|  |  |  |
| --- | --- | --- |
| **Student Name** |  | **Student Number** |
| **Himal Singh Airee** |  | **2332315** |

**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..............................................................................................5 Activity 1.4: 3V Calculations from either the 5V supply or 3.3V supply...................................................6

Activity 1.5: Voltage Divider circuit readings from Breadboard circuit....................................................6 Activity 1.6: LED Circuits.........................................................................................................................7

Activity 1.7: Current Measurement........................................................................................................8 Activity 1.8: Fritzing for 4 switches & LEDS.............................................................................................9

Activity 1.9: Fritzing for Number 0-7.....................................................................................................10

Workbook 2...............................................................................................................................................11

Activity 2.1: LED Flashing to show decimal number 63 as binary...........................................................11

Activity 2.2: 4 LED’s for counting up in binary from 0 to 15...................................................................12

Workbook 3...............................................................................................................................................14

Activity 3.1: Circuit Diagram of Button & LED........................................................................................14

Activity 3.2: 3 Switches & Led...........................................................................................................15

Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS)..........................................................................17

Workbook 4...............................................................................................................................................19 Activity 4.1: Serial Port..........................................................................................................................19

Activity 4.2: Serial Port binary to decimal..............................................................................................22

Activity 4.3: Calibrating Analogue Information......................................................................................24

Activity 4.4: Temperature Sensor & Serial Port......................................................................................27

Workbook 5...............................................................................................................................................29 Activity 5.1: RGB Led and switches........................................................................................................29

Activity 5.2: LED Matrix MAZE...............................................................................................................31

Activity 5.3: 1602 LCD Display................................................................................................................33

Workbook 6...............................................................................................................................................36

Activity 6.1: PWM..................................................................................................................................36 Workbook 7...............................................................................................................................................38 Activity 7.1: Windscreen Wiper Code using Servos & Temperature Sensor...........................................38

Individual Project (50%).............................................................................................................................39

Rationale................................................................................................................................................39

Timescales.............................................................................................................................................39

Equipment.............................................................................................................................................39

The Project............................................................................................................................................39 Step 1 produce a detailed description of your project.......................................................................39 Step 2 Circuit Diagram & Fritzing Schematic......................................................................................39 Step 3 A Program...............................................................................................................................40

Step 4 Testing.....................................................................................................................................40 Step 5 Conclusions.............................................................................................................................40 Layout................................................................................................................................................40 Demonstrations.................................................................................................................................40

Marking.............................................................................................................................................40

All sections carry equal marks...................................................................................................................40

## Workbook 1

Activity 1.1: Actual voltage across 5V breadboard pins.

Enter the Value you got here from Step 5.

5V

Activity 1.2: Actual voltage across 3.3V breadboard pins.

3V

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 we measure value using multi meter in a simulator such as TinkerCad we get an exact value and when we read a value with real multi meter we get different value because simulator is a software where we measure virtually (computer). In simulator there is no effect of time, temperature and humidity. But in real world there is effect of time, temperature and humidity. When read by multi meter we didn’t get exact value because according to time we need to calibrate the multi meter to give accurate value. Ambient temperature helps in charge loss so we didn’t get exact value.

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

4.84 The tolerance is 16 percent on a 5-volt supply.

Five volts - 4,84 volts = 0,16 volts, 16%.

Activity 1.3: Potential Divider Calculations

Show the working on how you achieved 2.5V

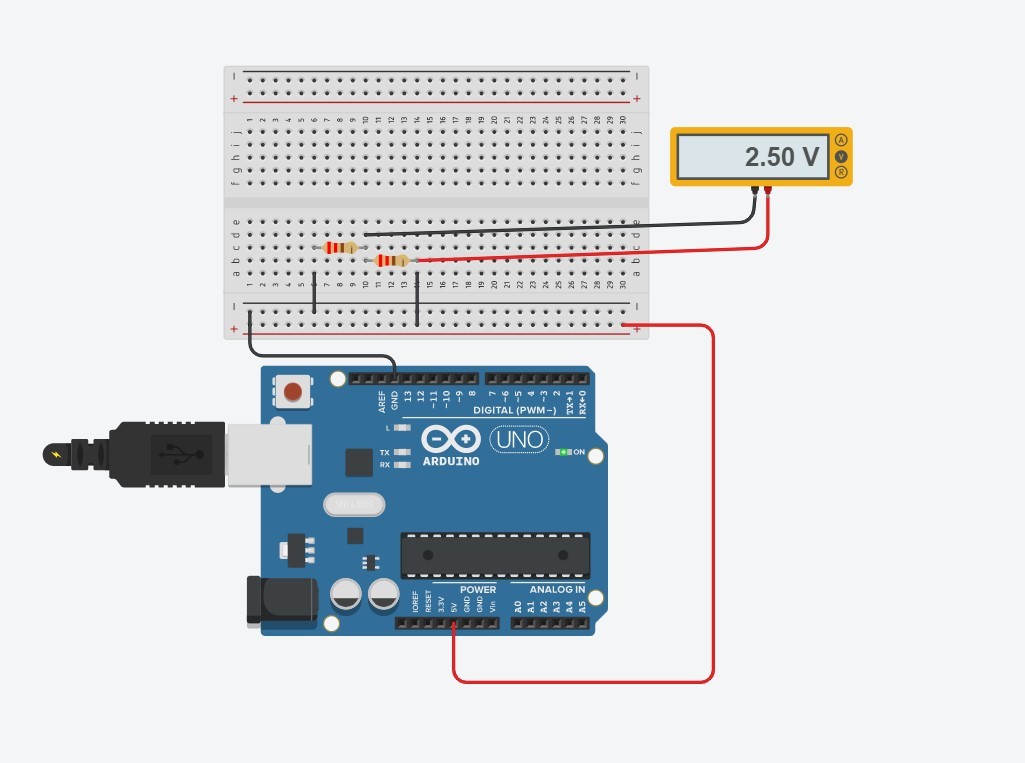
V=(R/R Total)\*V supply

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

V=(220/440)\*5

V=(1/2)\*5

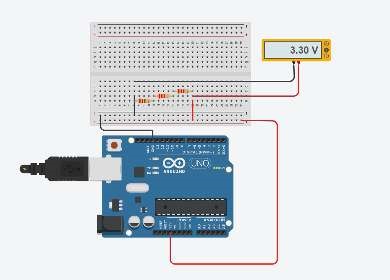
V=2.5v



Form 3.3-volt supply if 3 volts are required, we connect the 2.7 ohm resistance thus 0.3 volt is

passing through and the output is 3 volts.

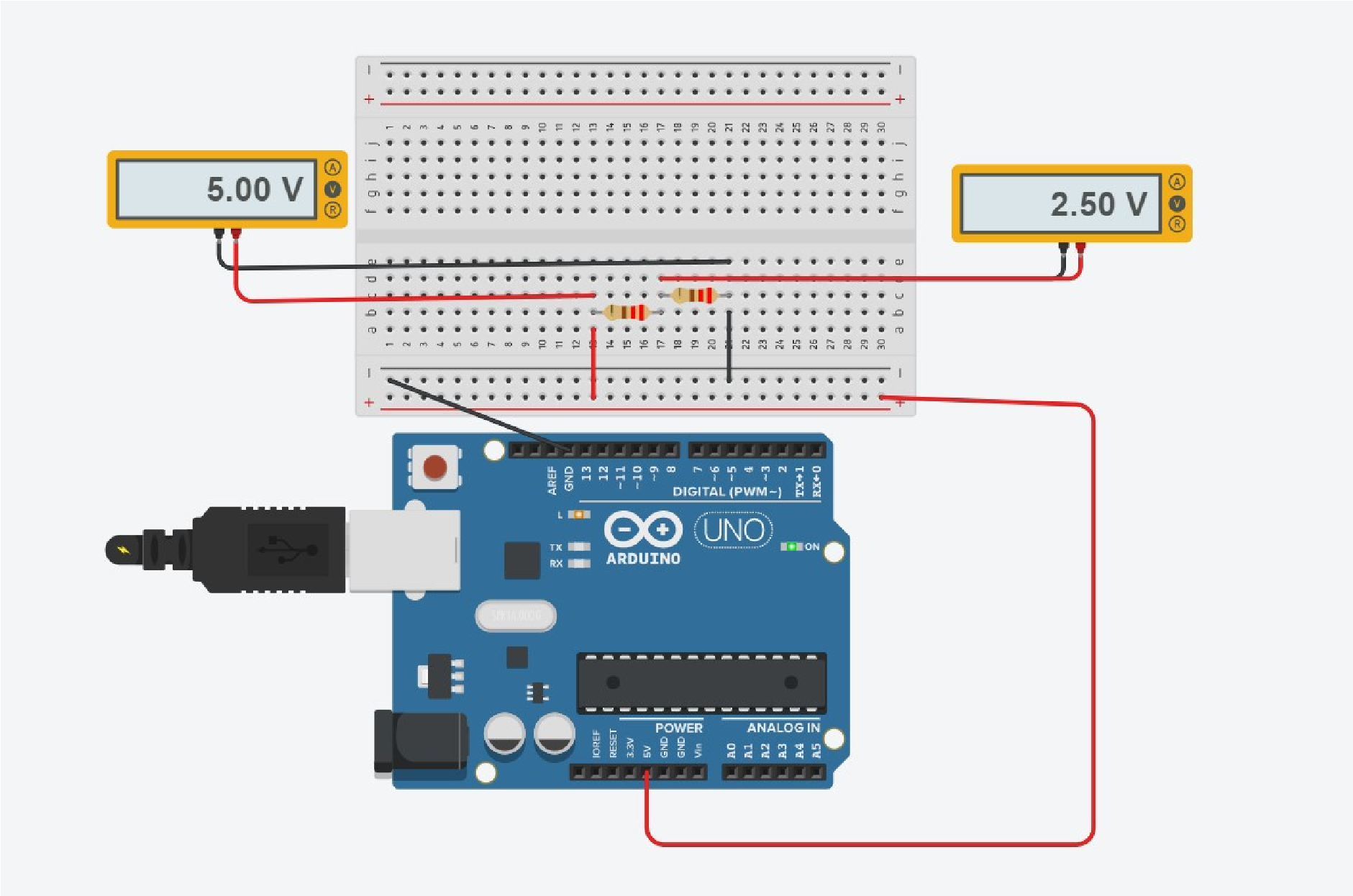
Form 5 volt supply we connect the 3 ohms resistance to 2 volts and receive 3 volts at output



.

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

Activity 1.5: Voltage Divider circuit readings from Breadboard circuit.



Activity 1.6: LED Circuits

Each resistor Value

|  |  |  |
| --- | --- | --- |
| 220 ohm |  | 220 ohm |

Total resistance Calculation

1/Req=1/R1+1/R2

1/Req=1/220+1/220

1/Req=1/110

Req=110 ohm

Measured Resistance

111 ohm,

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

The resistance has an influence on the temperature which makes the resistance tolerant.

The simulation shows the real value since no external or internal losses are seen, therefore we can say that the simulation shows the ideal value although the losses exist in practice such that the value differs from the original value.

Activity 1.7: Current Measurement

Calculation of current flowing into LED

30 milli amp

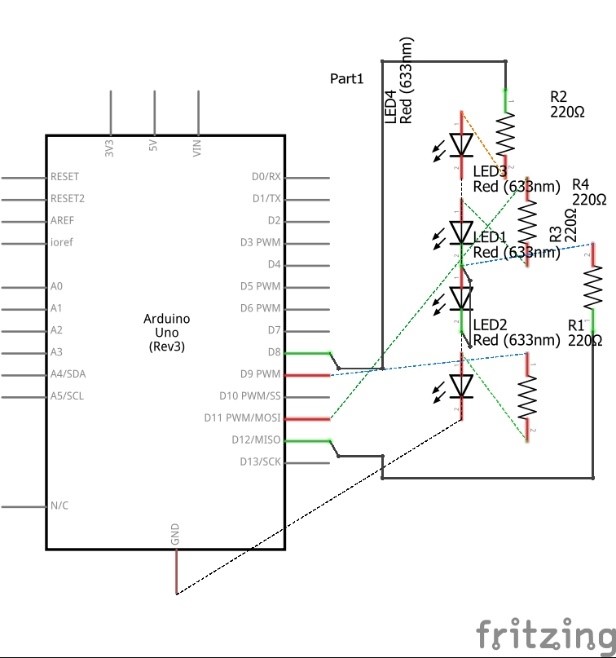
Actual measured value of current

27 milli amp

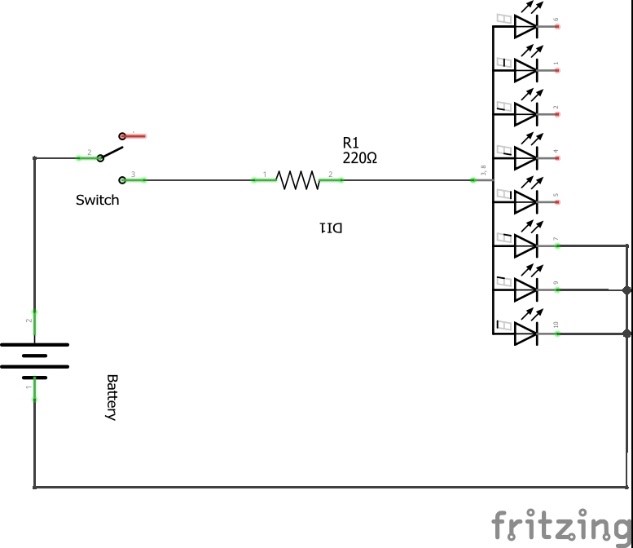
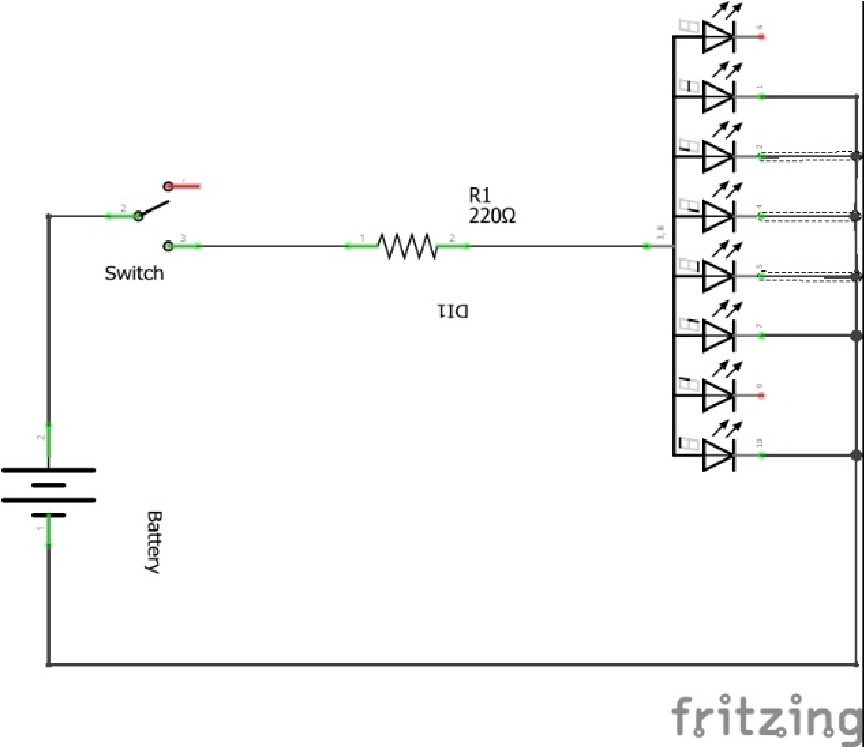
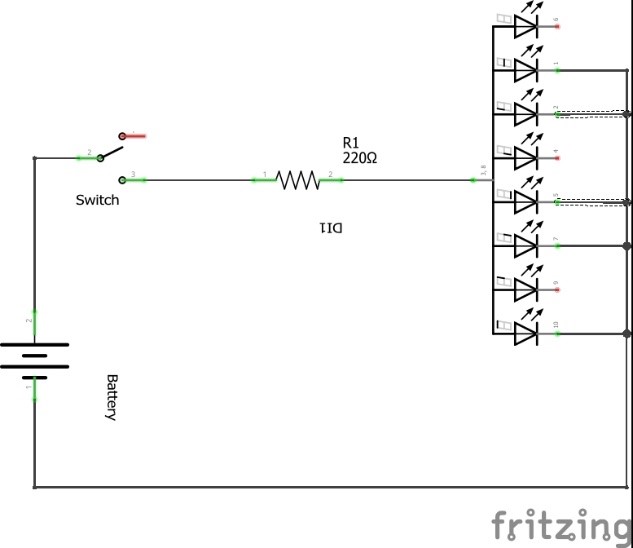
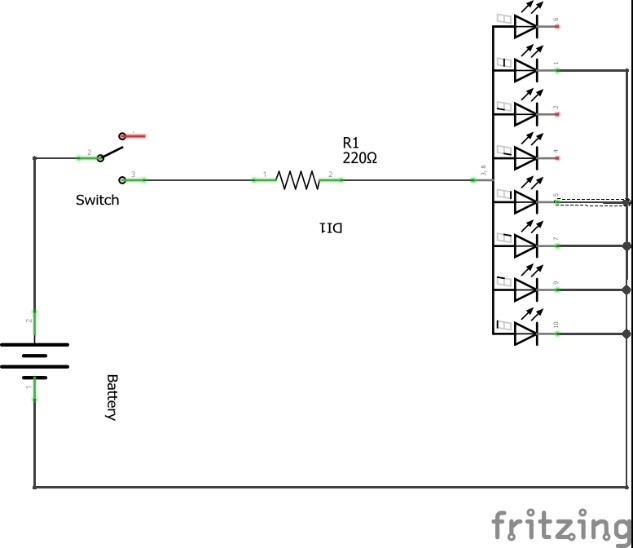
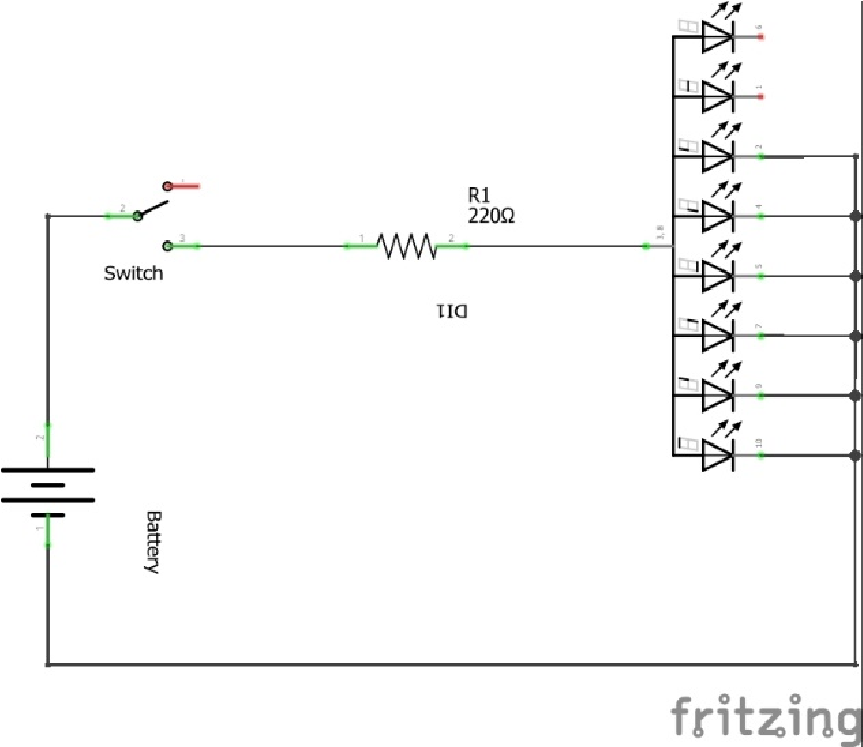
Why might they be different?

This is the reason why the resistance value increases a little, which is why the present is essentially declining.

Activity 1.8: Fritzing for 4 switches & LEDS



Activity 1.9: Fritzing for Number 0-7

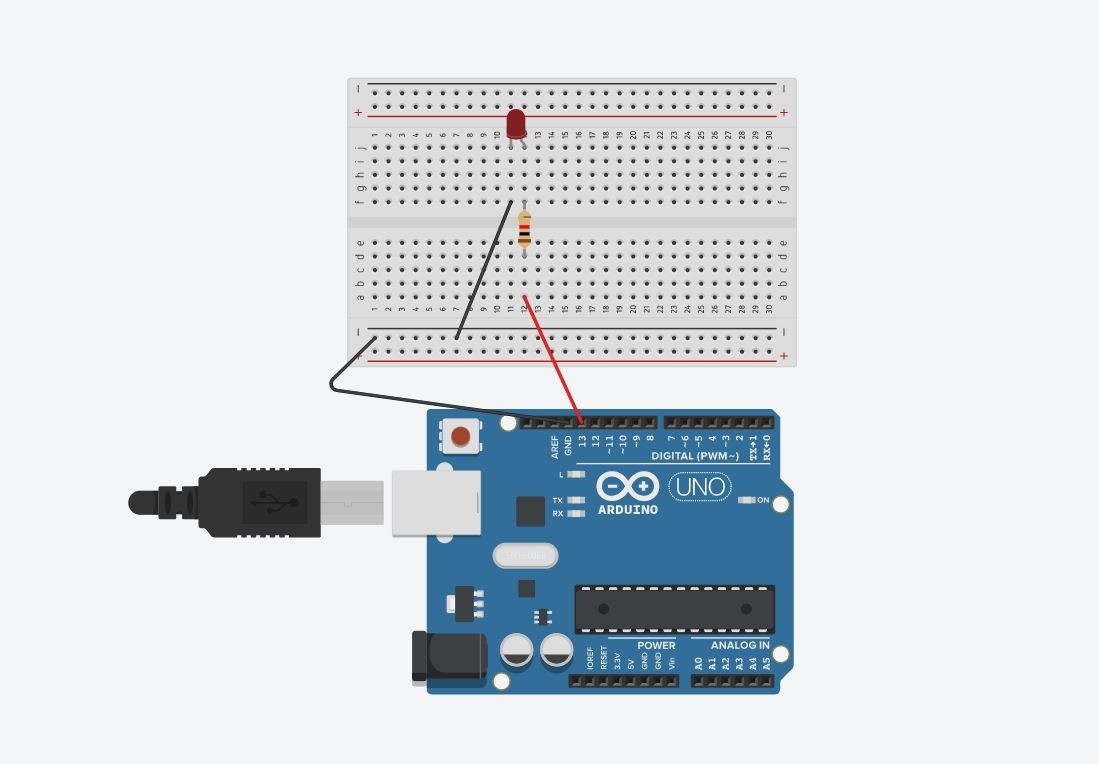


## Workbook 2

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

63

as binary, including working



In the above circuit we code in Arduino to blink LED of 63 decimal number to binary number which

is equal to 111111. Firstly, the digital pin 13 was initialized as the output and 6 times the loop was

used using for loop, whereby, 6 time the light the light blinked. We delay 1000 millisecond which is

equals to one second that means LED is on for one second and off for one second. When the loop is

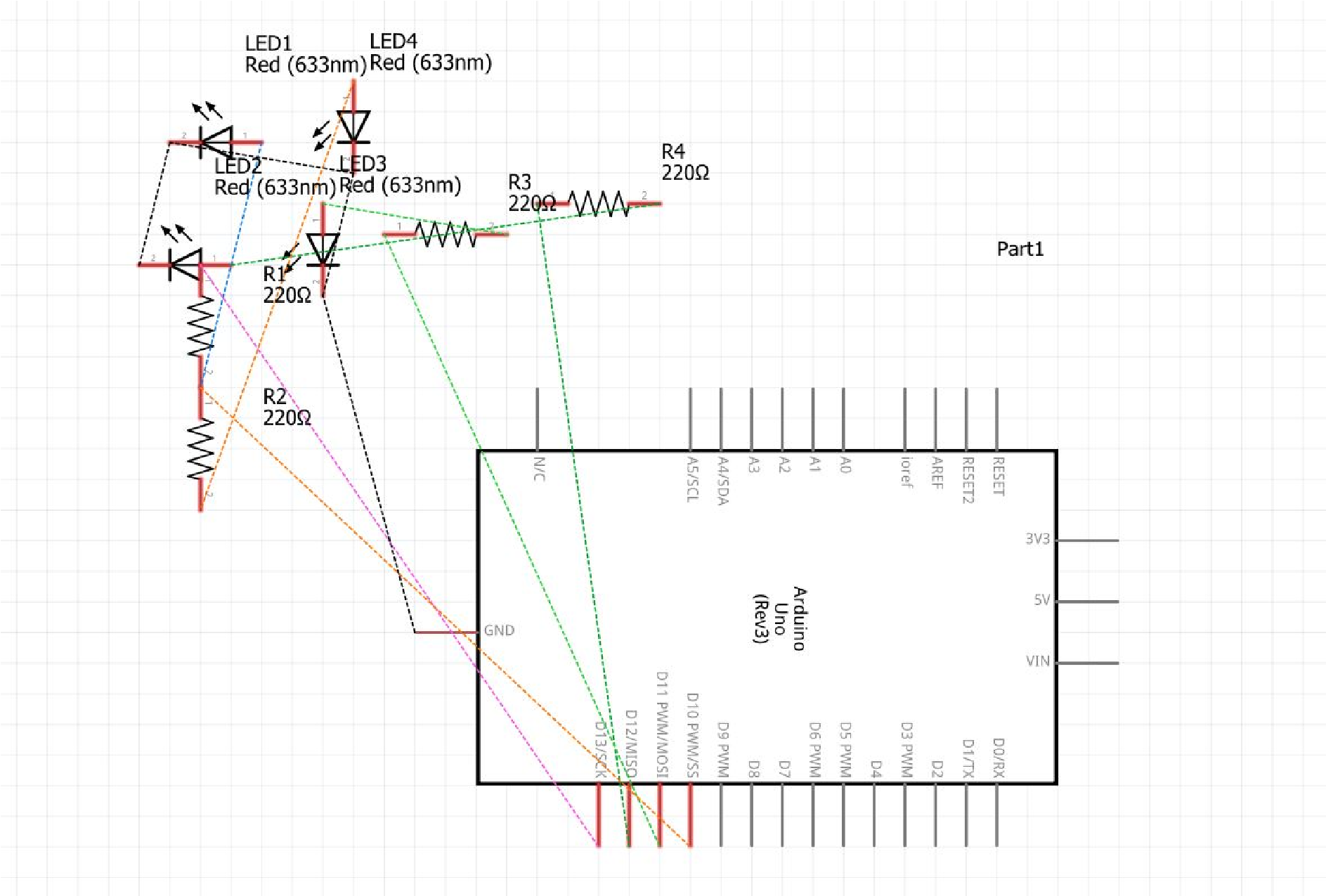
over LED is off for 5000 milliseconds and again loop resumes.

|  |  |
| --- | --- |
| // C++ code void setup()  {  pinMode(13, OUTPUT);  }  void loop()  { int count=0; |  |

lOMoARcPSD|24017307

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



lOMoARcPSD|24017307

Arduino Program for Step 4 i.e. 4 LEDs

|  |
| --- |
| // C++ code int first\_led=10; int second\_led=11; int third\_led=12; int fourth\_led=13; int time=2000;  void setup()  {  pinMode(first\_led, OUTPUT); pinMode(second\_led, OUTPUT); pinMode(third\_led, OUTPUT); pinMode(fourth\_led, OUTPUT);  }  void loop()  {  // for 0 digitalWrite(first\_led, LOW); digitalWrite(second\_led, LOW); digitalWrite(third\_led, LOW); digitalWrite(fourth\_led, LOW); delay(time); |

|  |  |  |
| --- | --- | --- |
| //for 1 digitalWrite(first\_led, HIGH); digitalWrite(second\_led, LOW); digitalWrite(third\_led, LOW); digitalWrite(fourth\_led, LOW); delay(time);    // for 2 digitalWrite(first\_led, LOW); digitalWrite(second\_led, HIGH); digitalWrite(third\_led, LOW); digitalWrite(fourth\_led, LOW); delay(time); // for 3 digitalWrite(first\_led, HIGH); digitalWrite(second\_led, HIGH); digitalWrite(third\_led, LOW); digitalWrite(fourth\_led, LOW); delay(time); //for 4 digitalWrite(first\_led, LOW); digitalWrite(second\_led, LOW); digitalWrite(third\_led, HIGH); digitalWrite(fourth\_led, LOW); delay(time);    //for 5 | |  |
| digitalWrite(first\_led, HIGH);  //for 6  digitalWrite(second\_led, LOW);  digitalWrite(first\_led, LOW);  digitalWrite(third\_led, HIGH);  digitalWrite(second\_led, HIGH);  digitalWrite(fourth\_led, LOW);  digitalWrite(third\_led, HIGH); delay(time);  digitalWrite(fourth\_led, LOW); delay(time); |  |
| //for 7 |

lOMoARcPSD|24017307

lOMoARcPSD|24017307

//for 10 digitalWrite(first\_led, LOW); digitalWrite(second\_led, HIGH); digitalWrite(third\_led, LOW); digitalWrite(fourth\_led, HIGH); delay(time); //for 11 digitalWrite(first\_led, HIGH); digitalWrite(second\_led, HIGH);

lOMoARcPSD|24017307

// for 15

digitalWrite(first\_led, HIGH);

digitalWrite(second\_led, HIGH);

digitalWrite(third\_led, HIGH);

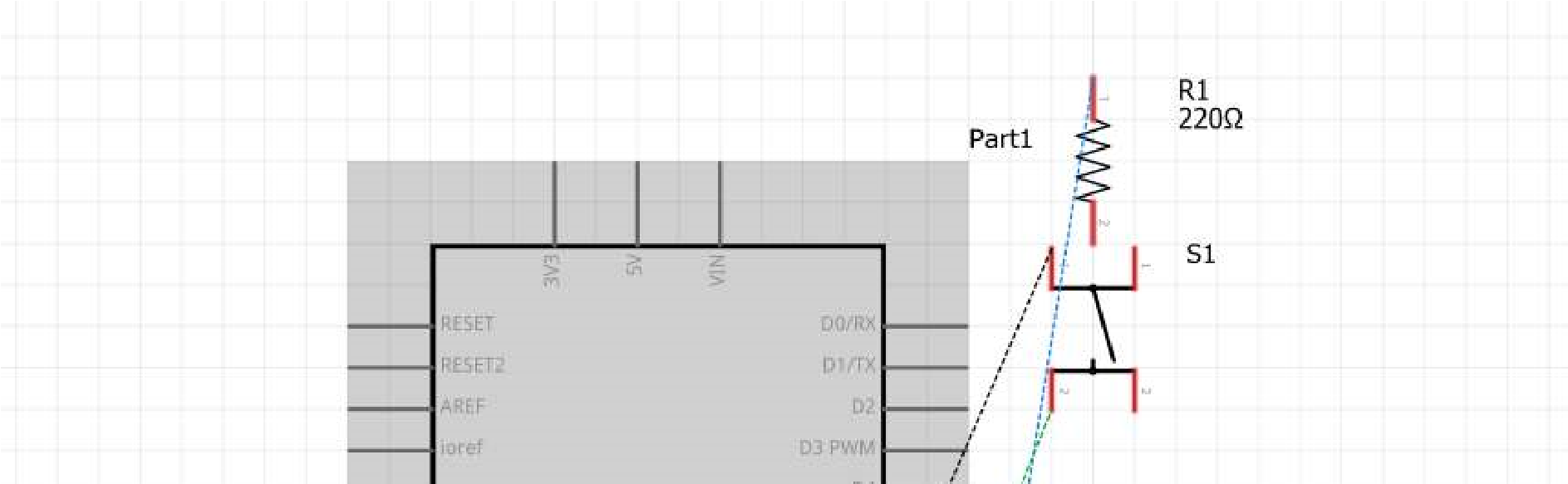
digitalWrite(fourth\_led, HIGH);

delay(time);

}

lOMoARcPSD|24017307

Fritzing



## Workbook 3

Activity 3.1: Circuit Diagram of Button & LED

|  |
| --- |
| lOMoARcPSD|24017307  Activity 3.2: 3 Switches & Led  Fritzing Circuit Diagram |
|  |
|  |

lOMoARcPSD|24017307

Arduino Program

|  |
| --- |
| // C++ code const int switchInput1= 2; const int switchInput2= 3; const int switchInput3= 4; const int Led1= 13; // const int Led2= 12; const int Led3= 11;  void setup() {  pinMode(switchInput1, INPUT); pinMode(switchInput2, INPUT); pinMode(switchInput3, INPUT);  pinMode(Led1, OUTPUT); pinMode(Led2, OUTPUT); pinMode(Led3, OUTPUT);  } void loop(){  int switchState1 = digitalRead(switchInputFirst); int switchState2 = digitalRead(switchInputSecond); int switchState3 = digitalRead(switchInputThird); |

|  |  |
| --- | --- |
| if(switchState1 == HIGH){ digitalWrite(LedFirst, HIGH);  delay(1000);  digitalWrite(LedFirst, LOW);  }  else if (switchState2 == HIGH){ digitalWrite(LedSecond, HIGH);  delay(2000);  digitalWrite(LedSecond, LOW);  }  else if(switchState3 == HIGH){ digitalWrite(LedThird, HIGH);  delay(3000); Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS) digitalWrite(LedThird, LOW);Fritzing | |
| }  } |  |
|  |  |

lOMoARcPSD|24017307

Arduino Program

|  |
| --- |
| // C++ code  // int Led1=6; int Led2=7; int Led3=8; int Led4=9; int Led5=10; int Led6=11; int Led7=12; int Led8=13; void setup()  {  pinMode(Led1, OUTPUT); pinMode(Led2, OUTPUT); pinMode(Led3, OUTPUT); pinMode(Led4, OUTPUT); pinMode(Led5, OUTPUT); pinMode(Led6, OUTPUT); pinMode(Led7, OUTPUT); pinMode(Led8, OUTPUT);  } int ledToDisplay[8]; |

Workbook 4

Activity 4.1: Serial Port

Fritzing

void displayOutput(int n)

{

int i=0;

if(n<=255){

while(n>0){

ledToDisplay[i]=n%2;

n=n/2;

i++;

}

}

}

void loop()

{

isplayValue=215;

int d

displ

ayOutput(displayValue);

digit

alWrite(Led1, ledToDisplay[1]);

digit

alWrite(Led2, ledToDisplay[2]);

digit

alWrite(Led3, ledToDisplay[3]);

digit

alWrite(Led4, ledToDisplay[4]);

digit

alWrite(Led5, ledToDisplay[5]);

digit

alWrite(Led6, ledToDisplay[6]);

digit

alWrite(Led7, ledToDisplay[7]);

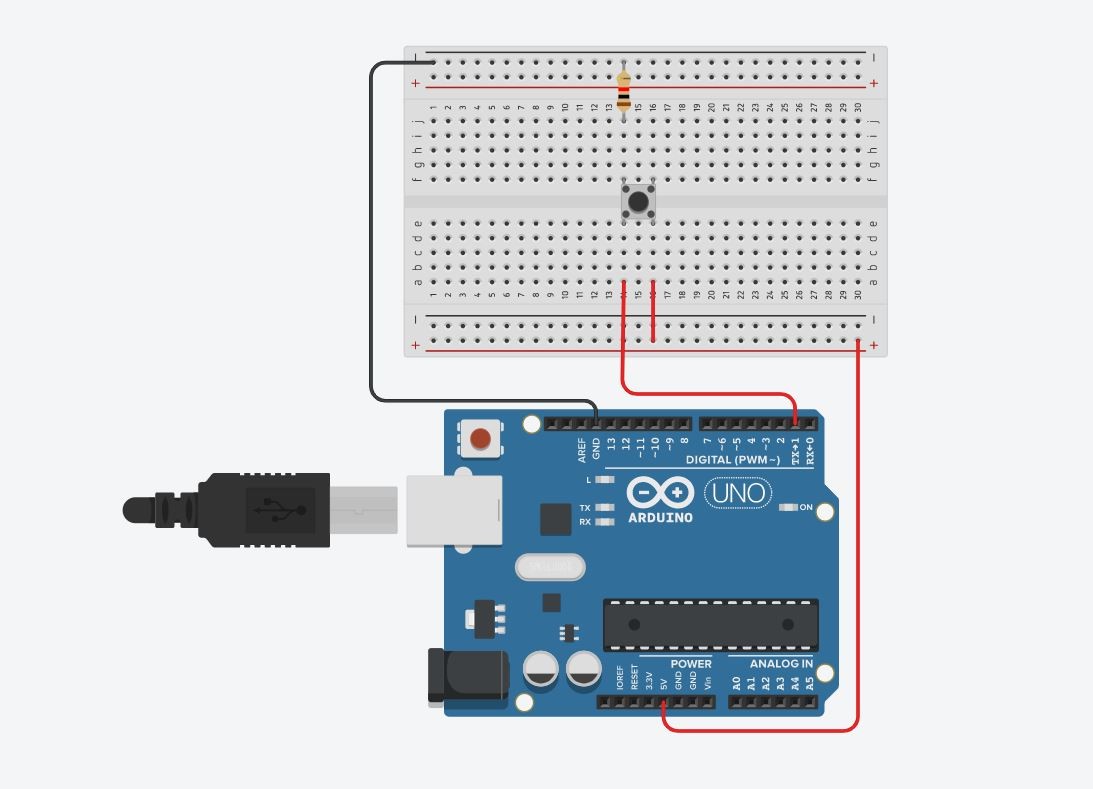
digit

alWrite(Led8, ledToDisplay[8]);

}

|  |
| --- |
| lOMoARcPSD|24017307  Arduino Program |
| // C++ code  // |
| int buttonPressed = 0;  void setup() {  Serial.begin(9600); pinMode(1, INPUT);  }  void loop() { buttonPressed = digitalRead(2); if (buttonPressed == HIGH) {  Serial.println("Saugat Karki"); Serial.println(2059754); delay(2000);  }  } |

Screen Shot of Serial Port



lOMoARcPSD|24017307

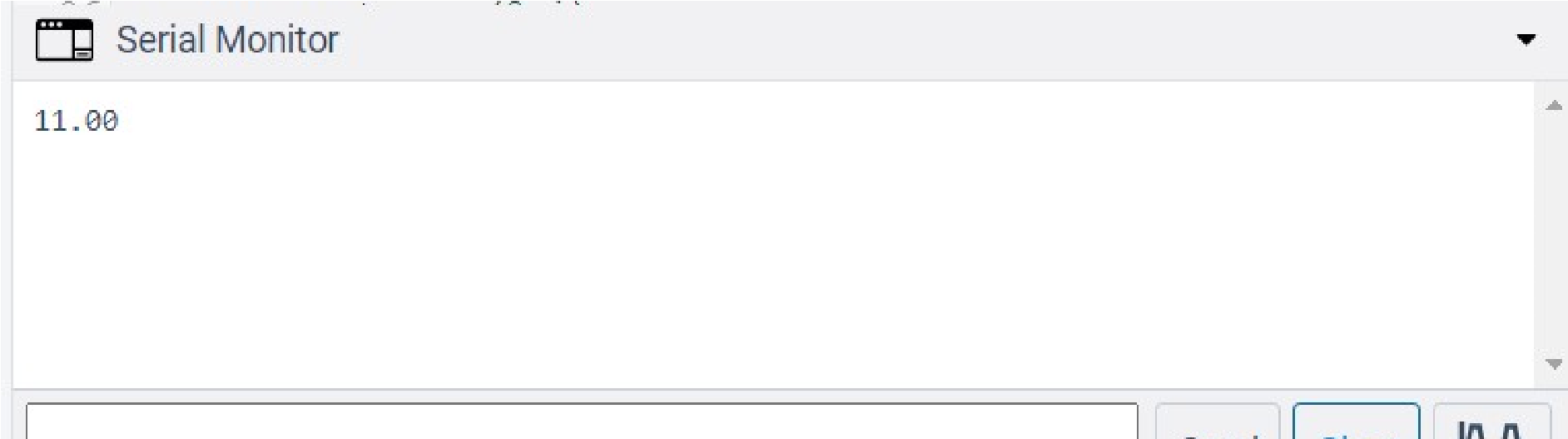
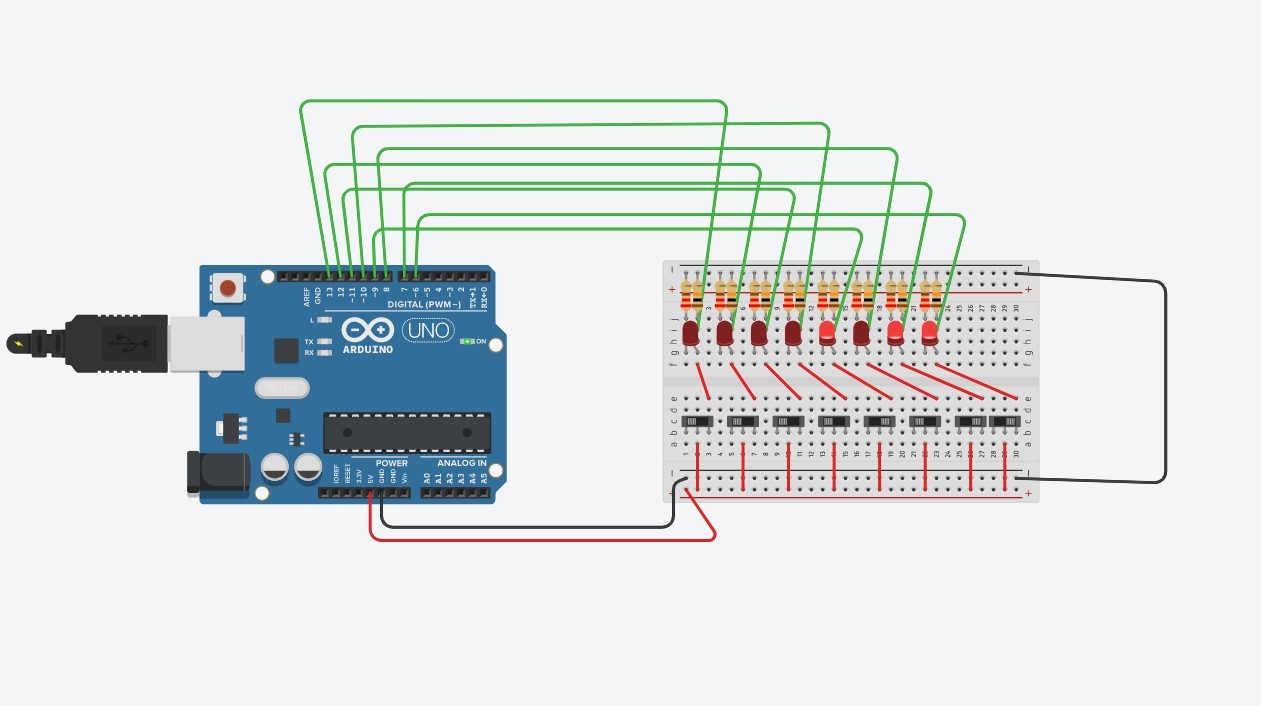
Activity 4.2: Serial Port binary to decimal

Code

|  |
| --- |
| // C++ code  // int input[] = {6,7,8,9,10,11,12,13}; int result[8]; void setup()  {  Serial.begin(1); for(int i=0; i<8; i++){ pinMode(input[i], INPUT);  }  }  void loop()  { calculateDecimal();  }  void calculateDecimal(){ float sum = 0; for(int i=0; i<8; i++){ result[i] = digitalRead(input[i]); if(result[i] == HIGH){ sum += pow(2,i);  }  } |

Serial.println(sum); delay(2000);

Screen Shot of Serial Port



}

lOMoARcPSD|24017307

Activity 4.3: Calibrating Analogue Information

Code

|  |
| --- |
| // C++ code const int button = 2; int buttonState = 0;  void setup(){ pinMode(button, INPUT);  Serial.begin(9600);  } void loop(){ buttonState = digitalRead(button); if (buttonState == HIGH){ int sensorValue = analogRead(A0); float voltage = sensorValue \*(5.0/1024.0);  Serial.print(voltage); Serial.println("V"); float resistance = (voltage\*250)/5;  Serial.print(resistance); Serial.println("KOhm"); delay(1000);  }    } |

Pot Resistance Clockwise

Pot Resistance Anti-clockwise

Sample of Values

Pot Resistance against Voltage change

Pot Resitance

Voltage Measured

249.76

Kohm

5

V

214.84

Kohm

4.30

V

159.91

Kohm

3.20

V

74.95

Kohm

1.50

V

24.90

0.50

V

Kohm

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

// C++ code

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");

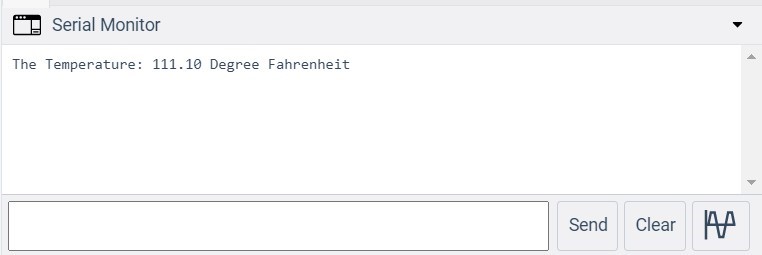
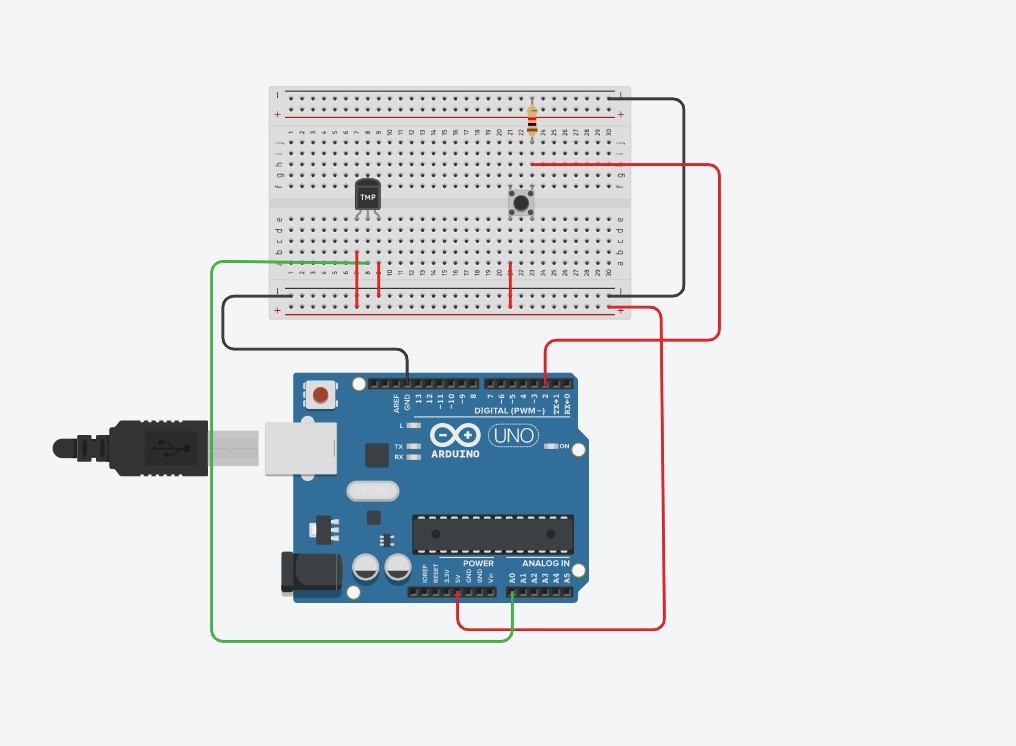
delay(1000);

exit(0);}

}

lOMoARcPSD|24017307

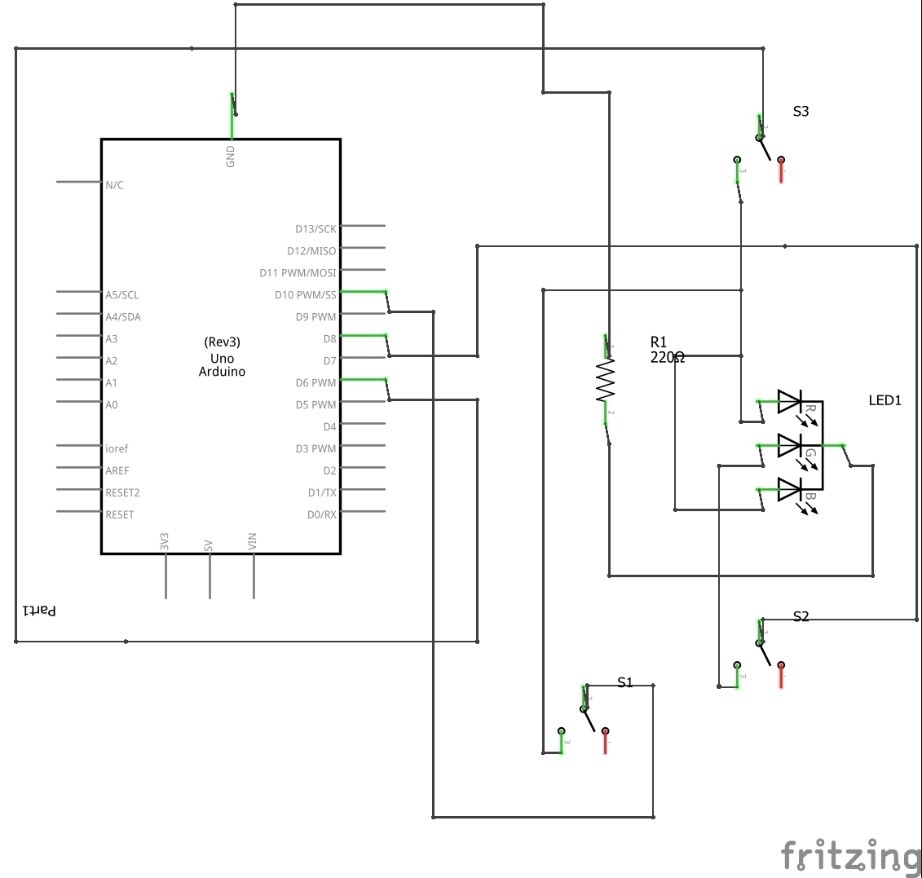
Screen Shot of Serial Port



## Workbook 5

Activity 5.1: RGB Led and switches

Fritzing



Arduino Program

|  |
| --- |
| // C++ code  // void setup()  {  pinMode(6, OUTPUT); pinMode(8, OUTPUT); pinMode(10, OUTPUT);  }  void loop()  { digitalWrite(6, HIGH); digitalWrite(8, HIGH); digitalWrite(10, HIGH);    } |

Activity 5.2: LED Matrix MAZE

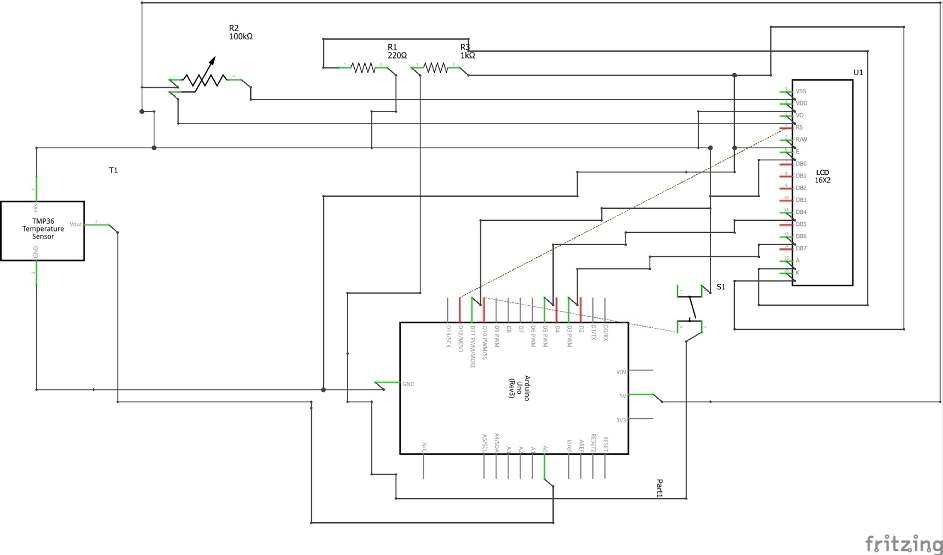
Arduino Code

There is no LED Matrix in tinkercad so we are instruct not to do in portfolio

Take a picture of your LED Matrix Maze 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 <LiquidCrystal.h>

float a=0;

int pushButton=0;

float b=0;

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

void setup() {

lcd.begin(16, 2);

pinMode(A0, INPUT);

pinMode(8,INPUT);

}

void loop() {

lcd.setCursor(0, 1);

a=analogRead(A0);

pushButton=digitalRead(8);

b=(a\*500)/1024;

if (pushButton==1){

lcd.print(b);

lcd.print(" SK|2021");

delay(1000);

}

else{

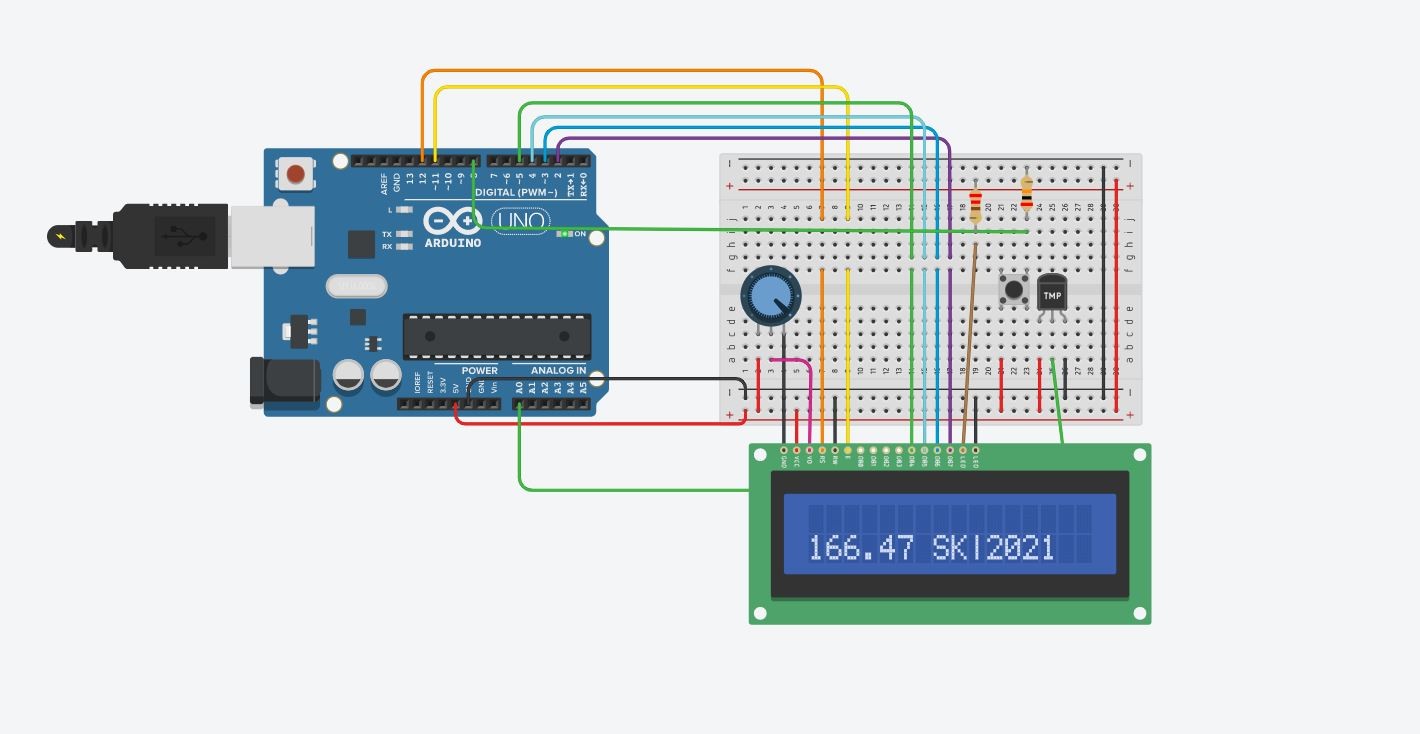
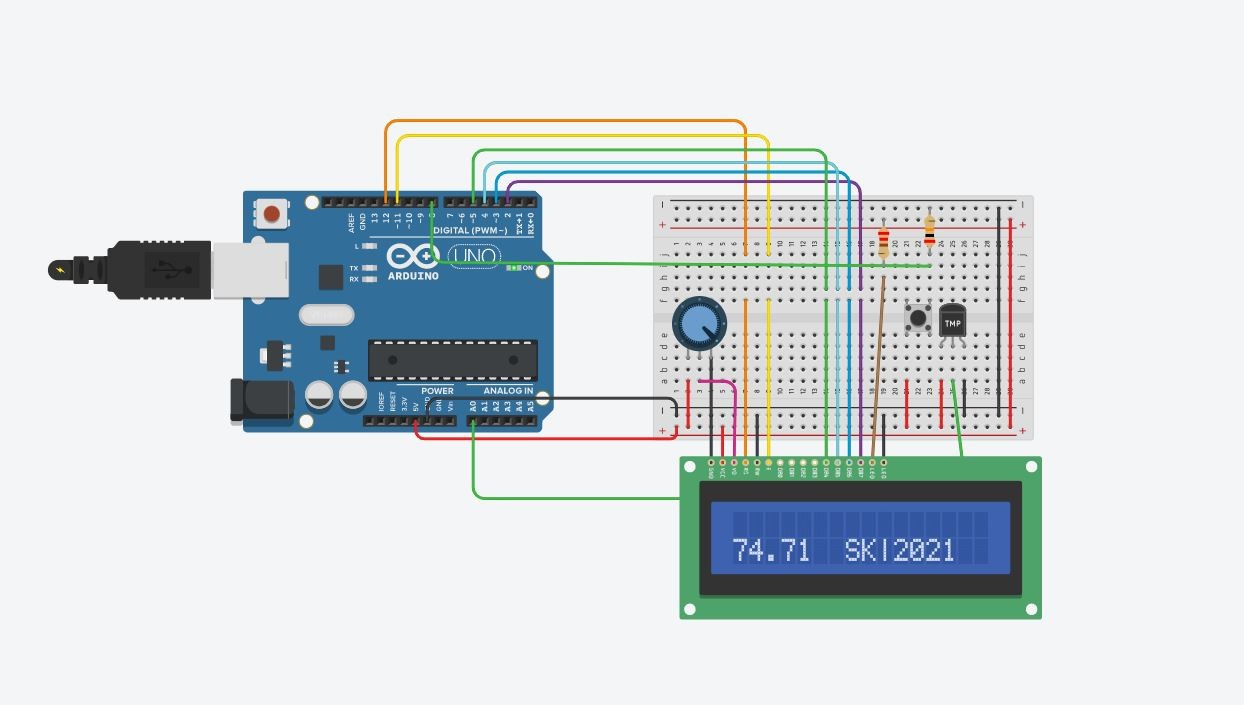
lcd.print(b\*1.8+32);

lcd.print(" SK|2021");

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

## Workbook 7

Activity 7.1: Windscreen Wiper Code using Servos & Temperature

Sensor

Arduino Code

## 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.

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.

### 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.