|  |  |  |
| --- | --- | --- |
| **Student Name** |  | **Student Number** |
| ALLAN FARAI MAKWEMBERE |  | 2108418 |

**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 2 4](#_Toc86301189)

[Activity 1.1: LED Flashing to show decimal number 63 as binary. 4](#_Toc86301190)

[Activity 1.2: 4 LED’s for counting up in binary from 0 to 15. 5](#_Toc86301191)

[Activity 1.3: Traffic Lights 7](#_Toc86301192)

[Workbook 3 9](#_Toc86301193)

[Activity 2.1: Circuit Diagram of Button & LED 9](#_Toc86301194)

[Activity 2.2: 3 Switches & Led 10](#_Toc86301195)

[Activity 2.3: 8 Buttons & LEDs (SWITCH STATEMENTS) 12](#_Toc86301196)

[Workbook 4 14](#_Toc86301197)

[Activity 3.1: Serial Port 14](#_Toc86301198)

[Activity 3.2: Serial Port binary to decimal 17](#_Toc86301199)

[Activity 3.3: Calibrating Analogue Information 19](#_Toc86301200)

[Activity 3.4: Temperature Sensor & Serial Port 22](#_Toc86301201)

[Workbook 5 24](#_Toc86301202)

[Activity 4.1: RGB Led and switches 24](#_Toc86301203)

[Activity 4.2: LED Matrix MAZE 26](#_Toc86301204)

[Activity 4.3: 1602 LCD Display 28](#_Toc86301205)

[Workbook 6 31](#_Toc86301206)

[Activity 5.1: PWM 31](#_Toc86301207)

[Workbook 7 33](#_Toc86301208)

[Activity 6.1: 2 Arduinos – using Digital Pins 33](#_Toc86301209)

[Activity 6.2: 2 Arduinos – using Serial I/O 35](#_Toc86301210)

[Workbook 8 37](#_Toc86301211)

[Activity 7.1: Stepper Motor 37](#_Toc86301212)

[Activity 7.2: 2 Stepper Motors 39](#_Toc86301213)

[Workbook 9 41](#_Toc86301214)

[Activity 8.1: Windscreen Wiper Code using Servos & Temperature Sensor 41](#_Toc86301215)

[Individual Project (50%) 42](#_Toc86301216)

[Rationale 42](#_Toc86301217)

[Timescales 42](#_Toc86301218)

[Equipment 42](#_Toc86301219)

[The Project 42](#_Toc86301220)

[Step 1 produce a detailed description of your project. 42](#_Toc86301221)

[Step 2 Circuit Diagram & Fritzing Schematic 42](#_Toc86301222)

[Step 3 A Program 43](#_Toc86301223)

[Step 4 Testing 43](#_Toc86301224)

[Step 5 Conclusions 43](#_Toc86301225)

[Layout 43](#_Toc86301226)

[Marking 43](#_Toc86301227)

[All sections carry equal marks. 43](#_Toc86301228)

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 2

## Activity 1.1: LED Flashing to show decimal number 63 as binary.

63 as binary, including working

Graphical user interface

Description automatically generated

(128 \* 0) + (64 \* 0) + (32 \* 1) + (16 \* 1) + (8 \* 1) + (4 \*1) +(2 \* 1) +(1 \*1) = 63

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

// ALLAN MAKWEMBERE 2108418

int led\_one = 13;

int led\_two = 12;

int led\_three = 11;

int led\_four = 10;

int led\_five = 9;

int led\_six = 8;

int led\_seven = 7;

int led\_eight = 6;

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

void setup() {

pinMode(13, OUTPUT);

pinMode(12, OUTPUT);

pinMode(11, OUTPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

pinMode(8, OUTPUT);

pinMode(7, OUTPUT);

pinMode(6, OUTPUT); }

void loop() {

digitalWrite(led\_one, LOW);

delay(1000);

digitalWrite(led\_two, LOW);

delay(1000);

digitalWrite(led\_three, HIGH);

delay(1000);

digitalWrite(led\_four, HIGH);

delay(1000);

digitalWrite(led\_five, HIGH);

delay(1000);

digitalWrite(led\_six, HIGH);

delay(1000);

digitalWrite(led\_seven, HIGH);

delay(1000);

digitalWrite(led\_eight, HIGH);

delay(5000);

for (int i = 0; i < sizeof(terminate); i++) {

digitalWrite(terminate[i],LOW);}

delay(5000);

}

## Activity 1.2: 4 LED’s for counting up in binary from 0 to 15.

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

Graphical user interface

Description automatically generated

Arduino Program for Step 4 i.e. 4 LEDs

int led\_one = 12;

int led\_two = 11;

int led\_three = 10;

int led\_four = 9;

int terminate[] = {12,11,10,9};

int binary[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};

void setup()

{

pinMode(led\_one, OUTPUT);

pinMode(led\_two, OUTPUT);

pinMode(led\_three, OUTPUT);

pinMode(led\_four, OUTPUT);

}

void loop()

{

for (int k = 0; k < sizeof(binary); k++){

for (int i = 0; i < sizeof(terminate); i++) {

if (binary[k] == 0) {

digitalWrite(terminate[i], LOW);

delay(150);}

if(binary[k] == 1) {

digitalWrite(led\_one, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if (binary[k] == 2) {

digitalWrite(led\_two, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 3) {

digitalWrite(led\_two, HIGH);

digitalWrite(led\_one, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if (binary[k] == 4) {

digitalWrite(led\_three, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 5) {

digitalWrite(led\_three, HIGH);

digitalWrite(led\_one, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 6) {

digitalWrite(led\_three, HIGH);

digitalWrite(led\_two, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if (binary[k] == 7){

digitalWrite(led\_three, HIGH);

digitalWrite(led\_two, HIGH);

digitalWrite(led\_one, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 8){

digitalWrite(led\_four, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 9){

digitalWrite(led\_four, HIGH);

digitalWrite(led\_one, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 10){

digitalWrite(led\_four, HIGH);

digitalWrite(led\_two,HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 11){

digitalWrite(led\_four, HIGH);

digitalWrite(led\_two,HIGH);

digitalWrite(led\_one, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 12){

digitalWrite(led\_four, HIGH);

digitalWrite(led\_three, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 13){

digitalWrite(led\_four, HIGH);

digitalWrite(led\_three, HIGH);

digitalWrite(led\_one, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 14){

digitalWrite(led\_four, HIGH);

digitalWrite(led\_three, HIGH);

digitalWrite(led\_two, HIGH);

digitalWrite(terminate[i],LOW);

delay(150);}

if(binary[k] == 15){

digitalWrite(led\_four, HIGH);

digitalWrite(led\_three, HIGH);

digitalWrite(led\_two, HIGH);

digitalWrite(led\_one, HIGH);

delay(150);

digitalWrite(terminate[i],LOW);}

}

}

}

## Activity 1.3: Traffic Lights

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

Graphical user interface, text

Description automatically generated

int amber = 13;

int red = 12;

int green = 11;

bool amber\_on = false;

void setup()

{

pinMode(amber, OUTPUT);

pinMode (red, OUTPUT);

pinMode (green, OUTPUT);

}

void loop()

{

do {

digitalWrite(red, HIGH);

delay(1000);

digitalWrite(amber, HIGH);

delay(1000);

digitalWrite(amber, LOW);

digitalWrite(red, LOW);

digitalWrite(green, HIGH);

delay(1000);

digitalWrite(green, LOW);

digitalWrite(amber, HIGH);

amber\_on = true;

delay(1000);

digitalWrite(amber, LOW);

}

while(amber\_on == true);

}

Arduino Program for Step 4 i.e. 4 LEDs

# Workbook 3

## Activity 2.1: Circuit Diagram of Button & LED

Fritzing

Graphical user interface

Description automatically generated with low confidence

## Activity 2.2: 3 Switches & Led

Fritzing Circuit Diagram

Graphical user interface

Description automatically generated

Arduino Program

const int REDLED = 13;

const int SWIT1 = 7;

const int WHITELED = 12;

const int SWIT2 = 4;

const int BLUELED = 11;

const int SWIT3 = 1;

void setup() {

pinMode(REDLED, OUTPUT);

pinMode(SWIT1, INPUT);

pinMode(WHITELED, OUTPUT);

pinMode(SWIT2, INPUT);

pinMode(BLUELED, OUTPUT);

pinMode(SWIT3, INPUT);

}

void loop() {

int switchstate = digitalRead(SWIT1);

if (switchstate == HIGH) {

digitalWrite(REDLED, HIGH);

while (switchstate == HIGH || REDLED == HIGH){

delay(1000);

digitalWrite(REDLED, LOW);

break;

}

}

else {

digitalWrite(REDLED, LOW);

}

int sw2 = digitalRead(SWIT2);

if (sw2 == HIGH) {

digitalWrite(WHITELED, HIGH);

while (sw2 == HIGH || WHITELED == HIGH){

delay(2000);

digitalWrite(WHITELED, LOW);

break;

}

}

else {

digitalWrite(WHITELED, LOW);

}

int sw3 = digitalRead(SWIT3);

if (sw3 == HIGH) {

digitalWrite(BLUELED, HIGH);

while (sw3 == HIGH || BLUELED == HIGH){

delay(3000);

digitalWrite(BLUELED, LOW);

break;

}

}

else {

digitalWrite(BLUELED, LOW);

}

}

## Activity 2.3: 8 Buttons & LEDs (SWITCH STATEMENTS)

Fritzing

Graphical user interface

Description automatically generated

Arduino Program

// C++ code

//

const int decimals[] = {128, 64, 32, 16, 8, 4, 2, 1};

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

int binary[8];

int value = 255;

//Enter value here

int solution[8];

void setup()

{

Serial.begin(9600);

pinMode(13, OUTPUT);

pinMode(12, OUTPUT);

pinMode(11, OUTPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

pinMode(8, OUTPUT);

pinMode(7, OUTPUT);

pinMode(6, OUTPUT);

}

void loop()

{

displayOutput(value);

delay(5000);

//ENTER VALUE HERE

}

int displayOutput(int value) {

for (int i = 0; i < sizeof(decimals); i++){

if (value >= decimals[i]){

value = value - decimals[i];

Serial.println(value);

binary[i] = 1;

if ( binary[i] == 1 && i == 0){

solution[i] = 13;

}

else if ( binary[i] == 1 && i == 1){

solution[i] = 12;

}

else if ( binary[i]== 1 && i == 2){

solution[i] = 11;

}

else if (binary[i] == 1 && i == 3){

solution[i] = 10;

}

else if (binary[i] == 1 && i == 4){

solution[i] = 9;

}

else if (binary[i] == 1 && i == 5) {

solution[i] = 8;

}

else if (binary[i] == 1 && i == 6) {

solution[i] = 7;

}

else if (binary[i] == 1 && i == 7) {

solution[i] = 6;

}

}

else {

binary[i] = 0;

}

delay(1000);

digitalWrite(solution[i], HIGH);

}

}

/\*

Wire 8 LED’s to the Arduino (these will represent an 8 bit binary number)

Write code so that you can pass a decimal value (0-255) to the Arduino which will light up the corresponding LED’s that will show the binary version.

Void displayOutput(int value) {

}

where “value” represents a decimal number corresponding to the binary combination

\*/

# Workbook 4

## Activity 3.1: Serial Port

Fritzing

Graphical user interface, application

Description automatically generated

Arduino Program

Graphical user interface, text

Description automatically generated

Screen Shot of Serial Port

Graphical user interface, text

Description automatically generated

## Activity 3.2: Serial Port binary to decimal

# Code

int button13;

int button12;

int button11;

int button10;

int button9;

int button7;

int button6;

int button5;

int binary[8];

int solution;

bool flag = false;

bool flag2 = false;

bool flag3 = false;

bool flag4 = false;

bool flag5 = false;

bool flag6 = false;

bool flag7 = false;

bool flag8 = false;

void setup()

{

Serial.begin(9600);

pinMode(13, INPUT);

pinMode(12, INPUT);

pinMode(11, INPUT);

pinMode(10, INPUT);

pinMode(9, INPUT);

pinMode(7, INPUT);

pinMode(6, INPUT);

pinMode(5, INPUT);

}

void loop(){

button13 = digitalRead(13);

button12 = digitalRead(12);

button11 = digitalRead(11);

button10 = digitalRead(10);

button9 = digitalRead(9);

button7 = digitalRead(7);

button6 = digitalRead(6);

button5 = digitalRead(5);

if (button13 == HIGH && flag == false) {

solution = solution + 128;

flag = true;

}

else if (button12 == HIGH && flag2 == false) {

solution = solution + 64;

flag2 = true;

}

else if (button11 == HIGH && flag3 == false) {

solution = solution + 32;

flag3 = true;

}

else if (button10 == HIGH && flag4 == false) {

solution = solution + 16;

flag4 = true;

}

else if (button9 == HIGH && flag5 == false) {

solution = solution + 8;

flag5 = true;

}

else if (button7 == HIGH && flag6 == false) {

solution = solution + 4;

flag6 = true;

}

else if (button6 == HIGH && flag7 == false) {

solution = solution + 2;

flag7 = true;

}

else if (button5 == HIGH && flag8 == false) {

solution = solution + 1;

flag8 = true;

}

Serial.println(solution);

}

Screen Shot of Serial Port

Graphical user interface, text

Description automatically generated

## Activity 3.3: 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 3.4: Temperature Sensor & Serial Port

int swit = 12;

int farhenheit = 0;

void setup()

{

Serial.begin(9600);

pinMode(swit, INPUT);

}

void loop()

{

farhenheit = digitalRead(swit);

float analogueReading = analogRead(0);

float degreesC;

float degreesF;

if (farhenheit == HIGH){

degreesC = (analogueReading \* 500) / 1024;

degreesF = (degreesC \* 1.8) + 32;

Serial.println(degreesF);

delay(100);}

if (farhenheit == LOW){

degreesC = (analogueReading \* 500) / 1024;

Serial.println(degreesC);

delay(100);}

}

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

Screen Shot of Serial Port

Text

Description automatically generated

# Workbook 5

## Activity 4.1: RGB Led and switches

Fritzing

Graphical user interface, application

Description automatically generated

Arduino Program

int button1;

int button2;

int button3;

void setup() {

pinMode(6, OUTPUT);

pinMode(8, OUTPUT);

pinMode(10, OUTPUT);

pinMode(5, INPUT);

pinMode(4, INPUT);

pinMode(3, INPUT);

}

void loop() {

button1 = digitalRead(5);

button2 = digitalRead(4);

button3 = digitalRead(3);

if (button1 == HIGH){

digitalWrite(6, HIGH);

delay(1000);

}

if (button1 == LOW) {

digitalWrite(6, LOW);

}

if (button2 == HIGH) {

digitalWrite(8, HIGH);

delay(1000);

}

if (button2 == LOW) {

digitalWrite(8, LOW);

}

if (button3 == HIGH) {

digitalWrite(10, HIGH);

delay(1000);

}

if (button3 == LOW){

digitalWrite(10, LOW);

}

}

## Activity 4.2: LED Matrix MAZE

# Arduino Code

#include <Adafruit\_NeoPixel.h>

#define PIN 2 // input pin Neopixel is attached to

#define NUMPIXELS 24 // number of neopixels in strip

Adafruit\_NeoPixel pixels = Adafruit\_NeoPixel(NUMPIXELS, PIN, NEO\_GRB + NEO\_KHZ800);

int delayval = 100; // timing delay in milliseconds

int redColor = 0;

int greenColor = 0;

int blueColor = 0;

void setup() {

// Initialize the NeoPixel library.

pixels.begin();

}

void loop() {

setColor();

for (int i=0; i < NUMPIXELS; i++) {

// pixels.Color takes RGB values, from 0,0,0 up to 255,255,255

pixels.setPixelColor(i, pixels.Color(redColor, greenColor, blueColor));

// This sends the updated pixel color to the hardware.

pixels.show();

// Delay for a period of time (in milliseconds).

delay(delayval);

}

}

// setColor()

// picks random values to set for RGB

void setColor(){

redColor = random(0, 255);

greenColor = random(0,255);

blueColor = random(0, 255);

}

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 ☺

Graphical user interface, application

Description automatically generated

## Activity 4.3: 1602 LCD Display

Fritzing

Graphical user interface, application

Description automatically generated

Arduino Program

#include <LiquidCrystal.h>

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

//object and assign the pins

int maxC=0, minC=100, maxF=0, minF=212;

int scale = 1;

int buttonPin=8;

void setup() {

lcd.begin(16, 2); // Set the display

//to 16 columns and 2 rows

analogReference(INTERNAL);

pinMode(buttonPin, INPUT);

lcd.clear();

}

void loop() {

lcd.setCursor(0,0); // set cursor to

//home position

int sensor = analogRead(0); // read the temp

//from sensor

int buttonState = digitalRead(buttonPin); // check for button

//press

switch (buttonState) { // change scale

//state if pressed

case HIGH:

scale=-scale; // invert scale

lcd.clear();

}

switch (scale) { // decide if C or F

scale;

case 1:

celsius(sensor);

break;

case -1:

fahrenheit(sensor);

}

delay(250);

}

void celsius(int sensor) {

lcd.setCursor(0,0);

int temp = sensor \* 0.1074188; // convert to C

lcd.print(temp);

lcd.write(B11011111); // degree symbol

lcd.print("AFM2022");

if (temp>maxC) {maxC=temp;}

if (temp<minC) {minC=temp;}

lcd.setCursor(0,1);

lcd.print("H=");

lcd.print(maxC);

lcd.write(B11011111);

lcd.print("C L=");

lcd.print(minC);

lcd.write(B11011111);

lcd.print("C ");

}

void fahrenheit(int sensor) {

lcd.setCursor(0,0);

float temp = ((sensor \* 0.1074188) \* 1.8)+32; // convert to F

lcd.print(int(temp));

lcd.write(B11011111); // print degree

//symbol

lcd.print("F ");

if (temp>maxF) {maxF=temp;}

if (temp<minF) {minF=temp;}

lcd.setCursor(0,1);

lcd.print("H=");

lcd.print(maxF);

lcd.write(B11011111);

lcd.print("F L=");

lcd.print(minF);

lcd.write(B11011111);

lcd.print("F ");

}

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

Graphical user interface, application

Description automatically generated

# Workbook 6

## Activity 5.1: PWM

Fritzing

Graphical user interface

Description automatically generated

Arduino Program

#define pwrout 11

int brightness = 0;

int fadeAmount = 5;

int potpin = A0;

int potval = 0;

void setup(){

pinMode(pwrout, OUTPUT);

Serial.begin(9600);

}

void loop(){

potval = analogRead(potpin);

Serial.println(potval);

analogWrite(pwrout, brightness);

brightness = brightness + fadeAmount;

if (brightness == 0 || brightness == 255){

fadeAmount = -fadeAmount;

}

delay(100);

}

# Workbook 7

## Activity 6.1: 2 Arduinos – using Digital Pins

Fritzing

Graphical user interface, diagram

Description automatically generated

Arduino Program

# Temperature Arduino Code

#include <LiquidCrystal.h>

// C++ code

//

int Testdata[10];

void setup()

{

Serial.begin(9600);

}

void loop()

{

float analogueReading = analogRead(0);

float degreesC;

degreesC = (analogueReading \* 500) / 1024;

Testdata[10] = degreesC;

Serial.write(Testdata, 4);

Serial.println(degreesC);

delay(100);

}

# LCD program

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

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

char Testdata[10];

void setup() {

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

Serial.begin(9600);

}

void loop() {

Serial.readBytes(Testdata,5);

Serial.println(Testdata);

if (Serial.available()){

Testdata = Serial.read();

lcd.print(Testdata);

}

}

## Activity 6.2: 2 Arduinos – using Serial I/O

Fritzing

Arduino Program

# Workbook 8

## Activity 7.1: Stepper Motor

Fritzing

Graphical user interface, diagram

Description automatically generated

# Arduino Program

#include <Stepper.h>

// C++ code

const int StepPerRevolution = 120;

Stepper mystepper(StepPerRevolution, 8, 9, 10, 11);

int stepcount = 0;

void setup()

{

Serial.begin(9600);

}

void loop()

{

int sensorReading = analogRead(A0);

int motorspeed = map(sensorReading,0, 1023, 0, 250);

delay(1000);

if (motorspeed > 0){

mystepper.setSpeed(motorspeed);

mystepper.step(StepPerRevolution/100);

Serial.println(sensorReading);

delay(1000);

mystepper.step(-StepPerRevolution);

//Counter-clockwise

Serial.println(sensorReading);

}

}

## Activity 7.2: 2 Stepper Motors

Fritzing

A picture containing text

Description automatically generated

# Arduino Program

#### Motor 1

#include <Stepper.h>

#include <SoftwareSerial.h>

// C++ code

const byte rxPin = 2;

const byte txPin = 3;

SoftwareSerial mySerial (rxPin, txPin);

const int StepPerRevolution = 120;

Stepper mystepper(StepPerRevolution, 8, 9, 10, 11);

int stepcount = 0;

int button1;

int button2;

int button3;

int button4;

int transmission[2];

void setup()

{

Serial.begin(9600);

pinMode(rxPin, INPUT);

pinMode(txPin, OUTPUT);

pinMode(7, INPUT);

pinMode(6, INPUT);

pinMode(5, INPUT);

pinMode(4, INPUT);

}

void loop()

{

button1 = digitalRead(7);

button2 = digitalRead(6);

button3 = digitalRead(5);

button4 = digitalRead(4);

int sensorReading = analogRead(A0);

int motorspeed = map(sensorReading,0, 1023, 0, 250);

delay(1000);

if (motorspeed > 0){

if (button1 = HIGH){

transmission = (0, 0);

mystepper.setSpeed(motorspeed);

mystepper.step(StepPerRevolution/100);

Serial.println(sensorReading);

}else if (button2 = HIGH){

transmission = (0, 1);

mystepper.step(-StepPerRevolution);

//Counter-clockwise

Serial.println(sensorReading);

}else if (button3 = HIGH) {

transmission = (1, 0);

mystepper.step(-StepPerRevolution);

}else if(button4 = HIGH) {

transmission = (1, 1);

mystepper.step(StepPerRevolution);

}

}

}

/\*The motors go forward with one button together

2) The motors revers with a second button click

3) Motor 1 goes forward, motor 2 reverses on a 3rd button

Motor 1 reverses, motor 2 goes forward on a 4th button

\*/

#### Motor2

#include <Stepper.h>

// C++ code

const int StepPerRevolution = 120;

Stepper mystepper(StepPerRevolution, 8, 9, 10, 11);

int stepcount = 0;

int button1;

int button2;

int button3;

int button4;

void setup()

{

Serial.begin(9600);

pinMode(7, INPUT);

pinMode(6, INPUT);

pinMode(5, INPUT);

pinMode(4, INPUT);

}

void loop()

{

button1 = digitalRead(7);

button2 = digitalRead(6);

button3 = digitalRead(5);

button4 = digitalRead(4);

int sensorReading = analogRead(A0);

int motorspeed = map(sensorReading,0, 1023, 0, 250);

delay(1000);

if (motorspeed > 0){

if (button1 = HIGH){

mystepper.setSpeed(motorspeed);

mystepper.step(StepPerRevolution/100);

Serial.println(sensorReading);

}else if (button2 = HIGH){

mystepper.step(-StepPerRevolution);

//Counter-clockwise

Serial.println(sensorReading);

}else if (button3 = HIGH) {

mystepper.step(StepPerRevolution);

}else if(button4 = HIGH) {

mystepper.step(-StepPerRevolution);

}

}

}

# Workbook 9

## Activity 8.1: Windscreen Wiper Code using Servos & Temperature Sensor

Graphical user interface

Description automatically generated

#### Arduino Code

// C++ code

//

#include <Servo.h>

int pos = 0;

Servo servo\_9;

Servo servo\_8;

int temp\_count = 0;

void setup()

{

Serial.begin(9600);

servo\_9.attach(9, 500, 2500);

servo\_8.attach(8, 500, 2500);

}

void loop(){

float analogueReading = analogRead(0);

float degreesC;

degreesC = (analogueReading \* 500) / 1024;

Serial.println(degreesC);

while (degreesC > 65){

for (pos = 0; pos <= 180; pos += 1) {

servo\_9.write(pos);

servo\_8.write(pos);

delay(15);

}

for (pos = 0; pos <= 180; pos += 1) {

servo\_9.write(pos);

servo\_8.write(pos);

delay(15);

}

temp\_count+=1;

degreesC = degreesC - temp\_count;

Serial.println(degreesC);

if (degreesC <= 65) {

delay(10000);

}

if (temp\_count = 8){

temp\_count-=8;}

}

}

/\*

using the temperature sensor, have the servos behave like windscreen wipers on a car

when a threshold temperature is reached.

 Turn off the wipers once the threshold temperature falls back down.\*/

# 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 hand-written work is not acceptable.

### Marking

# All sections carry equal marks.