|  |  |  |
| --- | --- | --- |
| **Student Name** |  | **Student Number** |
| Mansoud Mansouri |  | 1916829 |

**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](#_Toc54016754)

[Activity 1.1: Actual voltage across 5V breadboard pins. 4](#_Toc54016755)

[Activity 1.2: Actual voltage across 3.3V breadboard pins. 4](#_Toc54016756)

[Activity 1.3: Potential Divider Calculations 5](#_Toc54016757)

[Activity 1.4: 3V Calculations from either the 5V supply or 3.3V supply 6](#_Toc54016758)

[Activity 1.5: Voltage Divider circuit readings from Breadboard circuit. 6](#_Toc54016759)

[Activity 1.6: LED Circuits 7](#_Toc54016760)

[Activity 1.7: Current Measurement 8](#_Toc54016761)

[Activity 1.8: Fritzing for 4 switches & LEDS 9](#_Toc54016762)

[Activity 1.9: Fritzing for Number 0-7 10](#_Toc54016763)

[Workbook 2 11](#_Toc54016764)

[Activity 2.1: LED Flashing to show decimal number 63 as binary. 11](#_Toc54016765)

[Activity 2.2: 4 LED’s for counting up in binary from 0 to 15. 12](#_Toc54016766)

[Workbook 3 14](#_Toc54016767)

[Activity 3.1: Circuit Diagram of Button & LED 14](#_Toc54016768)

[Activity 3.2: 3 Switches & Led 15](#_Toc54016769)

[Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS) 17](#_Toc54016770)

[Workbook 4 19](#_Toc54016771)

[Activity 4.1: Serial Port 19](#_Toc54016772)

[Activity 4.2: Serial Port binary to decimal 22](#_Toc54016773)

[Activity 4.3: Calibrating Analogue Information 24](#_Toc54016774)

[Activity 4.4: Temperature Sensor & Serial Port 27](#_Toc54016775)

[Workbook 5 29](#_Toc54016776)

[Activity 5.1: RGB Led and switches 29](#_Toc54016777)

[Activity 5.2: LED Matrix MAZE 31](#_Toc54016778)

[Activity 5.3: 1602 LCD Display 33](#_Toc54016779)

[Workbook 6 36](#_Toc54016780)

[Activity 6.1: PWM 36](#_Toc54016781)

[Workbook 7 38](#_Toc54016782)

[Activity 7.1: Windscreen Wiper Code using Servos & Temperature Sensor 38](#_Toc54016783)

[Individual Project (50%) 39](#_Toc54016784)

[Rationale 39](#_Toc54016785)

[Timescales 39](#_Toc54016786)

[Equipment 39](#_Toc54016787)

[The Project 39](#_Toc54016788)

[Step 1 produce a detailed description of your project. 39](#_Toc54016789)

[Step 2 Circuit Diagram & Fritzing Schematic 39](#_Toc54016790)

[Step 3 A Program 40](#_Toc54016791)

[Step 4 Testing 40](#_Toc54016792)

[Step 5 Conclusions 40](#_Toc54016793)

[Layout 40](#_Toc54016794)

[Demonstrations 40](#_Toc54016795)

[Marking 40](#_Toc54016796)

[All sections carry equal marks. 40](#_Toc54016797)

# Workbook 1

## Activity 1.1: Actual voltage across 5V breadboard pins.

Enter the Value you got here from Step 5.

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

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.

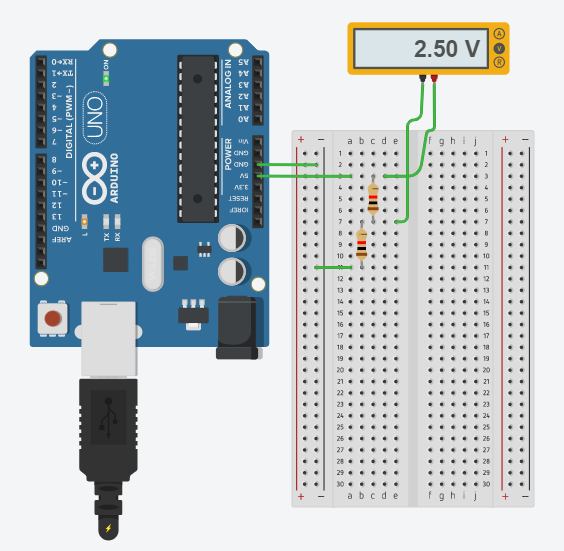
The values shown in TinkerCad are differ from values while using Arduino because TinkerCad is a simulation and does not use any of the voltage in the process. While its quite different when using the Arduino. The actual component of the Arduino will be using some of the voltage in the process thus the multimeter is not showing the full 5 volts.

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

Given that 4.84V is 3% less that the actual 5 volt, then it is saver to quote a tolerance of 5%. To allow tolerance greater than +/- 5% would effect the performance and accuracy of the system.

## Activity 1.3: Potential Divider Calculations

Show the working on how you achieved 2.5V

Showing the 2.5 V in TinkerCad with the same value resistors.

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

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

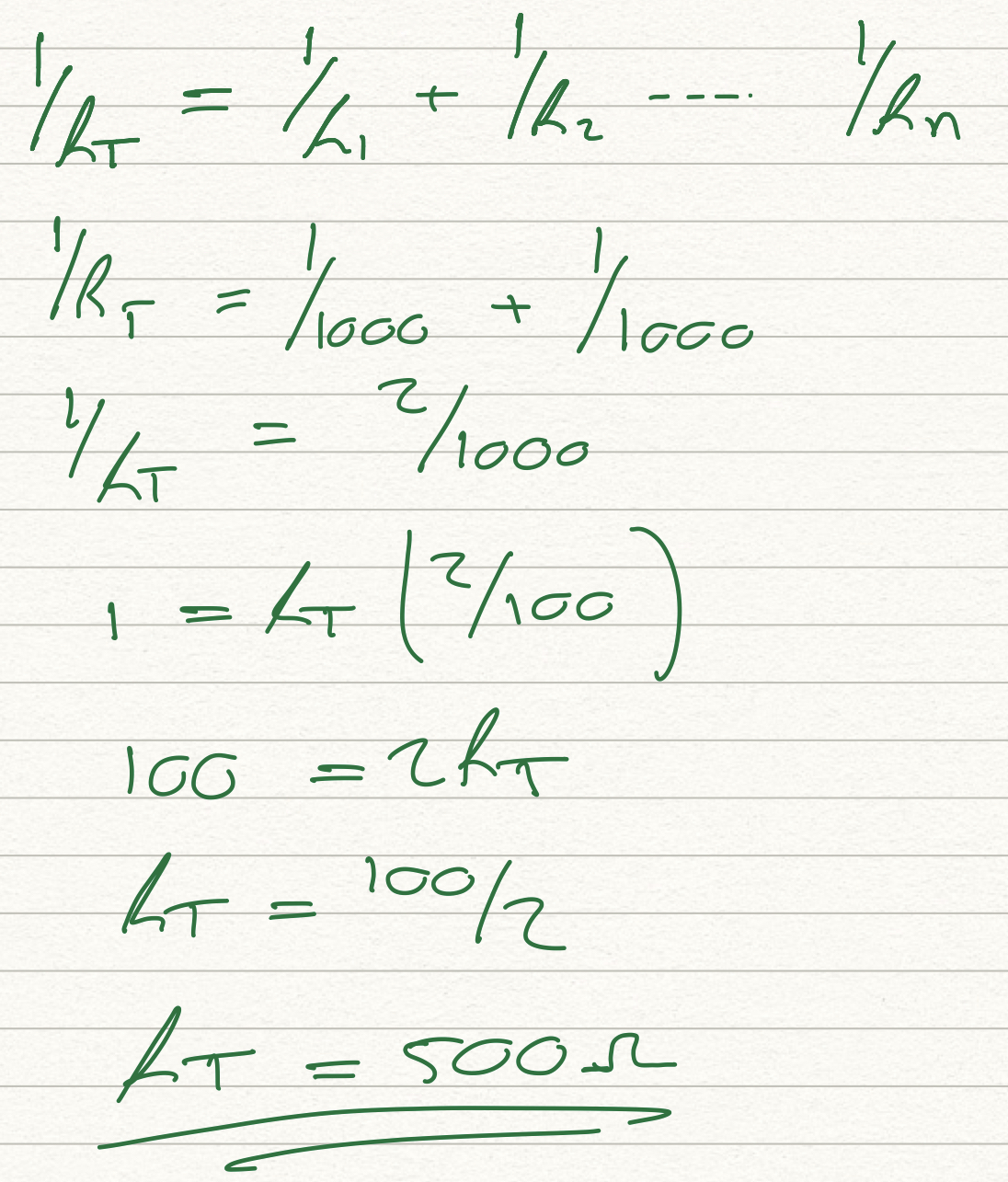
## Activity 1.6: LED Circuits

Each resistor Value

1 kohm

1 Kohm

Total resistance Calculation



Measured Resistance

500 ohm

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

The values are the same. If this got simulated and the connection is not the correct then you will get different reading. While measuring the resistance the resistors must not be connected to any other components or circuits.

## Activity 1.7: Current Measurement

Calculation of current flowing into LED

2.50 mA

Actual measured value of current

2.5 mA

Why might they be different?

If the wiring is not done correctly, positive and negative terminals, the display values going to be negative.

## Activity 1.8: Fritzing for 4 switches & LEDS

## Activity 1.9: Fritzing for Number 0-7

# Workbook 2

63 = 111111 in binary

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

63 as binary, including working

// Name: Mansoud Mansouri

// Student ID: 1916829

// choosing or setting the output pin as variable

// and also setting the delay time as variable.

int pinLive = 13;

int delaytime = 5000;

// the main setup

void setup()

{

pinMode(pinLive, OUTPUT);

}

// the infinite loop of blinking leds :D

void loop()

{

digitalWrite(pinLive, HIGH); // setting pin 13 high

delay(delaytime); // time for leds to stay on

digitalWrite(pinLive, LOW); // setting the pin 13 to low/ leds off

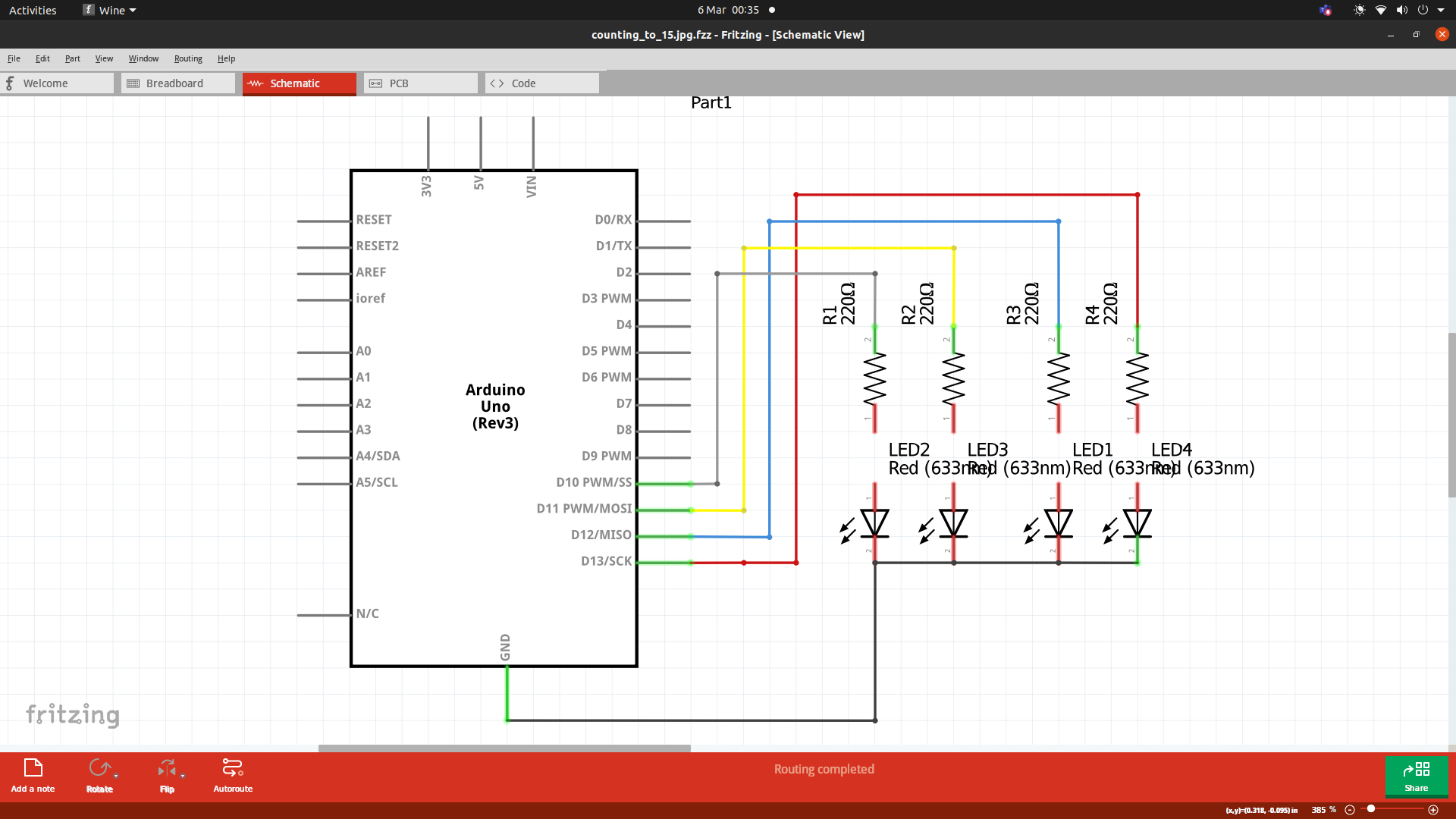
delay(delaytime); //time for leds to stay off

}

Copy & Post your code with a suitable comment at the top of code with your nam & student number 

## 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

// choosing the output pins and setting up global veriables

int ledOne = 13;

int ledTwo = 12;

int ledThree = 11;

int ledFour = 10;

int delayTime = 500;

int delayTimeEnd = 2000;

// SETTING UP THE MAIN

// setting up ledOne, ledTwo, ledThree and ledFour as output.

void setup()

{

pinMode(ledOne, OUTPUT);

pinMode(ledTwo, OUTPUT);

pinMode(ledThree, OUTPUT);

pinMode(ledFour, OUTPUT);

}

// the main loop of the program

// to count up to forteen and then start all over again!

// got nothing to do!

void loop()

{

// 0000

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, LOW);

delay(delayTime);

// 0001

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, LOW);

delay(delayTime);

// 0010

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, LOW);

delay(delayTime);

// 0011

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, LOW);

delay(delayTime);

// 0100

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, LOW);

// 0101

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, LOW);

delay(delayTime);

// 0110

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, LOW);

delay(delayTime);

// 0111

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, LOW);

delay(delayTime);

// 1000

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, HIGH);

delay(delayTime);

// 1001

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, HIGH);

delay(delayTime);

// 1010

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, HIGH);

delay(delayTime);

// 1011

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, LOW);

digitalWrite(ledFour, HIGH);

delay(delayTime);

// 1100

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, HIGH);

delay(delayTime);

// 1101

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, LOW);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, HIGH);

delay(delayTime);

// 1110

digitalWrite(ledOne, LOW);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, HIGH);

delay(delayTime);

// 1111

digitalWrite(ledOne, HIGH);

digitalWrite(ledTwo, HIGH);

digitalWrite(ledThree, HIGH);

digitalWrite(ledFour, HIGH);

delay(delayTimeEnd);

}

# Workbook 3

## Activity 3.1: Circuit Diagram of Button & LED

Fritzing

## Activity 3.2: 3 Switches & Led

Fritzing Circuit Diagram

Arduino Program

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

Fritzing

Arduino Program

# Workbook 4

## Activity 4.1: Serial Port

Fritzing

Arduino Program

Screen Shot of Serial Port

## Activity 4.2: Serial Port binary to decimal

Code

Screen Shot of Serial Port

## Activity 4.3: Calibrating Analogue Information

Code

Pot Resistance Clockwise

Pot Resistance Anti-clockwise

Sample of Values

Pot Resistance against Voltage change

|  |  |
| --- | --- |
| Pot Resitance | Voltage Measured |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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

Screen Shot of Serial Port

# Workbook 5

## Activity 5.1: RGB Led and switches

Fritzing

Arduino Program

## Activity 5.2: LED Matrix MAZE

Arduino Code

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

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 hand-written work is not acceptable.

### Marking

# All sections carry equal marks.