|  |  |  |
| --- | --- | --- |
| **Student Name** |  | **Student Number** |
| Akriti Kumari Dev | 23329584 |

**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 ................................................. 5

Activity 1.5: Voltage Divider circuit readings from Breadboard circuit. ................................................ 6

Activity 1.6: LED Circuits ..................................................................................................................... 6

Activity 1.7: Current Measurement ..................................................................................................... 7

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

Activity 2.3: Traffic Lights ................................................................................................................... 18

Workbook 3 .......................................................................................................................................... 21

Activity 3.1: Circuit Diagram of Button & LED ..................................................................................... 21

Activity 3.2: 3 Switches & Led ............................................................................................................ 23

Activity 3.3: 8 Buttons & LEDs (SWITCH STATEMENTS) ....................................................................... 26

Workbook 4 .......................................................................................................................................... 29

Activity 4.1: Serial Port....................................................................................................................... 29

Activity 4.2: Serial Port binary to decimal ........................................................................................... 32

Activity 4.3: Calibrating Analogue Information ................................................................................... 36

Activity 4.4: Temperature Sensor & Serial Port .................................................................................. 39

Workbook 5 .......................................................................................................................................... 42

Activity 5.1: RGB Led and switches ..................................................................................................... 42

Activity 5.2: Distance Sensor .............................................................................................................. 45

Activity 5.3: 1602 LCD Display ............................................................................................................ 48

Workbook 6 .......................................................................................................................................... 53

Activity 6.1: PWM .............................................................................................................................. 53 Workbook 7 .......................................................................................................................................... 55

Activity 7.1: 2 Arduinos – using Digital Pins ........................................................................................ 55

Activity 7.2: 2 Arduinos – using Serial I/O ........................................................................................... 57

Workbook 8 .......................................................................................................................................... 59

Activity 8.1: Stepper Motor Circuit Diagram ....................................................................................... 59

Activity 8.2: 2 Stepper Motors ........................................................................................................... 61

Workbook 9 .......................................................................................................................................... 62

Activity 9.1: Windscreen Wiper Code using Servos & Temperature Sensor ......................................... 62

Individual Project (50%) ......................................................................................................................... 64

Rationale ........................................................................................................................................... 64

Timescales ......................................................................................................................................... 64

Equipment ......................................................................................................................................... 64

The Project ........................................................................................................................................ 64

Step 1 produce adetailed description of your project. .................................................................... 64

Step 2 Circuit Diagram&Fritzing Schematic .................................................................................... 64

Step 3 A Program ........................................................................................................................... 65

Step 4 Testing ................................................................................................................................ 65

Step 5 Conclusions ......................................................................................................................... 65

Layout ........................................................................................................................................... 65

Marking ......................................................................................................................................... 65

All sections carry equal marks. ............................................................................................................... 65

|  |  |
| --- | --- |
| 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.

5.2V

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

3.3V

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.

The value may change based on variables like as internal resistance, temperature, and tolerance

(occurring during the manufacturing process). As a result, tolerance can be found in the physical form. For our convenience, we take use of excellent situations. When conditions are perfect, TinkerCad works. As a result, in TinkerCad we frequently get ideal values like 5V in the example above. In reality, we frequently get results that are either little higher or slightly lower than

optimal, like 5.2.

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

Tolerance = (L-R)/L\*100

Tolerance = =(4.84 − 5)/5 × 100%

= (−0.16)/5 × 100%

= −3.2%

From calculation, we get the value of tolerance to be 3.2%. So the sensible tolerance to quote is +- 3.2%.

Both resistors are connected in series that’s why voltage is divided..

## Activity 1.3: Potential Divider Calculations

Show the working on how you achieved 2.5V

|  |
| --- |
| Starting out, we apply a 5V voltage. To create a 2.5V voltage drop, we use a potential divider. We serially connect the resistors of the two 220-ohm resistors to measure the voltage between them on either side of the battery. First, we must determine the overall resistance, which is R= 220 + 220 ohm = 440 ohms.  V=(resistance/total resistance)\*voltage  V= (220/440) \*5  NOW,  V= (1/2)\*5  V= 2.5V |

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

Here, R1=220Ω Rt=R1+R2 (Total Resistance)

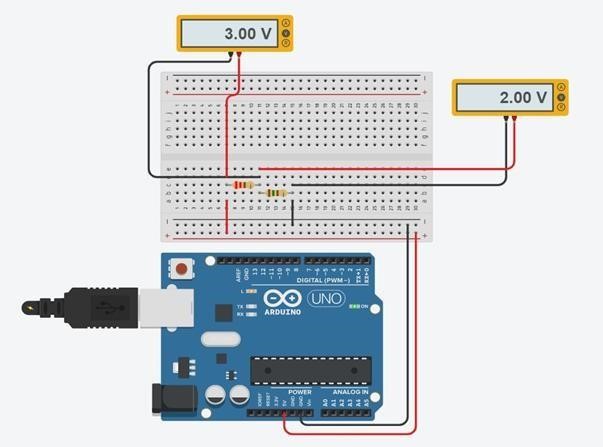
V= 5V (Taking 5V)

Vd=Voltage drop=3V

After Calculation we get,

220+R2= 366.67Ω [here 366.67-220=146.67 The value of second resistance(R2) is 147Ω.

## 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

Here,

1/R=1/R1+1/R2

=1/220+1/220

=2/220

=1/110

Measured Resistance

110 ohm

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

Due to device tolerance, the measured resistance displayed on multimeters and electrical equipment may not reflect the precise resistance of the object being tested. The allowed range of departure from the actual value is what is meant by tolerance. Additionally, the resistance of the wires used to link the measurement device to the multimeter or other measuring tools may have an impact on the measured resistance. Therefore, there is a chance that the measured resistance will differ from the object's actual resistance.

## Activity 1.7: Current Measurement

Calculation of current flowing into LED

61.5 A

Actual measured value of current

60 A

Why might they be different?

In real-world situations, it's possible that the estimated or ideal value of a measurement used for computations won't match the actual value. This is because a number of factors can change the measurement value, such as tolerance, which is brought on by the substance being measured's natural fluctuation. The measurement value may also be affected in various ways by additional factors like time, temperature, and humidity.

## Activity 1.8: Fritzing for 4 switches & LEDS

A picture containing text, diagram, line, plot

Description automatically generated

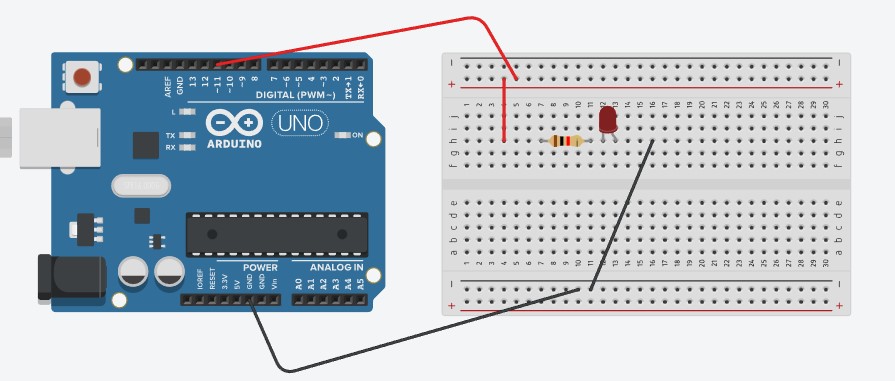
# Activity 1.9: Fritzing for Number 0-7

A picture containing text, diagram, line, parallel

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 

|  |
| --- |
| // C++ code  // Akriti Kumari Dev // Student Id- 2329584  int i = 0; int counter; void setup()  {  pinMode(11, OUTPUT);  }  void loop(){  //binary for 63 is 111111 the led blinks for 6 times and stops itself for (counter = 0; counter < 6; ++counter) { digitalWrite(11, HIGH); delay(1000); // Wait for 1000 millisecond(s) digitalWrite(11, LOW); delay(1000); // Wait for 1000 millisecond(s)  exit(0)  }  } |

## 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**

A picture containing text, diagram, plan, schematic

Description automatically generated

**Arduino Program for Step 4 i.e., 4 LEDs**

// C++ code

// Akriti Kumari Dev

// Student ID - 2329584

// this code will show the binary count from 0-15 serially.

int led1=12; int led2=10; int led3=8; int led4=7;

void setup()

{

pinMode(led1, OUTPUT); pinMode(led2, OUTPUT); pinMode(led3, OUTPUT); pinMode(led4, OUTPUT);

}

void loop()

{

//for binary 0 digitalWrite(led1,LOW); digitalWrite(led2,LOW); digitalWrite(led3,LOW); digitalWrite(led4,LOW); delay(1000);

//for binary 1

digitalWrite(led1,LOW); digitalWrite(led2,LOW); digitalWrite(led3,LOW); digitalWrite(led4,HIGH); delay(1000);

//for binary 2 digitalWrite(led1,LOW); digitalWrite(led2,LOW); digitalWrite(led3,HIGH); digitalWrite(led4,LOW); delay(1000);

//for binary 3 digitalWrite(led1,LOW); digitalWrite(led2,LOW); digitalWrite(led3,HIGH); digitalWrite(led4,HIGH); delay(1000);

//for binary 4 digitalWrite(led1,LOW); digitalWrite(led2,HIGH); digitalWrite(led3,LOW); digitalWrite(led4,LOW); delay(1000);

//for binary 5 digitalWrite(led2,HIGH); digitalWrite(led3,LOW); digitalWrite(led4,HIGH); delay(1000);

//for binary 6 digitalWrite(led1,LOW); digitalWrite(led2,HIGH); digitalWrite(led3,HIGH); digitalWrite(led4,LOW); delay(1000);

//for binary 7 digitalWrite(led1,LOW); digitalWrite(led2,HIGH); digitalWrite(led3,HIGH); digitalWrite(led4,HIGH); delay(1000);

//for binary 8 digitalWrite(led1,HIGH); digitalWrite(led2,LOW); digitalWrite(led3,LOW); digitalWrite(led4,LOW); delay(1000);

//for binary 9 digitalWrite(led1,HIGH); digitalWrite(led3,LOW); digitalWrite(led4,HIGH); delay(1000);

//for binary 10 digitalWrite(led1,HIGH); digitalWrite(led2,LOW); digitalWrite(led3,HIGH); digitalWrite(led4,LOW); delay(1000);

//for binary 11 digitalWrite(led1,HIGH); digitalWrite(led2,LOW); digitalWrite(led3,HIGH); digitalWrite(led4,HIGH); delay(1000);

//for binary 12 digitalWrite(led1,HIGH); digitalWrite(led2,HIGH); digitalWrite(led3,LOW); digitalWrite(led4,LOW); delay(1000);

//for binary 13 digitalWrite(led1,HIGH); digitalWrite(led2,HIGH);

digitalWrite(led4,HIGH); delay(1000);

//for binary 14 digitalWrite(led1,HIGH); digitalWrite(led2,HIGH); digitalWrite(led3,HIGH); digitalWrite(led4,LOW); delay(1000);

//for binary 15 digitalWrite(led1,HIGH); digitalWrite(led2,HIGH); digitalWrite(led3,HIGH); digitalWrite(led4,HIGH); delay(1000);

}

### Activity 2.3: Traffic Lights

**Fritzing Circuit diagram for traffic light**

A picture containing text, diagram, line, font

Description automatically generated

**Arduino Program for traffic light**

//Akriti Kumari Dev //2329584

int redLight = 8; int yellowLight = 10; int greenLight = 12;

void setup() {

pinMode(redLight, OUTPUT); pinMode(yellowLight, OUTPUT); pinMode(greenLight, OUTPUT);

}

void loop() { digitalWrite(redLight, HIGH); delay(5000);

digitalWrite(yellowLight, HIGH); delay(2000);

digitalWrite(redLight, LOW); digitalWrite(yellowLight, LOW); digitalWrite(greenLight, HIGH); delay(5000);

digitalWrite(greenLight, LOW); digitalWrite(yellowLight, HIGH); delay(2000);

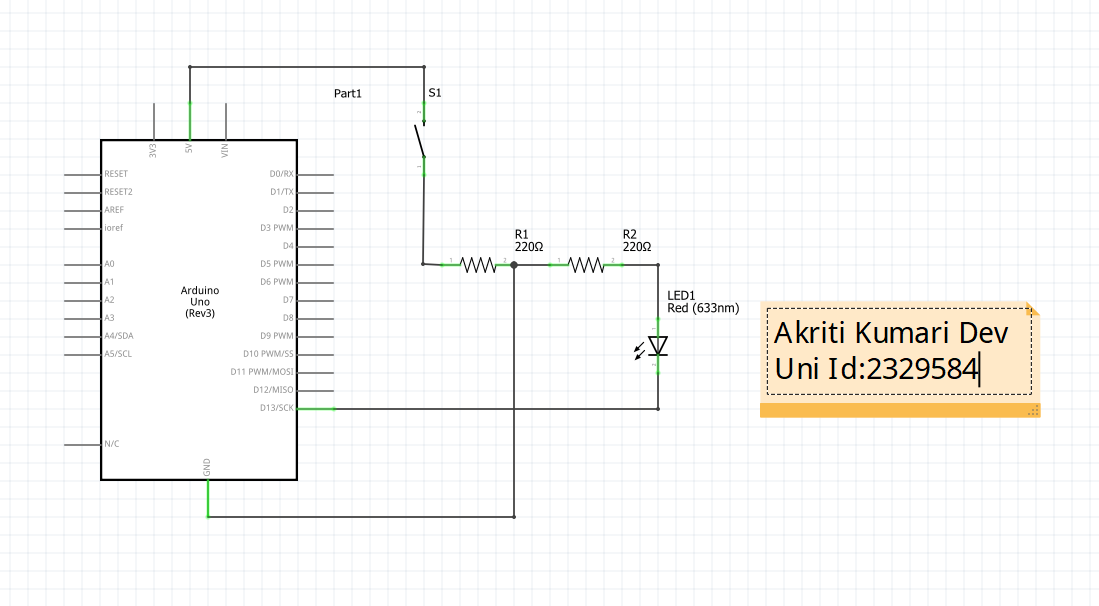
digitalWrite(2000, LOW);

}

Workbook 3

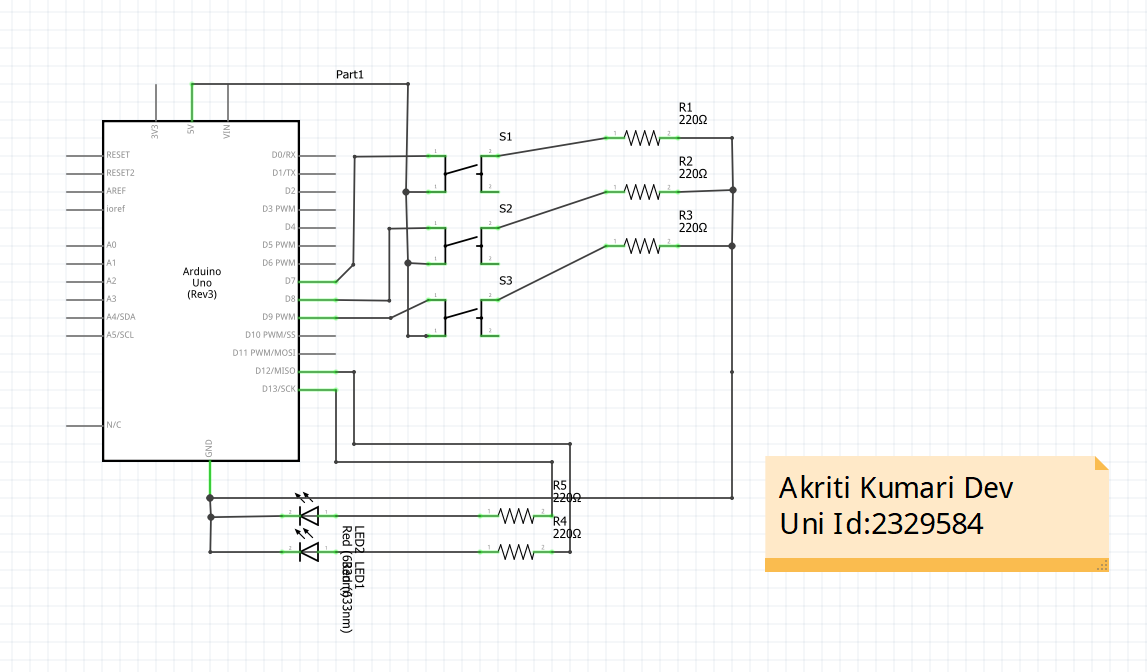
### Activity 3.1: Circuit Diagram of Button & LED

**Fritzing Diagram**



### Activity 3.2: 3 Switches & Led

**Fritzing Diagram**



**Arduino Program**

//Akriti Kumari Dev

//2329584

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**

A picture containing text, diagram, line, parallel

Description automatically generated

**Arduino Program**

//Akriti Kumari Dev

//2329584

// C++ code

//

int led[8]={6,7,8,9,10,11,12,13}; int value=255; void setup()

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

{

pinMode(led[i], OUTPUT);

}

}

void loop()

{

displayOutput(value);

}

void displayOutput(int value) {

for (int i = 0; i < 8; i++) { if (bitRead(value, i) == 1) { digitalWrite(led[i], HIGH); } else {

digitalWrite(led[i], LOW);

}

}

}

Workbook 4

### Activity 4.1: Serial Port **Fritzing Diagram**

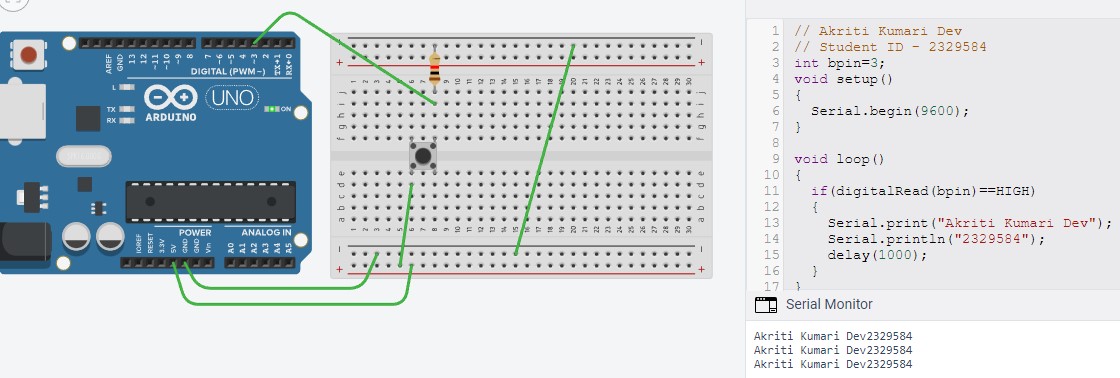
### 

**Arduino Program**

|  |
| --- |
| //Akriti Kumari Dev  //2329584  // C++ code  // int bpin=13; void setup()  {  Serial.begin(9600);  }    void loop()  {  if(digitalRead(bpin)==HIGH)  {  Serial.print("Akriti Kumari Dev "); Serial.println("2329584"); delay(1000);  }  } |

Screen Shot of Serial

Port



### Activity 4.2: Serial Port binary to decimal

**Arduino Code**

//Akriti Kumari Dev

//2329584

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

int i;

void setup()

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

{

pinMode(switchInput[i], INPUT);

}

Serial.begin(9600);

}

void loop()

{ int n; float result = 0; for (i = 0; i < 8; i++)

{

int switchState = digitalRead(switchInput[i]); if (switchState == HIGH)

{ n = 1;

}

else

{ n = 0; }

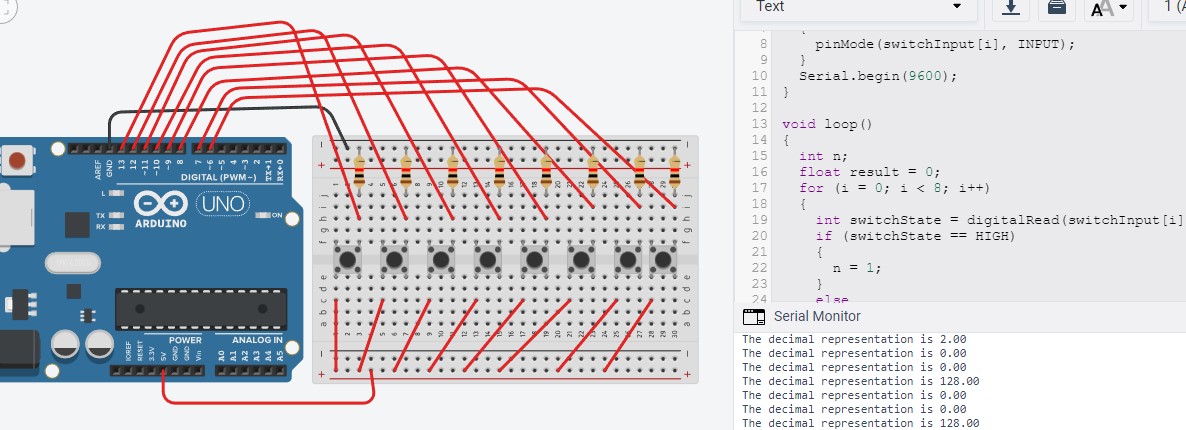
result = result + n \* pow(2, i);

}

Serial.print("The decimal representation is "); Serial.println(result); delay(1000);

}

Screen Shot of Serial Port



### Activity 4.3: Calibrating Analogue Information

**Arduino Code**

//Akriti Kumari Dev

//2329584

// C++ code

// int buttonPin=11; void setup()

{

pinMode(buttonPin,INPUT);

Serial.begin(9600);

}

void loop()

{

int buttonValue=digitalRead(buttonPin); if(buttonValue==HIGH)

{

int value=analogRead(A0);

Serial.println(value);

float voltage=value\*(5.0/1023.0);

Serial.print("The corresponding voltage is "); Serial.println(voltage); delay(1000); float resistance=voltage\*250.0/5.0;

Serial.print("The corresponding resistance is "); Serial.println(resistance); delay(1000);

}

}

Pot Resistance Clockwise

0 K ohm

Pot Resistance Anti-clockwise

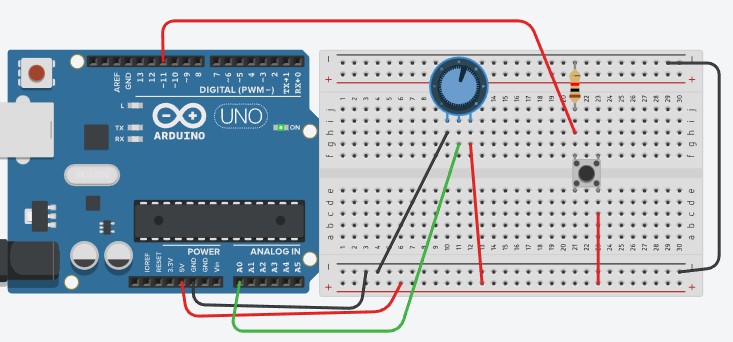
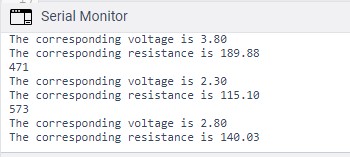
249.80 K ohm

Sample of Values

Pot Resistance against Voltage change

|  |  |
| --- | --- |
| Pot Resitance | Voltage Measured |
| 138.56 K ohm | 2.77 V |
| 96.77 K ohm | 1.94 V |
| 170.33 K ohm | 3.41 V |
| 114.61 K ohm | 2.29 V |
| 163.49 K ohm | 3.27 V |

**Screen Shot of Meaningful Serial Port Output, not just numbers.**

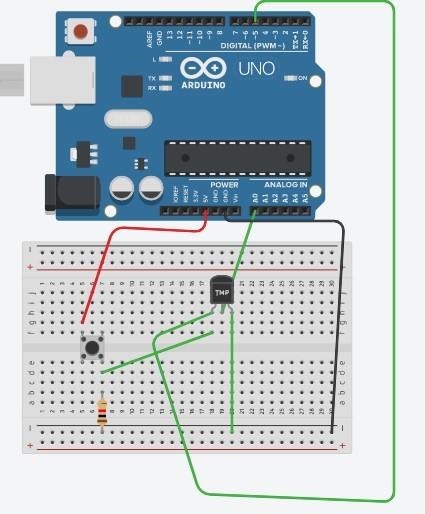


### Activity 4.4: Temperature Sensor & Serial Port

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

|  |
| --- |
| //Akriti Kumari Dev  //2329584 void setup()  {  pinMode(A0, INPUT);  Serial.begin(9600);  }  void loop()  {  float ana = analogRead(A0); float degreeC = ana\*500/1023; float degreeF = (degreeC \* 1.8)+32;  Serial.print("The temperature is ");  Serial.print(degreeF);  Serial.println(" degree Fahrenheit."); delay(5000);  } |

**Screen Shot of Serial Port**



# Workbook 5

### Activity 5.1: RGB Led and switches.

**Fritzing Diagram**

**A diagram of a circuit

Description automatically generated with low confidence**

**Arduino Program**

//Akriti Kumari Dev

//2329584

// C++ code

// int rled= 10; int bled= 11; int gled= 12; 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, LOW);

}

else

{

digitalWrite(rled, HIGH);

}

if (digitalRead(bpin)==HIGH) {

digitalWrite(bled, LOW);

} else

{

digitalWrite(bled, HIGH);

}

if (digitalRead(gpin)==HIGH)

{

digitalWrite(gled, LOW);

} else

{

digitalWrite(gled, HIGH);

}

}

### Activity 5.2: Distance Sensor Arduino Code

//Akriti Kumari Dev

// Student Id- 2329584 // C++ code int echoPin=7; int triggerPin=11; void setup()

{

pinMode(triggerPin, OUTPUT); pinMode(echoPin, INPUT); Serial.begin(9600);

}

void loop()

{

int highPulseDuration; int calculateDistanceCm;

//set the triggerpoint to low before setting digitalWrite(triggerPin, LOW); delayMicroseconds(5);

//create 10 seconds pulse on the trig pin digitalWrite(triggerPin, HIGH); delayMicroseconds(10);

//set the pin to low to end the pulse digitalWrite(triggerPin, LOW);

//read the duration of the highpulse on the echo pin highPulseDuration=pulseIn(echoPin,HIGH);

//calculating the distance calculateDistanceCm = highPulseDuration\*0.034/2;

//sound wave divided by 2 (go and back)

//Displays the distance on the serialpulseIn Monitor

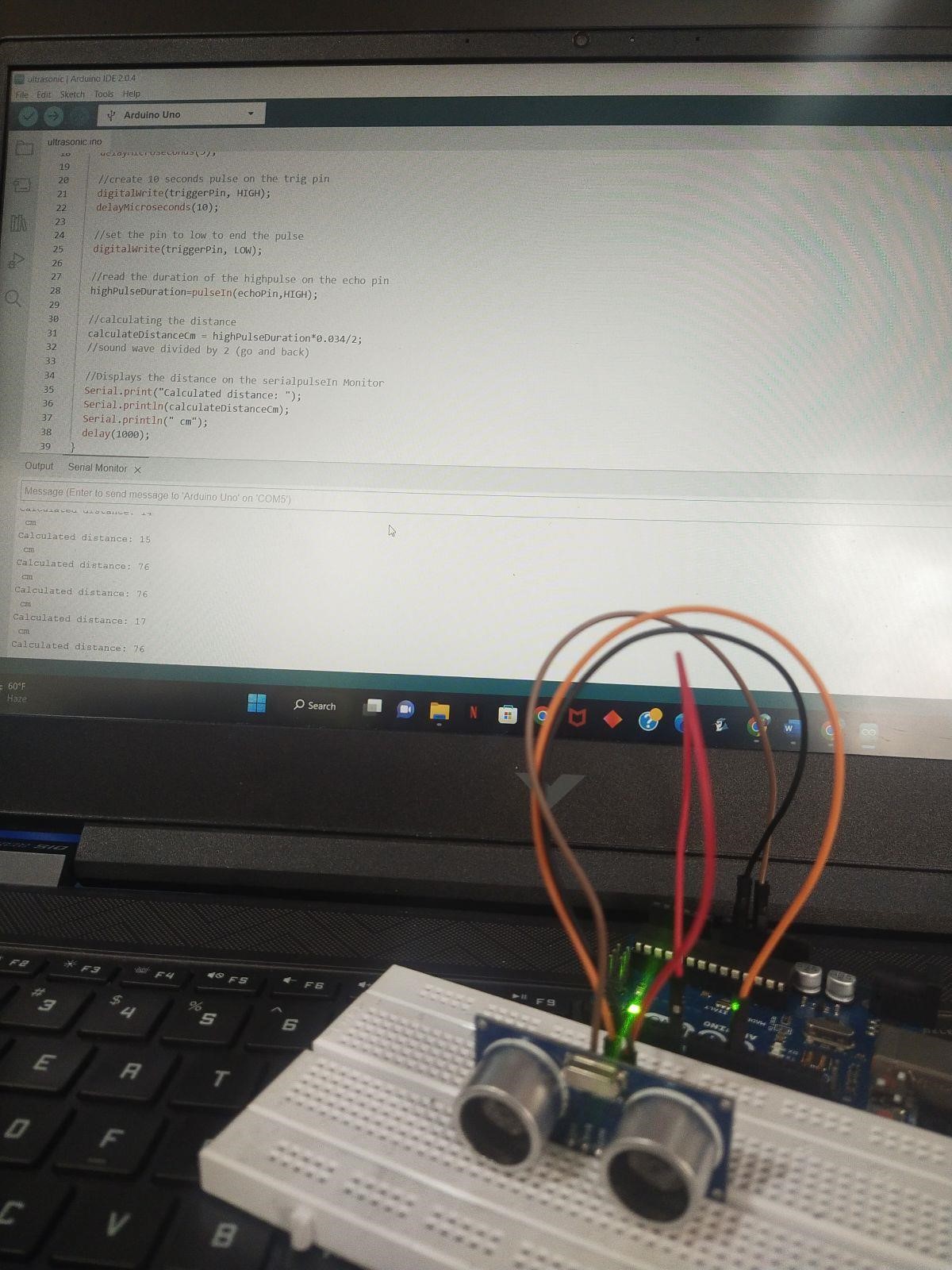
Serial.print("Calculated distance: ");

Serial.println(calculateDistanceCm);

Serial.println(" cm"); delay(1000);

}

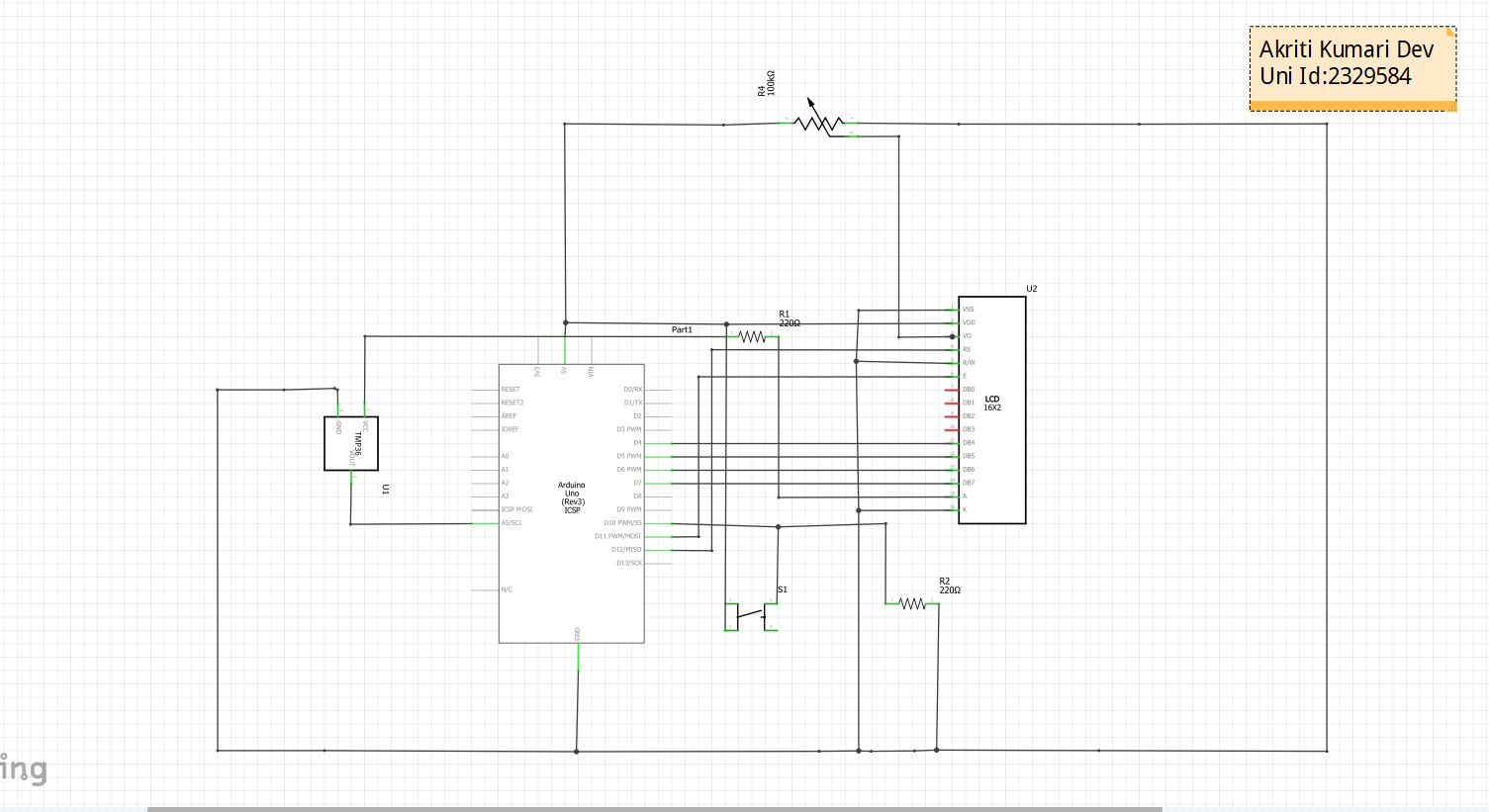
Take a picture of your distance sensor and include it here, please reduce the size and quality as it will be



too large else 

### Activity 5.3: 1602 LCD Display

**Fritzing Diagram**



**Arduino Program**

// C++ code

// Akriti Kumari Dev

// Student ID - 2329584

// include the library code:

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2); float value; int pushSwitch = 0; void setup() { pinMode(8,INPUT);

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

lcd.begin(16, 2);

}

void loop() { value=analogRead(A0); pushSwitch=digitalRead(8); float degC = (value/1024.0)\*500; float degF = (degC\*1.8+32); if(pushSwitch == HIGH){

lcd.clear(); lcd.setCursor(0,0); lcd.print(degC); lcd.print(" C "); lcd.setCursor(0,1); lcd.print("NB\_21"); delay(1000);

}

else{

lcd.clear(); lcd.setCursor(0,0); lcd.print(degF); lcd.print(" F "); lcd.print("NB\_21"); 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**

|  |
| --- |
|  |

|  |
| --- |
| //Akriti Kumari Dev  //2329584  //Program: Control of LED using potentiometer  //varibale for storing Analog values  int pot;  //potentiometer connected to pin A0  //LED connected to pin 11  int rled =11;  void setup()  {  Serial.begin(9600);  //initialise the analog pin A0 as input  pinMode(A0,INPUT);  //initialise pin 11 as output  pinMode(rled,OUTPUT);  }  void loop()  {  //Read the analog value and store in a varible  pot = analogRead(A0);  //map the variable from 0-1023 to 0-255  pot = map(pot,0,1023,0,255);  //control the brightness of the LED  analogWrite(rled,pot);  Serial.println(pot); // Printing potientometer value  //delay of 200ms  delay(200);  } |

# Workbook 7

### Activity 7.1: 2 Arduinos – using Digital Pins

Fritzing

|  |
| --- |
|  |

Arduino Program

// Akriti Kumari Dev

//2329584

// code for Lcd Display

// Code for Reciever Arduino

#include &lt;LiquidCrystal.h&gt;

#include &lt;Wire.h&gt;

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

void setup()

{

lcd.begin(16, 2); // Starting lcd display

Serial.begin(9600);

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

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

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

Wire.onReceive(dataReceived);

lcd.print(&quot;Waiting for data...&quot;);

lcd.setCursor(0,1);

lcd.print(&quot;Please wait...&quot;);

pinMode(8, OUTPUT);

}

void loop(){

}

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

{

int CelsiusVal = Wire.read(); //reading the sent Value

Serial.print(&quot;Temperature in celsius: &quot;);

Serial.print(CelsiusVal);

Serial.print(&quot;degC&quot;);

Serial.println();

int FahrenheitVal = Wire.read(); //reading the sent value

Serial.print(&quot;Temperature in fahrenheit: &quot;);

Serial.print(FahrenheitVal);

Serial.print(&quot;degF&quot;);

Serial.println();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(&quot;Celsius: &quot;);

lcd.print(CelsiusVal);

lcd.print(&quot;C&quot;);

lcd.setCursor(0, 1);

lcd.print(&quot;Fahrenheit: &quot;);

lcd.print(FahrenheitVal);

lcd.print(&quot;F&quot;);

}

// Akriti Kumari Dev

// 2329584

// Code for Temperature sensor connected Arduino

//allows comunication with I2C/TWI devices

#include &lt;Wire.h&gt;

int tempPin = A0; //initializing tempPin at A0

void setup()

{

delay(500); //wait for 500 millisecond

Wire.begin(); //begin I2Ccommunication for master or sender

Serial.begin(9600);

}

void loop()

{

float tempdata= analogRead(tempPin); //reading ADC value from the temperature sensor

float Vout= (tempdata\*5)/1023.0; // Convert ADC value to voltage (assuming 5V reference voltage)

int celsius = (Vout - 0.5) \* 100; //Converting the voltage into celsius

int fahrenheit = celsius \* (9.0 / 5.0) + 32.0; //Converting the data into fahrenheit

Wire.beginTransmission(8);//transmit to device with address 8

Wire.write(celsius); //send tmp value in Celsius

Wire.write(fahrenheit); //send tmp value in Fahrenheit

Wire.endTransmission(); //stop transmitting

delay(1000);

}

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

|  |
| --- |
|  |

Fritzing

Arduino Program

//Akriti Kumari Dev

// 2329584

// code for reciever

#include &lt;Wire.&gt;

void setup() {

Serial.begin(9600);

Wire.begin(8); // Set I2C address to 8 (slave address)

Wire.onReceive(receiveEvent); // Set receiveEvent as the function that gets called when data

is received

Serial.println(&quot;Waiting for data...&quot;);

}

void loop() {

// The loop function is empty as all the functionality is handled in the receiveEvent function

}

void receiveEvent(int bytesReceived) {

if (bytesReceived == 2) { // Check if two bytes were received

int celsius = Wire.read(); // Read the Celsius value

int fahrenheit = Wire.read(); // Read the Fahrenheit value

Serial.print(&quot;Temperature in Celsius: &quot;);

Serial.print(celsius);

Serial.print(&quot;degC, &quot;);

Serial.print(&quot;Temperature in Fahrenheit: &quot;);

Serial.print(fahrenheit);

Serial.print(&quot;degF&quot;);

Serial.println();

}

//Akriti Kumari Dev

// 2329584

// Code for for sender

#include &lt;Wire.h&gt;

int tempPin = A0;

void setup() {

delay(500);

Wire.begin(); // Start I2C communication

Serial.begin(9600);

}

void loop() {

float tempdata = analogRead(tempPin);

float Vout = (tempdata \* 5) / 1023.0;

int celsius = (Vout - 0.5) \* 100;

int fahrenheit = celsius \* 1.8 + 32;

Wire.beginTransmission(8); // Set the receiver&#39;s I2C address to 8

Wire.write(celsius); // Write the Celsius value

Wire.write(fahrenheit); // Write the Fahrenheit value

Wire.endTransmission();

delay(1000);

}

Workbook 8

### Activity 8.1: Stepper Motor Circuit Diagram

|  |
| --- |
|  |

Circuit Diagram

|  |
| --- |
| //Akriti Kumari Dev  //2329584  // Including the Arduino Stepper.h library:  #include <Stepper.h>  // Defining the number of steps per rotation:  const int stepsPerRevolution = 2048;  // Creating stepper object  Stepper myStepper = Stepper(stepsPerRevolution, 8, 10, 9, 11);  void setup() {  // Setting the speed to 5 rpm:  myStepper.setSpeed(5);  // Begin Serial communication at a baud rate of 9600:  Serial.begin(9600);  }  void loop() {  // First step one revolution in one direction:  Serial.println(“clockwise”);  myStepper.step(stepsPerRevolution);  delay(500);  //Second step revolution in the other direction:  Serial.println(“counterclockwise”);  myStepper.step(-stepsPerRevolution);  delay(500);  } |

### Activity 8.2: 2 Stepper Motors

Arduino Program

//Akriti Kumari Dev

//2329584

// Including the AccelStepper Library

#include <AccelStepper.h>

// Define step constants

#define FULLSTEP 4

#define HALFSTEP 8

// Creating two instances

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

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

void setup() {

// setting the maximum speed, acceleration factor,

stepper1.setMaxSpeed(1000.0);

stepper1.setAcceleration(50.0);

stepper1.setSpeed(200);

stepper1.moveTo(2038);

// setting the same for the motor 2

stepper2.setMaxSpeed(1000.0);

stepper2.setAcceleration(50.0);

stepper2.setSpeed(200);

stepper2.moveTo(-2038);

}

void loop() {

// Changing the direction once the motor reaches target position

if (stepper1.distanceToGo() == 0)

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

if (stepper2.distanceToGo() == 0)

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

// Moving the motor one step

stepper1.run();

stepper2.run();

}

Workbook 9

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

Arduino Code

//Akriti Kumari Dev

//2329584

#include &lt;Servo.h&gt;

Servo myServo;

int servoPin1=9;

int servoPin2= 10;

Servo myServo1;

Servo myServo2;

int angle=0;

int tempPin = A0;

void setup(){

myServo1.attach(servoPin1);

myServo2.attach(servoPin2);

pinMode(tempPin, INPUT);

}

void loop(){

float tempdata= analogRead(tempPin);

float Vout= (tempdata\*5)/1023.0;

int temp=(Vout - 0.5) \* 100;

if (temp&gt;=25){

for(angle=0; angle&lt;=90; angle++){

myServo1.write(angle);

myServo2.write(angle);

delay(10);

}

for (angle=90; angle&gt;=0; angle--){

myServo1.write(angle);

myServo2.write(angle);

delay(10);

}

}

else

{

myServo1.write(0);

myServo2.write(0);

delay(15);

}

}

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 handwritten work is not acceptable.

Marking

All sections carry equal marks.