWrite(gled, LOW);

digitalWrite(Rled, LOW);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 10

digitalWrite(rled, HIGH);

digitalWrite(gled, LOW);

digitalWrite(Rled, LOW);

digitalWrite(Gled, HIGH);

delay(1000);

//Binary 11

digitalWrite(rled, HIGH);

digitalWrite(gled, LOW);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 12

digitalWrite(rled, HIGH);

digitalWrite(gled, HIGH);

digitalWrite(Rled, LOW);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 13

digitalWrite(rled, HIGH);

digitalWrite(gled, HIGH);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 14

digitalWrite(rled, HIGH);

digitalWrite(gled, HIGH);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, LOW);

delay(1000);

//Binary 15

digitalWrite(rled, HIGH);

digitalWrite(gled, HIGH);

digitalWrite(Rled, HIGH);

digitalWrite(Gled, HIGH);

delay(1000);

}

## Activity 2.3: Traffic Lights

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

Chart

Description automatically generated

// C++ code

//

Name = Alan Shah

// Student number = 2329767

//

// Define the pins for the traffic lights

int redPin = 4;

int yellowPin = 3;

int greenPin = 2;

// Define the duration of each light in milliseconds

int redDuration = 5000;

int yellowDuration = 2000;

int greenDuration = 5000;

int redYellowDuration = 1000;

void setup() {

// Set the traffic light pins to output mode

pinMode(redPin, OUTPUT);

pinMode(yellowPin, OUTPUT);

pinMode(greenPin, OUTPUT);

}

void loop() {

// Red light

digitalWrite(redPin, HIGH);

delay(redDuration);

// Red and yellow lights

digitalWrite(redPin, HIGH);

digitalWrite(yellowPin, HIGH);

delay(redYellowDuration);

digitalWrite(redPin, LOW);

digitalWrite(yellowPin, LOW);

// Green light

digitalWrite(greenPin, HIGH);

delay(greenDuration);

// Yellow light

digitalWrite(greenPin, LOW);

digitalWrite(yellowPin, HIGH);

delay(yellowDuration);

digitalWrite(yellowPin, LOW);

}

Arduino Program for Step 4 i.e. 4 LEDs

# Workbook 3

## Activity 3.1: Circuit Diagram of Button & LED

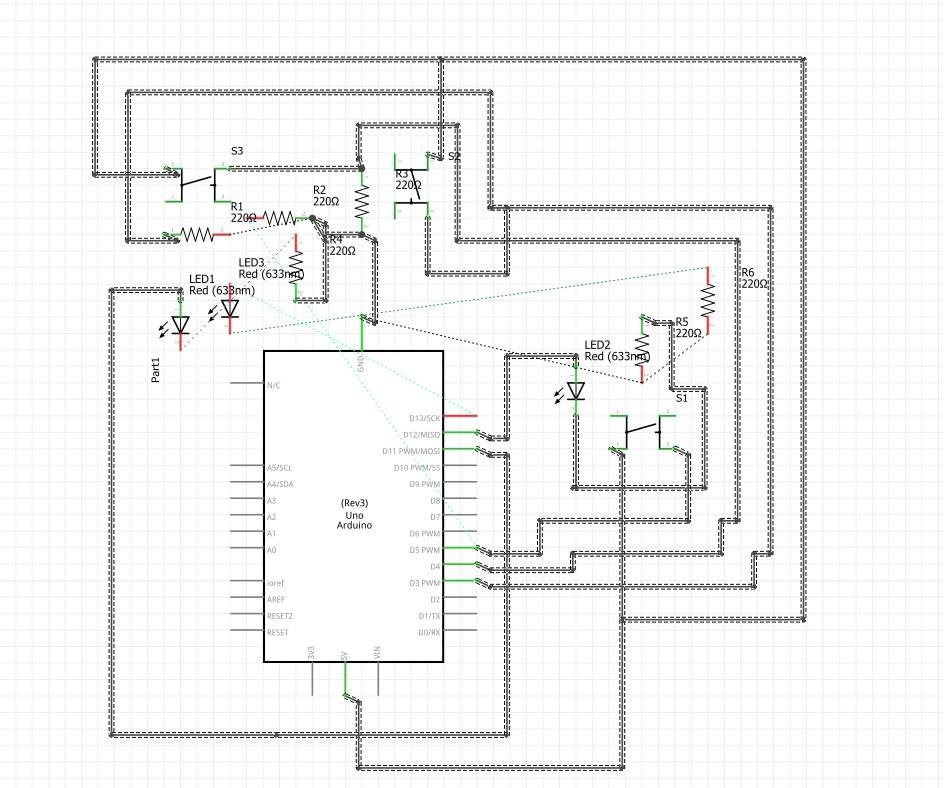
Fritzing

Diagram

Description automatically generated

## Activity 3.2: 3 Switches & Led

Fritzing Circuit Diagram



Arduino Program

//

Name = Alan Shah

// Student number = 2329767

int switchPin1 = 2;

int switchPin2 = 3;

int switchPin3 = 4;

int gled = 12;

int rled = 13;

void setup() {

pinMode(switchPin1, INPUT);

pinMode(switchPin2, INPUT);

pinMode(switchPin3, INPUT);

pinMode(gled, OUTPUT);

pinMode(rled, OUTPUT);

}

void loop() {

if (digitalRead(switchPin1) == HIGH) {

digitalWrite(gled, HIGH);

delay(1000);

digitalWrite(gled, LOW);

}

if (digitalRead(switchPin2) == HIGH) {

digitalWrite(rled, HIGH);

delay(2000);

digitalWrite(rled, LOW);

}

if (digitalRead(switchPin3) == HIGH) {

digitalWrite(gled, HIGH);

delay(3000);

digitalWrite(gled, LOW);

}

}

## Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS)

Fritzing

Diagram

Description automatically generated

Arduino Program

//

Name = Alan Shah

// Student number = 2329767

int value= 255;

int pin[8]= {6,7,8,9,10,11,12,13};

void setup()

{

for(int i=0; i<8;i++)

{

pinMode(pin[i], OUTPUT);

}

}

void loop()

{

myFunction(val);

}

void myFunction(int value)

{

int i;

for(i=0;i<8;i++)

{

if(val//2==0)

{

digitalWrite(pin[i],LOW);

}

else

{

// if the value is odd the pin will be high

digitalWrite(pin[i],HIGH);

}

value=value/2;

}

}

# Workbook 4

## Activity 4.1: Serial Port

Fritzing

Diagram, schematic

Description automatically generated

Arduino Program

//

Name = Alan Shah

// Student number = 2329767

int pushButton=2;

char sname[7]= "Alan";

long snum=220214 ;

void setup()

{

pinMode(pushButton, INPUT);

Serial.begin(9600);

}

void loop()

{

Serial.print("The name of the student is: ");

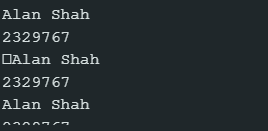
Serial.println(sname);

Serial.print("The student number is: ");

Serial.println(snum);

}

Screen Shot of Serial Port



## Activity 4.2: Serial Port binary to decimal

Code

//

Name = Alan Shah

// Student number = 2329767

int switchInput[8] = {6,7,8,9,10,11,12,13};

int i;

void setup() {

// put your setup code here, to run once:

for(i=0; i<8;i++)

{

pinMode(switchInput[i],INPUT);

}

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

int n;

float result=0;

for(i=0;i<8;i++)

{

int switchState = digitalRead(switchInput[i]);

if(switchState == HIGH)

{

n = 1;

}

else

{

n = 0;

}

result = result+n\*pow(2,i);

}

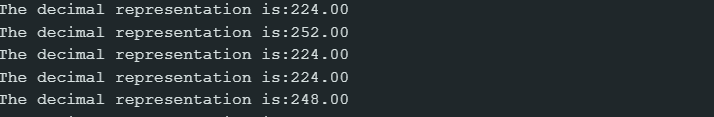
Serial.print("The decimal representation is:");

Serial.println(result);

delay(1000);

}

Screen Shot of Serial Port



## Activity 4.3: Calibrating Analogue Information

Code

//

Name = Alan Shah

// Student number = 2329767

int switchInput=6;

void setup()

{

pinMode(switchInput, INPUT);

Serial.begin(9600);

}

void loop()

{

float voltage;

int switchState = digitalRead(switchInput);

int sensorRead = analogRead(A0);

if(switchState == HIGH)

{

voltage = sensorRead \* (5.0/1024.0);

float resistance=(voltage\*250.0)/5.0;

Serial.print("The voltage measured is:- ");

Serial.println(voltage);

Serial.print("Measured Pot Resistance is:- ");

Serial.println(resistance);

}

}

Pot Resistance Clockwise

398k Ω

Pot Resistance Anti-clockwise

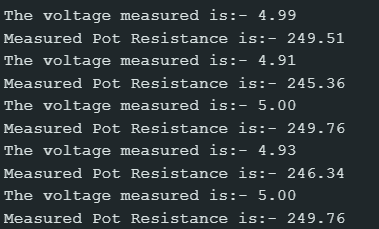
586k Ω

Sample of Values

Pot Resistance against Voltage change

|  |  |
| --- | --- |
| Pot Resitance | Voltage Measured |
| 249.51 Ω | 4.99V |
| 245.36 Ω | 4.91V |
| 249.76 Ω | 5.00V |
| 240.34 Ω | 4.93V |
| 249.76 Ω | 5.00V |

Screen Shot of Meaningful Serial Port Output, not just numbers



## Activity 4.4: Temperature Sensor & Serial Port

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

//

Name = Alan Shah

// Student number = 2329767

const int button = 2;

int buttonState = 0;

void setup(){

pinMode(button, INPUT);

Serial.begin(9600);

}

void loop(){

buttonState = digitalRead(button);

if (buttonState == HIGH){

float analogueReading = analogRead(A0);

float degreesC = (analogueReading \* 500.0)/ 1024.0;

float fahrenheit = degreesC\*1.8+32;

Serial.print("The Temperature: ");

Serial.print(fahrenheit);

Serial.println(" Degree Fahrenheit");

delay(1000);

exit(0);}

}

Screen Shot of Serial Port

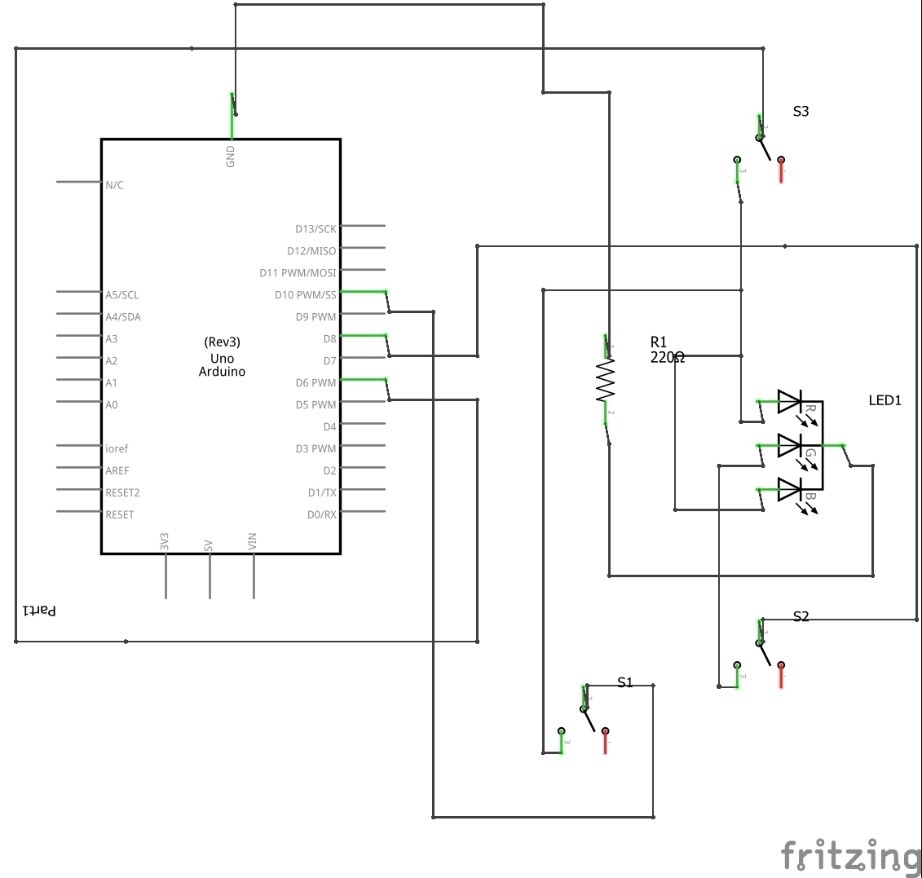
Graphical user interface, text, website

Description automatically generated

# Workbook 5

## Activity 5.1: RGB Led and switches

Fritzing



Arduino Program

//

Name = Alan Shah

// Student number = 2329767

int rled= 9;

int bled= 10;

int gled= 11;

int rpin= 2;

int bpin= 3;

int gpin= 4;

void setup()

{

pinMode(rled, OUTPUT);

pinMode(bled, OUTPUT);

pinMode(gled, OUTPUT);

pinMode(rpin, INPUT);

pinMode(bpin, INPUT);

pinMode(gpin, INPUT);

}

void loop()

{

if (digitalRead(rpin)==HIGH)

{

digitalWrite(rled, HIGH);

}

else

{

digitalWrite(rled, LOW);

}

if (digitalRead(bpin)==HIGH)

{

digitalWrite(bled, HIGH);

}

else

{

digitalWrite(bled, LOW);

}

if (digitalRead(gpin)==HIGH)

{

digitalWrite(gled, HIGH);

}

else

{

digitalWrite(gled, LOW);

}

}

## Activity 5.2: Distance Sensor

Arduino Code

//

Name = Alan Shah

// Student number = 2329767

int trigPin= 7;

int

echoPin= 3;

void setup() {

Serial.begin (9600); //view value received

pinMode(echoPin, INPUT\_PULLUP);

pinMode(trigPin, OUTPUT);

}

void loop()

{

long duration, inches, cm;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

inches = duration/74/2;

cm = duration/29/2;

Serial.print("The distance is:- ");

Serial.println(duration);

Serial.print("The inches is:- ");

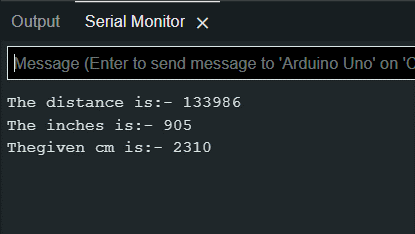
Serial.println(inches);

Serial.print("The

given cm is:- ");

Serial.println(cm);

}

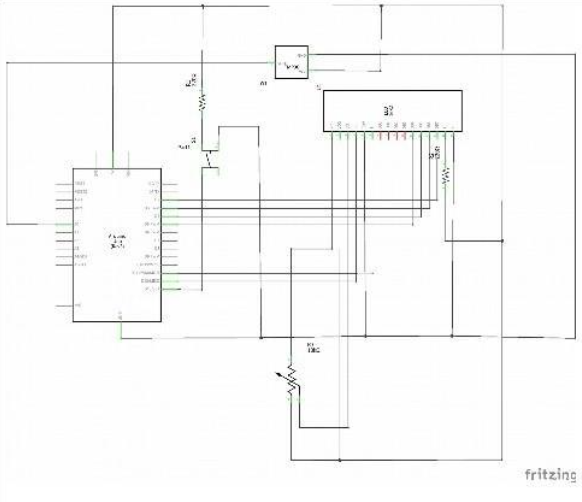


Take a picture of your distance sensor and include it here, please reduce the size and quality as it will be too large else ☺



## Activity 5.3: 1602 LCD Display

Fritzing



Arduino Program

#include <dht.h> ;

//

Name = Alan Shah

// Student number = 2329767

#include <LiquidCrystal.h>;

LiquidCrystal lcd(12,11, 2,3,4,5);

int dPin= 8;

dht DHT;

void setup() {

  lcd.begin(16,2);

  Serial.begin(9600);

}

void loop() {

  lcd.clear();

  int readData = DHT.read11(dPin);

  float t = DHT.temperature;

  float f= ((t\*9.0)/5.0+32.0);

  lcd.setCursor(2, 0);

     // Read temperature

  lcd.print(t);

  lcd.print("C");

  lcd.setCursor(2, 1);

  lcd.print(f);

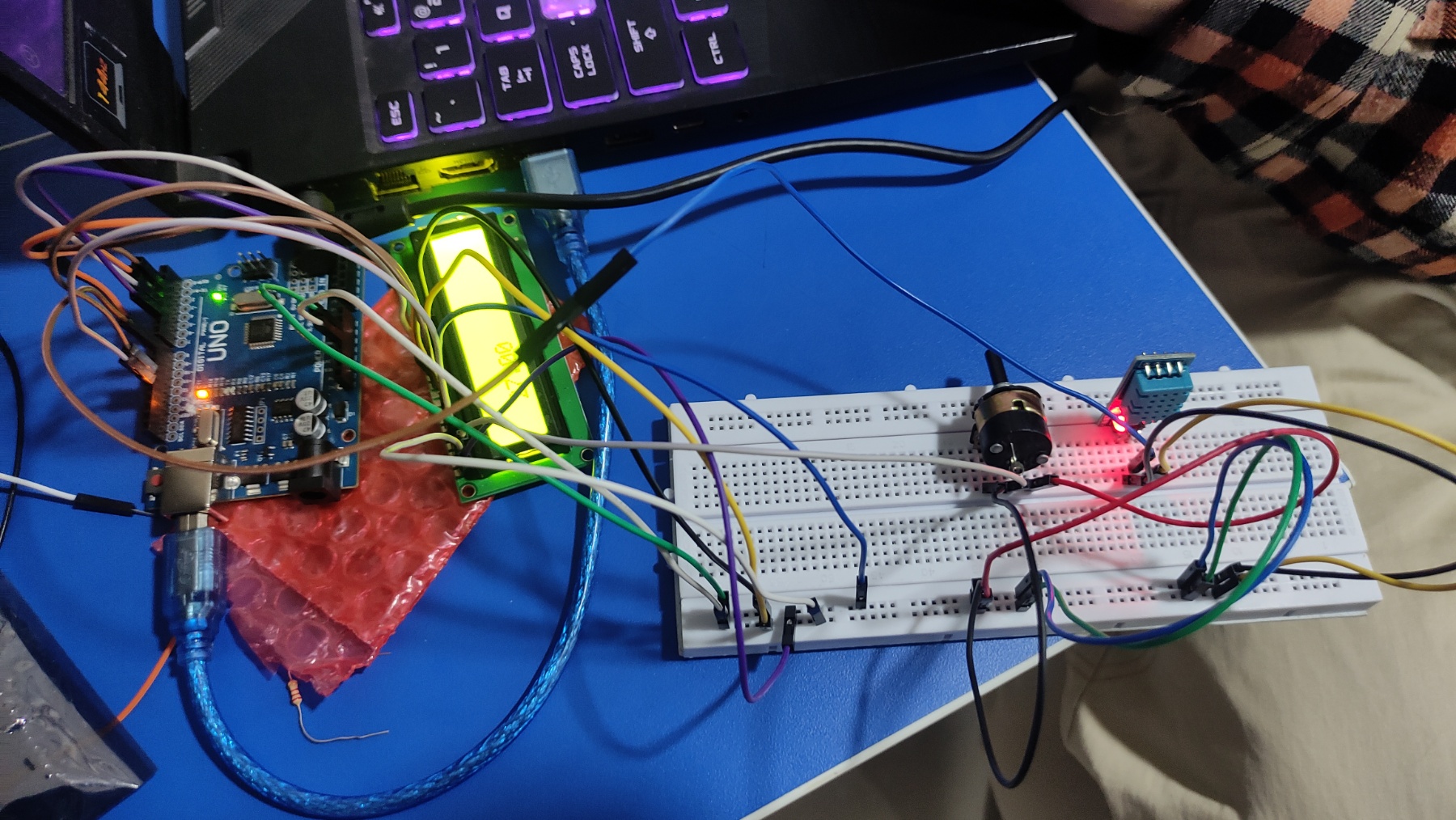
  lcd.print("F");

  delay(2000); // wait two seconds

  Serial.print(t);

}

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

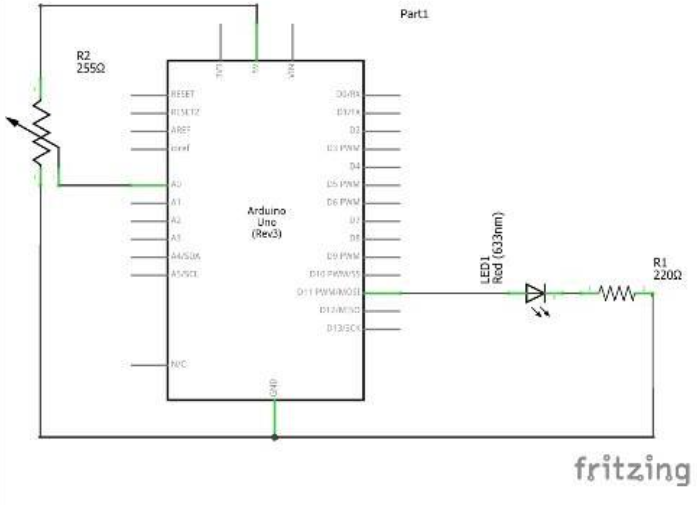




# Workbook 6

## Activity 6.1: PWM

Fritzing



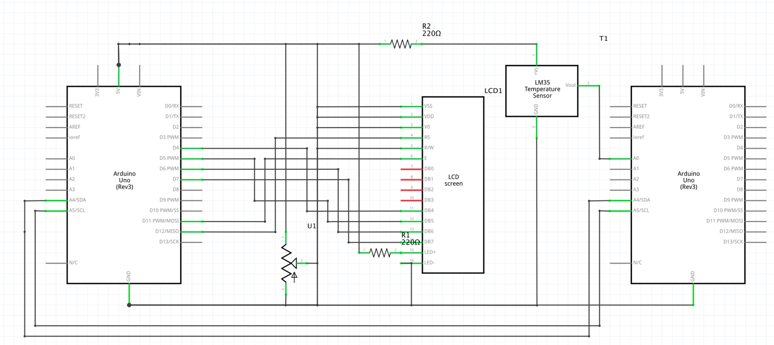
Arduino Program

// Alan Shah 2329767

int rPin = 3;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(rPin, OUTPUT);  
}  
  
void loop()  
{  
  int val = analogRead(A0);  
  int bright = map(val , 0, 1023, 0, 255);  
  Serial.println(bright);  
  delay(1000);  
  analogWrite(rPin, bright);  
}

# Workbook 7

## Activity 7.1: 2 Arduinos – using Digital Pins



Fritzing

Arduino Program

// Alan Shah 2329767

#include <LiquidCrystal.h>

#include <Wire.h>

LiquidCrystal lcd(12, 11, 4,5,6,7);

void setup()

{

lcd.begin(16,2);

Serial.begin(9600);

Wire.begin(8);//begin I2C or TWI having address 8

/\*act as slave component(dataReceiver) to display temp reading

when master component(dataSendor) want to write temp data to it\*/

}

void loop()

{

Wire.onReceive(dataReceived);

}

void dataReceived(int x)//x means how many data received

{

int CelsiusVal = Wire.read();

Serial.print("Temperature in celsius: ");

Serial.print(CelsiusVal);

Serial.print("degC");

Serial.println();

int FahrenheitVal = Wire.read();

Serial.print("Temperature in fahrenheit: ");

Serial.print(FahrenheitVal);

Serial.print("degF");

Serial.println();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Celsius: ");

lcd.print(CelsiusVal);

lcd.print("C");

lcd.setCursor(0, 1);

lcd.print("Fahrenheit: ");

lcd.print(FahrenheitVal);

lcd.print("F");

Serial.print(x);

}

## Activity 7.2: 2 Arduinos – using Serial I/O

Fritzing

A diagram of a computer

Description automatically generated with low confidence

Arduino Program

// Alan Shah 2329767

#include <Wire.h>

#include <DHT.h>

#define DHTPIN 2 // DHT11 signal pin

#define DHTTYPE DHT11 // DHT11 sensor type

DHT dht(DHTPIN, DHTTYPE);

void setup()

{

Wire.begin();

Serial.begin(9600);

dht.begin();

}

void loop()

{

float temperature = dht.readTemperature(); // read temperature from DHT11

float fahrenheit = (temperature \* 9.0 / 5.0) + 32.0;

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.print(" °C, ");

Serial.print(fahrenheit);

Serial.println(" °F");

//transmit to device with address 8

Wire.beginTransmission(8);

Wire.write((int)temperature); //send temperature

Wire.write((int)fahrenheit); //send Fahrenheit temperature

Wire.endTransmission();

delay(1000);

}

# Workbook 8

Diagram, schematic

Description automatically generated

## Activity 8.1: Stepper Motor Circuit Diagram

Circuit Diagram

Arduino Program

// Alan Shah 2329767

#include <Stepper.h>

// Defines the number of steps per rotation

const int stepsPerRevolution = 2038;

// Creates an instance of stepper class

// Pins entered in sequence IN1-IN3-IN2-IN4 for proper step sequence

Stepper myStepper = Stepper(stepsPerRevolution, 8, 10, 9, 11);

void setup() {

// Nothing to do (Stepper Library sets pins as outputs)

}

void loop() {

// Rotate CW slowly at 5 RPM

myStepper.setSpeed(5);

myStepper.step(stepsPerRevolution);

delay(1000);

// Rotate CCW quickly at 10 RPM

myStepper.setSpeed(10);

myStepper.step(-stepsPerRevolution);

delay(1000);

}

## Activity 8.2: 2 Stepper Motors

Arduino Program

// Alan Shah 2329767

#define FULLSTEP 4

#define HALFSTEP 8

// Creates two instances

// Pins entered in sequence IN1-IN3-IN2-IN4 for proper step sequence

AccelStepper stepper1(HALFSTEP, 8, 10, 9, 11);

AccelStepper stepper2(FULLSTEP, 4, 6, 5, 7);

void setup() {

// set the maximum speed, acceleration factor,

// initial speed and the target position for motor 1

stepper1.setMaxSpeed(1000.0);

stepper1.setAcceleration(50.0);

stepper1.setSpeed(200);

stepper1.moveTo(2038);

// set the same for motor 2

stepper2.setMaxSpeed(1000.0);

stepper2.setAcceleration(50.0);

stepper2.setSpeed(200);

stepper2.moveTo(-2038);

}

void loop() {

// Change direction once the motor reaches target position

if (stepper1.distanceToGo() == 0)

stepper1.moveTo(-stepper1.currentPosition());

if (stepper2.distanceToGo() == 0)

stepper2.moveTo(-stepper2.currentPosition());

// Move the motor one step

stepper1.run();

stepper2.run();

}

# Workbook 9

## Activity 9.1: Windscreen Wiper Code using Servos & Temperature Sensor

Arduino Code

#include <Servo.h>

#include <dht.h>

#define DHT11\_PIN A0

#define SERVO\_PIN 9

dht DHT;

Servo servo;

void setup() {

  servo.attach(SERVO\_PIN);

  Serial.begin(9600);

}

void loop() {

  int chk = DHT.read11(DHT11\_PIN); // read the temperature and humidity sensor

  Serial.print("Temperature: ");

  Serial.print(DHT.temperature);

  Serial.print(" °C");

  Serial.print("\t");

  Serial.print("Humidity: ");

  Serial.print(DHT.humidity);

  Serial.println(" %");

  if (DHT.temperature > 25) { // if it's hot, turn on the wipers

    servo.write(90); // set the servo to the middle position

    delay(1000);

    servo.write(0); // set the servo to the left position

    delay(1000);

    servo.write(180); // set the servo to the right position

    delay(1000);

    servo.write(90); // set the servo to the middle position

    delay(1000);

  }

  delay(2000); // wait at least 2 seconds between readings for the DHT11 sensor

}

# 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 adetailed 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 hand-written work is not acceptable.

### Marking

# All sections carry equal marks.