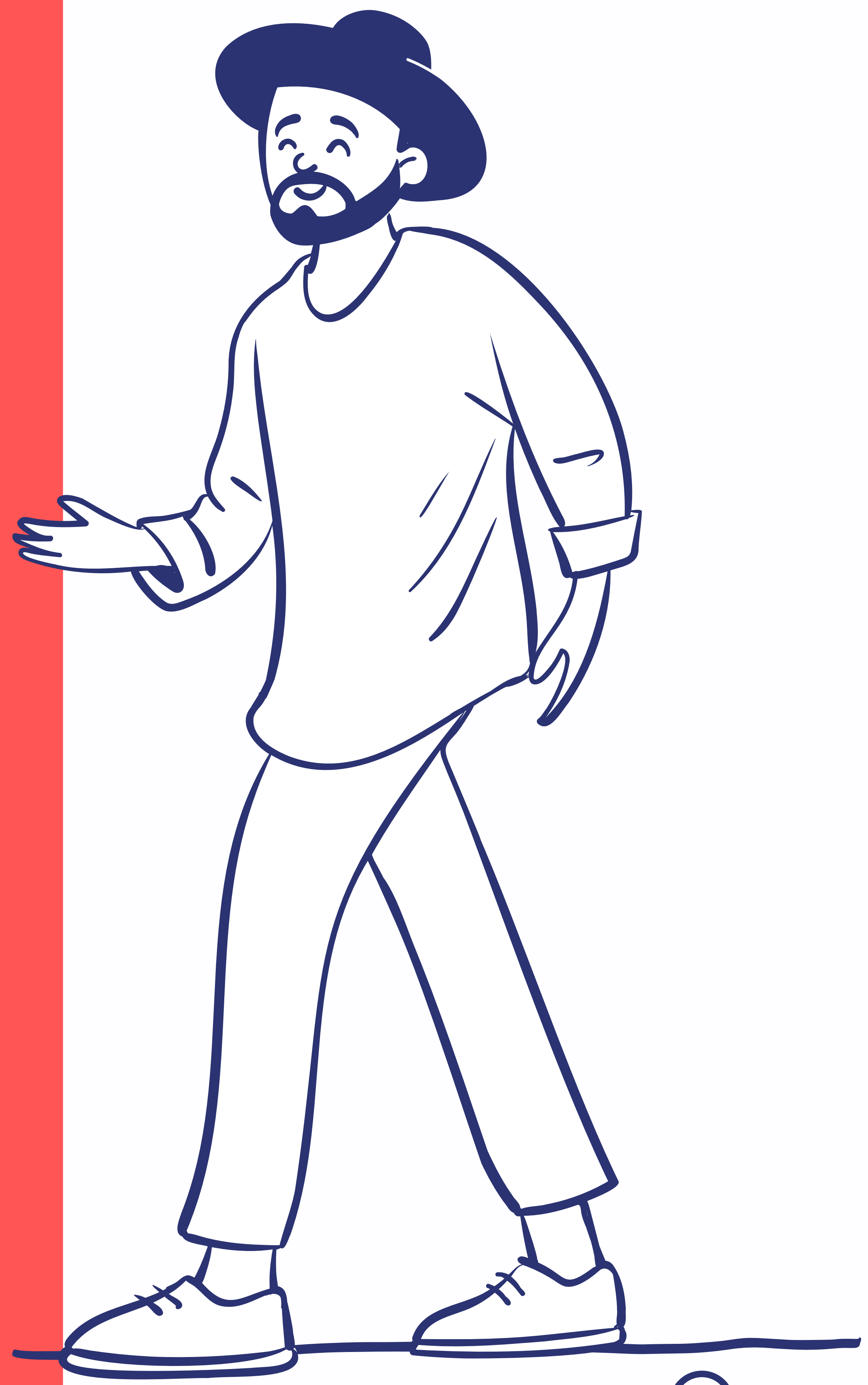
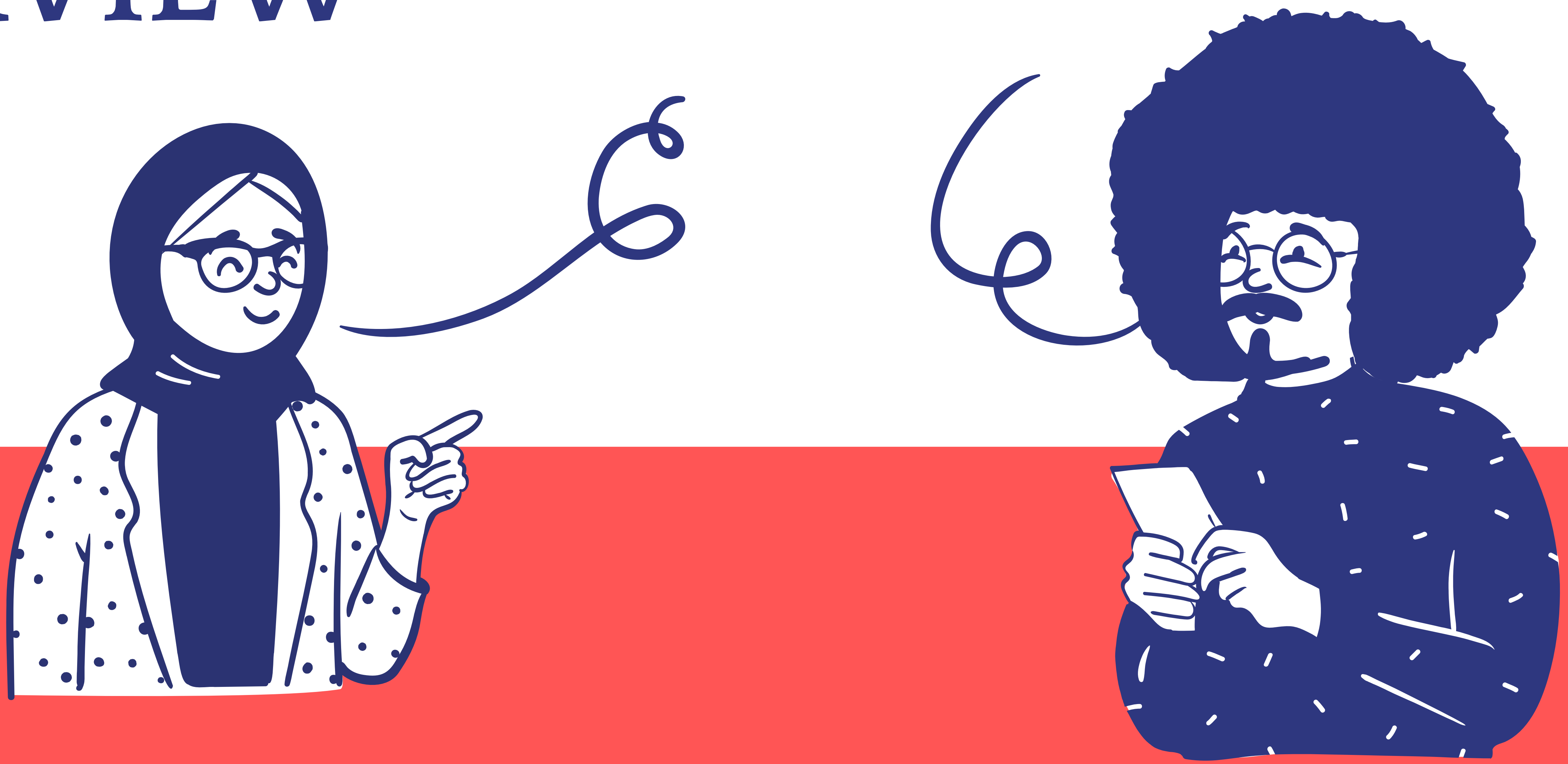
Thermal Regulation in Active Temperature Sensing

-Final Presentation-

By:
Alisha Hilmi
Dharren Sandhi
Gracia Aretha



1. OVERVIEW



Thermal regulation in active temperature sensing means that the nichrome (heater) temperature needs to be maintained constant by running a current through it and when a hand is placed on the nichrome, theoretically, it would increase the resistance of the heater, which would then causes a change in current. This change in current can then be used to calculate the temperature of the hand.

2. EQUIPMENTS

01 Power Supply (12V 10A)

02 Resistors
Nine 100 Ohm 2 Watt Resistors
One 10k Ohm 0.25 Watt Resistor

03 Nichrome wire (10 cm)
The Nichrome wire is a fuse wire that is used as a conductor as it can easily conduct electric current. It has a resistivity of 100 10-8 Ωmand a melting point of 1672℃,

04 K-Type Thermocouple

K-Type Thermocouples have a Chromel positive leg and an Alumel negative leg and is used to measure the temperature of its surroundings.

05 MOSFET IRLZ44N

The IRLZ44N is suitable to be used for low voltages and is used as a switch with an Arduino for controlling of higher power devices.

06 Arduino

07 LCD

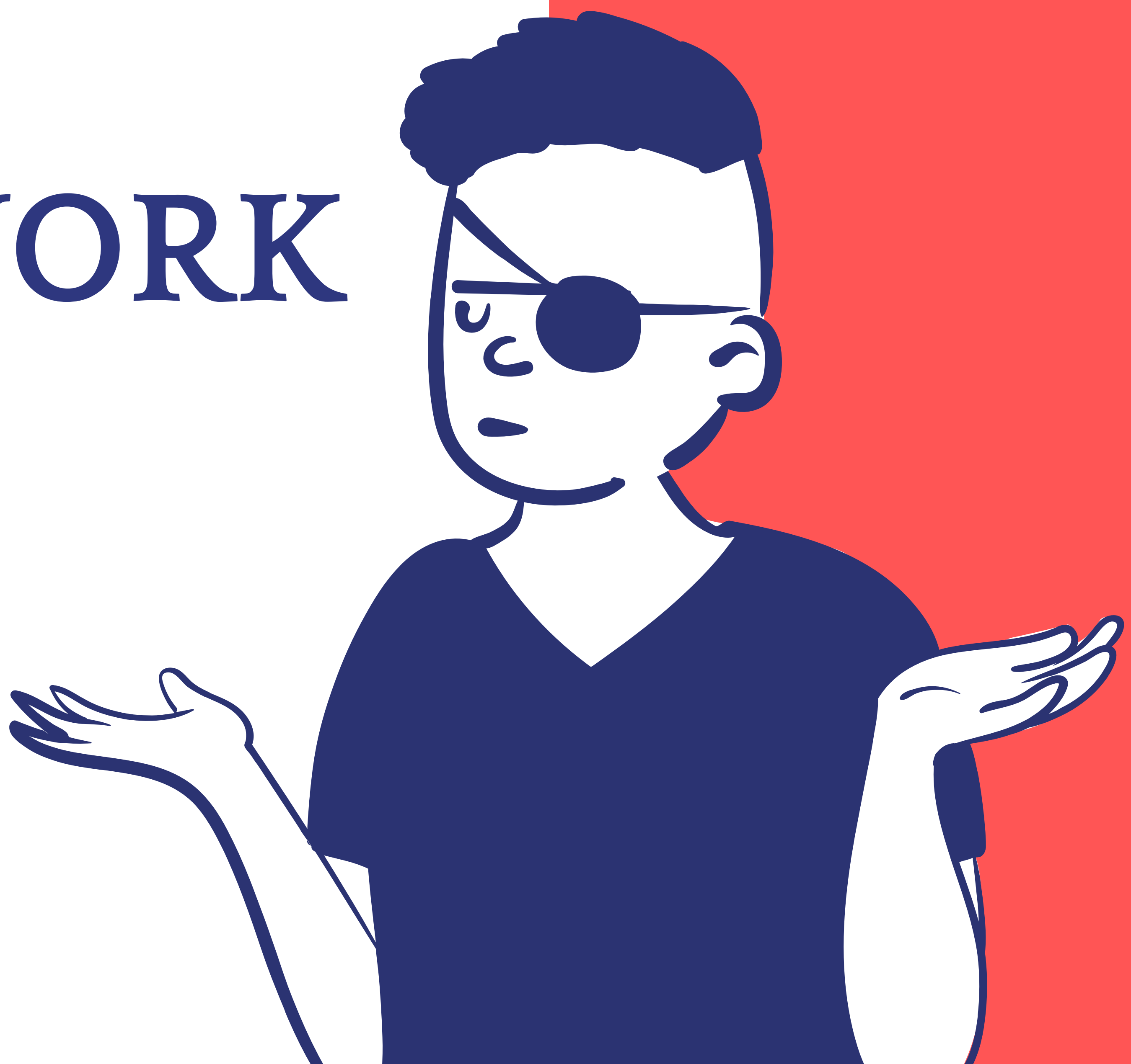
08 MAX6675

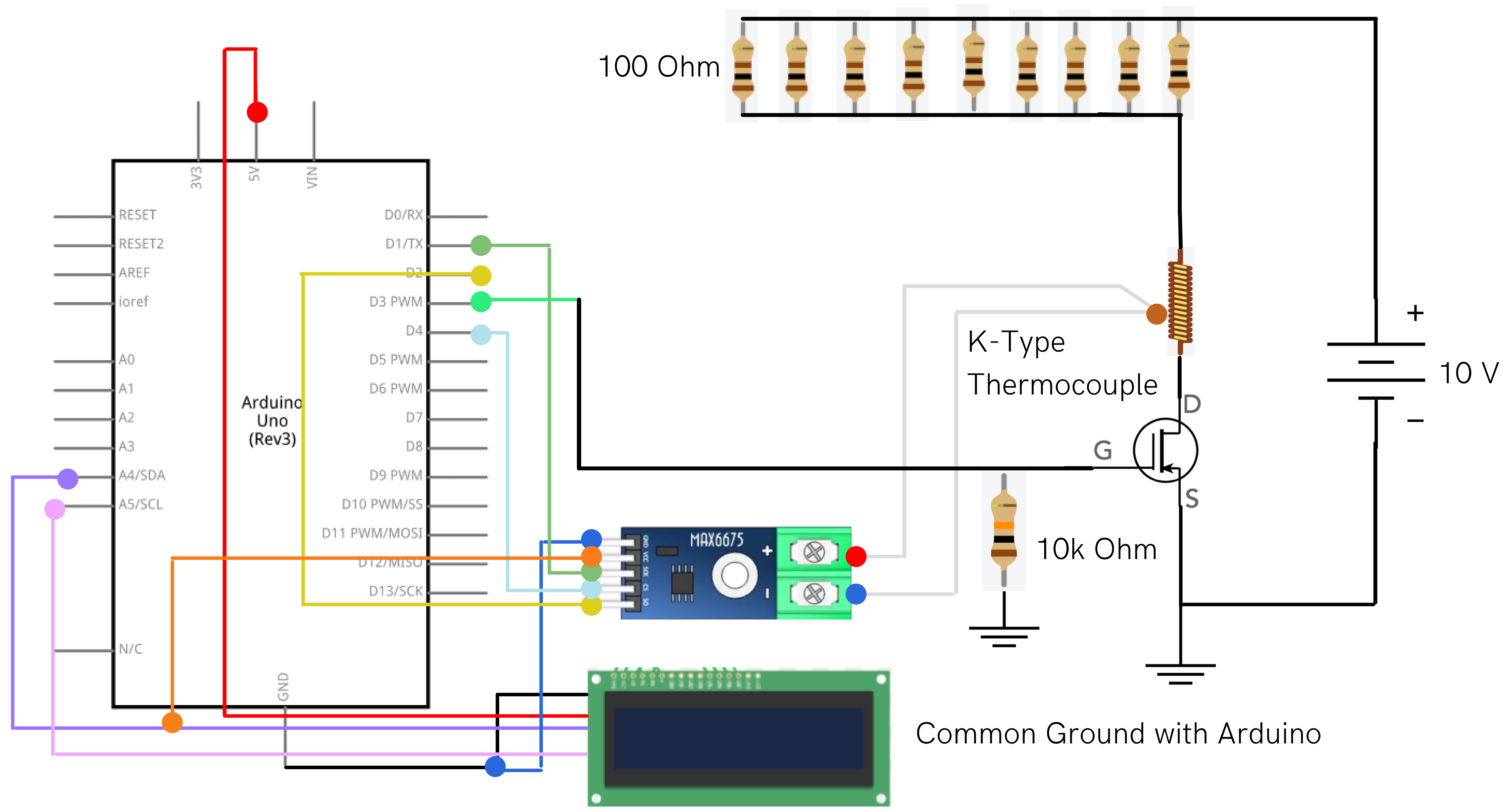
The MAX6675 converts signals from K-type thermocouples. It displays thermocouple temperatures with resolution of 0.25 C, ranging from 0°C to 1024°C.

09 Thermal Pad

Pre-formed rectangles of solid material and is a very good conductor of heat but a poor conductor of electricity

3. HOW DOES IT WORK





100 Ohm
3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
Arduino
Uno
(Rev3)
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
MAX6675
K-Type
Thermocouple
D
G
S
10k Ohm
+
10 V
−
Common Ground with Arduino

A) THERMAL REGULATION

PID CONTROL

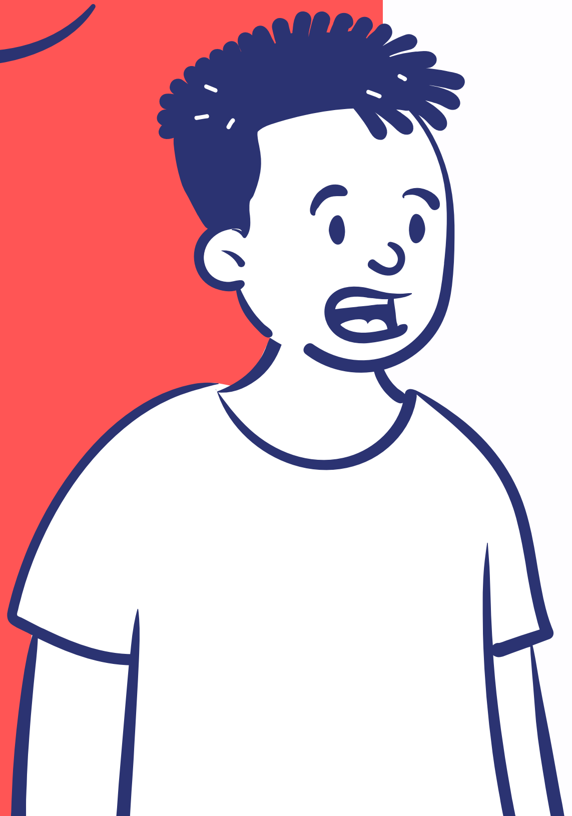
In this project only PI control is implemented

$$u(t) = K_p e(t) + K_i \int e(t)dt$$

Proportional - Calculates the error between the measured temperature and the setpoint temperature (30C) and is used to determine the rate of system response

Integral - Sums up all the previous errors and is used to reduce the offset

The output of the PI control is mapped to the PWM value, so it will output a value of 0 to 255



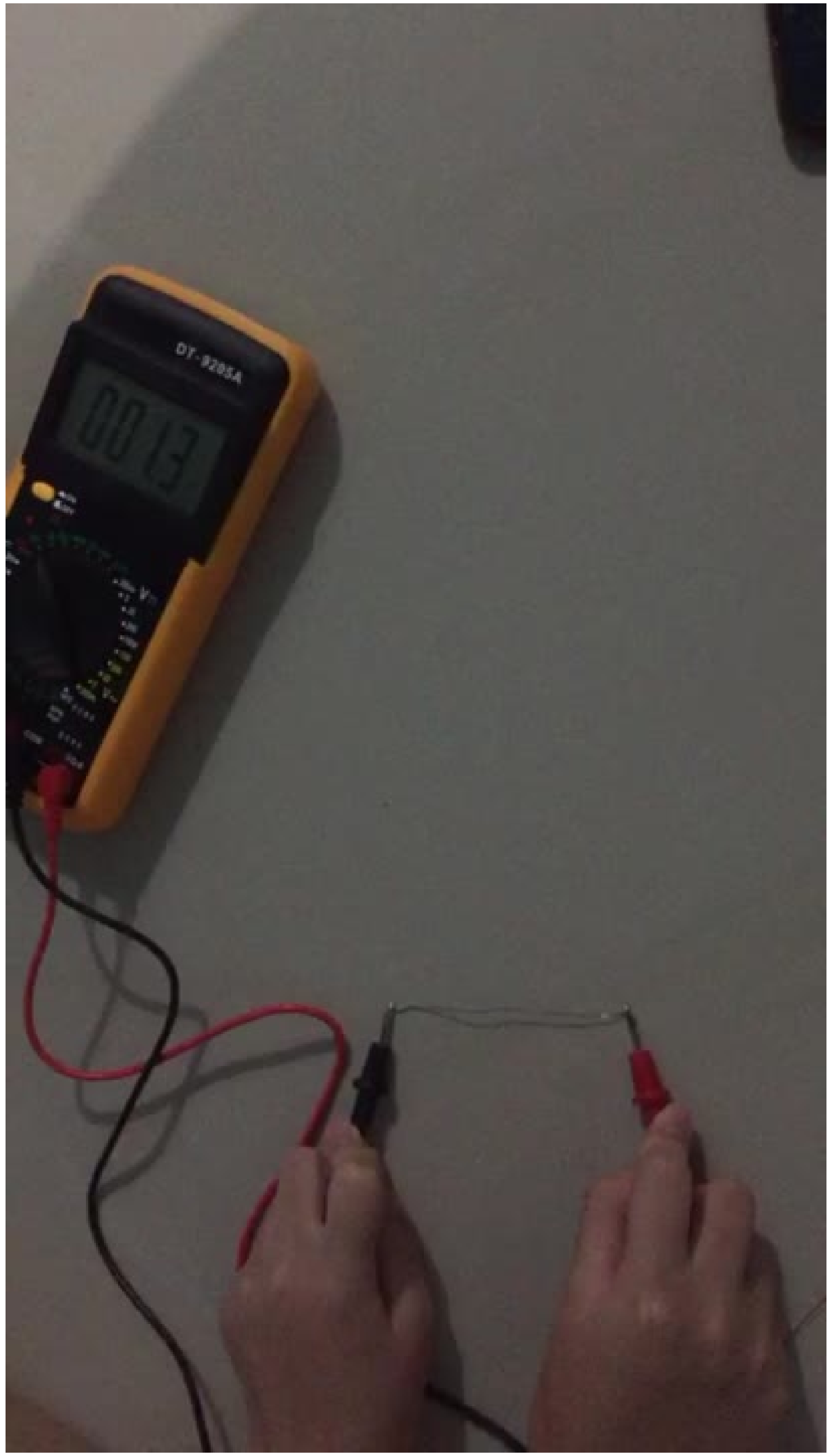
B) Temperature Sensing

Problem Encountered With Nichrome

01 For active temperature sensing, a noticeable change in current is needed, because this change in current will be used to calculate the temperature of the body. However, the nichrome wire did not have a noticeable change in resistance over small temperatures (like the body temperature), thus very little to no change in current

02 A new method is approached to face the problem, by using the change in nichrome temperature when the hand is placed

BEFORE



AFTER

How the Temperature was Measured

01 After running the program, it was found that at best, the temperature fluctuates between 29.75 C to 30.25 C (setpoint was set to 30 C)

02 When the temperature reached the range of fluctuations, the hand was placed onto the nichrome wire

03 When the temperature increased above 30.25 C, an algorithm was done to put all the temperature readings into an array while the hand was placed

04 From the array, the highest value was taken and subtracted to the setpoint value, and was put into a linearized equation to calculate hand temperature

HOW TO GET THE LINEARIZED EQUATION

1.

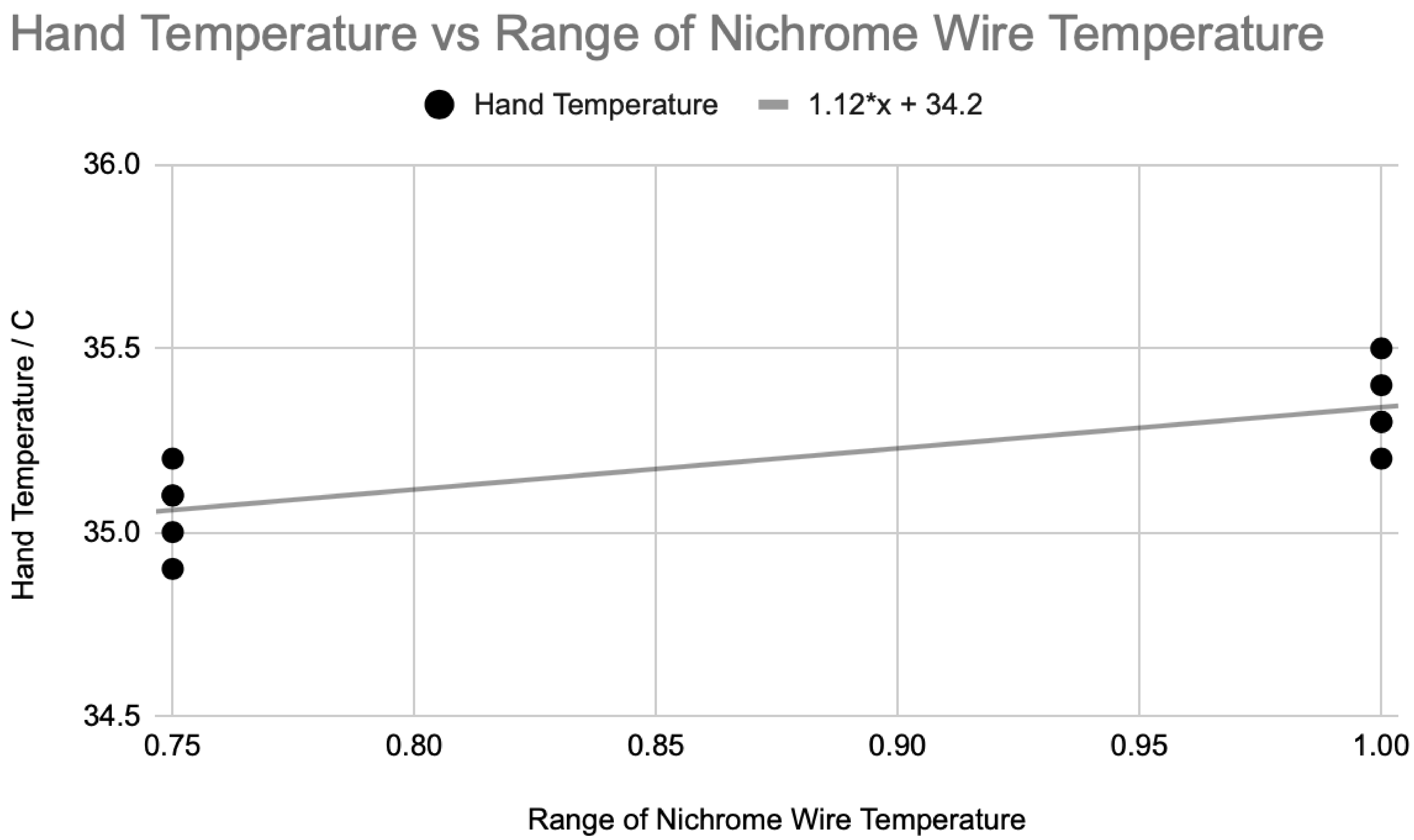
Hand temperature was measured using a body thermometer

2.

When the nichrome wire reached a stable temperature (30C), the hand was placed on to the wire for 2 minutes and the largest change (range) in nichrome wire temperature was recorded

3.

Procedures 1 and 2 was repeated 10 times and a linearised graph was made



Equation for the relationship between range of nichrome wire temperature and hand temperature:

$$T = 1.12x + 34.2$$

SCENARIO EXAMPLE

[30.50, 30.50, 30.75, 30.50, 30.75, 30.75, 31, 30.50, 30.25, 30.50]

1st Highest Value

30.50 - 30 = 0.50

T = 1.12(0.50) +34.2 = 34.76

2nd Highest Value

30.75 - 30 = 0.75

T = 1.12(0.75) +34.2 = 35.04

3rd Highest Value

31 - 30 = 1

T = 1.12(1) +34.2 = 35.32 → Final Hand Temp.

Final Coding

Initialising Variables and Libraries

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

//thermocouple
#include <max6675.h>

int MAX6675_CS = 10;
int MAX6675_SO = 12;
int MAX6675_SCK = 13;

MAX6675 ktc(MAX6675_SCK, MAX6675_CS, MAX6675_SO);

//pid
#include <PID_v1.h>

//Pins
int PWM_pin = 3;

//Variables
#define SAMPLETIME 500
double Setpoint, currentTemp, Output, Voltage, Current, bodyTemp, rangeTemp,
const unsigned int numReadings = 10;
double tempVal[numReadings];

PID myPID(&currentTemp, &Output, &Setpoint, 230, 0.9, 0 , DIRECT);
```

P I D

```
void setup() {
  Serial.begin(9600);
  pinMode(PWM_pin,OUTPUT);
  currentTemp = ktc.readCelsius();
  Setpoint = 30;
  myPID.SetMode(AUTOMATIC);
  myPID.SetSampleTime(SAMPLETIME);
  lcd.init();
  lcd.backlight();
}
```

Starting the PID

Sets how often the PID calculates the output

```
void loop() {
  //Temperature Reading
  currentTemp = ktc.readCelsius();
  //Convert PWM to Current Reading
  Voltage = (Output/255)*10;
  Current = (Voltage/13)*1000;

 //Active Temperature Sensing
 if (currentTemp > (Setpoint+0.25))
 {
   for (int i = 0; i <= (numReadings-1); i++)
   {
     tempVal[i] = ktc.readCelsius();
     delay(500);
     if (tempVal[i] > largest)
     {
       largest = tempVal[i];
     }
     rangeTemp = largest - Setpoint;
     bodyTemp = (rangeTemp*1.12) + 34.2;
   }
 }
 //
```

Start making an array and adding thermocouple readings to the array

Algorithm to determine the highest value in an array

```
    myPID.Compute();
    analogWrite(PWM_pin, Output);
    Serial.println(currentTemp);
    delay(SAMPLETIME);

    lcd.setCursor(0, 0);
    lcd.print("S:");
    lcd.setCursor(2, 0);
    lcd.print(Setpoint);

    lcd.setCursor(9, 0);
    lcd.print("C:");
    lcd.setCursor(11, 0);
    lcd.print(Current);

    lcd.setCursor(0, 1);
    lcd.print("T:");
    lcd.setCursor(2, 1);
    lcd.print(currentTemp);

    lcd.setCursor(9, 1);
    lcd.print("B:");
    lcd.setCursor(11, 1);
    lcd.print(bodyTemp);
}
```

Automatically computes PID output value

Uses the output value of PID as the PWM signal

6. Conclusion and Suggestion

01 PID was successfully implemented with a satisfactory offset of only 0.25 from the set temperature (fluctuating between 29.75 to 30.25)

02 Hand (body) temperature was able to be calculated through a linearised equation between real hand temperature and the range of the nichrome wire temperature when the hand is placed

01 Nichrome is not a suitable active temperature sensing device because of the fact that it does not have a noticeable change in resistance over a small temperature (such as the body temperature).