DEPARTMENT OF ELECTRONICS ENGINEERING

Dwarkadas J. Sanghvi College of Engineering

Plot U-15, J.V.P.D Scheme, Bhaktivedanta Swami Marg, Vile Parle (West), Mumbai-400056.

Mumbai University

25th April, 2018.

Temperature Control of Peltier Element

A mini project report submitted for

Mini Project-II (semester VI)

By

Gagan Agarwal (60001150002)

Dhairya Gada (60001150011)

Kenil Gadhiya (60001150012)

Under the guidance of

Prof. Bahar Soparkar

(SIGN & DATE)

Temperature Control of Peltier Element

Gagan Agarwal
T.E. Electronics
D.J. Sanghvi C.O.E.
60001150002
agarwal.gagan5101@gmail.com

Dhairya Gada
T.E. Electronics
D.J. Sanghvi C.O.E.
60001150011
dhairyagada1997@gmail.com

Kenil Gadhiya
T.E. Electronics
D.J. Sanghvi C.O.E.
60001150012
kenilgadhiya@gmail.com

***Abstract*— The aim of this project is to control and maintain the temperature of a Peltier Element using PID control loop.**

I. OBJECTIVE

The objective of this project is to accept a temperature value from user, calculate its deviation from the sensed current temperature value and control the current of a Peltier element to achieve a steady state at the desired temperature with the help of a PID control loop logic.

II. APPROACH

A. Testing the Peltier Element

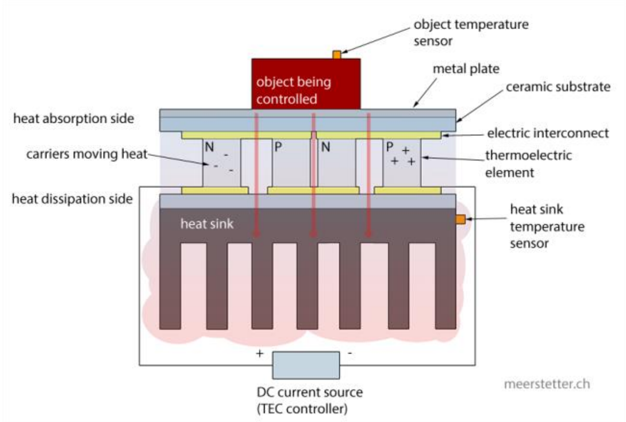

Figure 1 Peltier Element

Peltier Element also known as a Thermo Electric Cooler is a semiconductor technology that concentrates the heat on one of its faces making the other one cooler.Therefore one of the sides becomes extremely warm and the other one becomes extremely cold.

The polarity of the supply decides which face is warm or cool. Reversing the polarity makes the warmer side cool and vice versa.

The peltier works on a 12 V DC supply but draws a large current. The peltier we'll be using in our project has a capacity to draw 4 A Current for maximum Heating/Cooling. The amount of current it draws decides the magnitude of Heating/Cooling it can provide. Thus the amount of temperature change that we want can be controlled by controlling the amount of current supplied to it.

B. Heat Sink

The efficiency of how cold a side of the peltier gets depends upon how efficiently the other side of the device dissipates the heat. Thus the requirement of a heat sink and a fan to dissipate/spread the heat on one of the faces.

C. Low Pass Filter

The microcontroller regulating the temperature will produce PWM pulses. These pulses need to be filtered so that the Peltier device sees relatively smooth current.

The following filter circuit takes PWM pulses as input to the gate terminals of the N-Channel MOSFETS and produces a smooth output current proportional to the value of PWM signal.

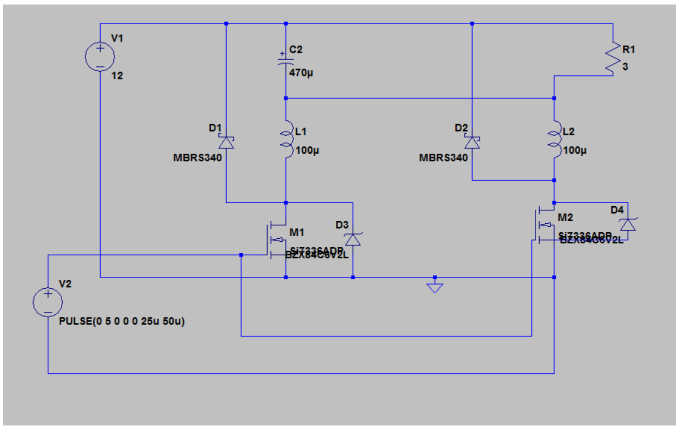

Figure 2 Filter Circuit

The 3 Ohm Resistor represents the Peltier Element.

D. Microcontoller Setup

We have used Arduino UNO that uses Atmel Microcontroller ATMEGA 328P as the control unit for this project.

The Arduino is interfaced with a 16x2 LCD and a 22K Potentiometer. The potentiometer acts a frontend input for the user to set a desired temperature value. The LCD displays the current temperature reading along with temperature set by the user.

E. Temperature Sensing

The temperature is sensed by the sensor DHT11 which is a temperature and a humidity sensor.

F. Relay Mechanism

A relay Mechanism is used to switch the polarity of the Peltier Element connections depending on the requirement if the top plate needs to be below the room temperature or above the room temperature.

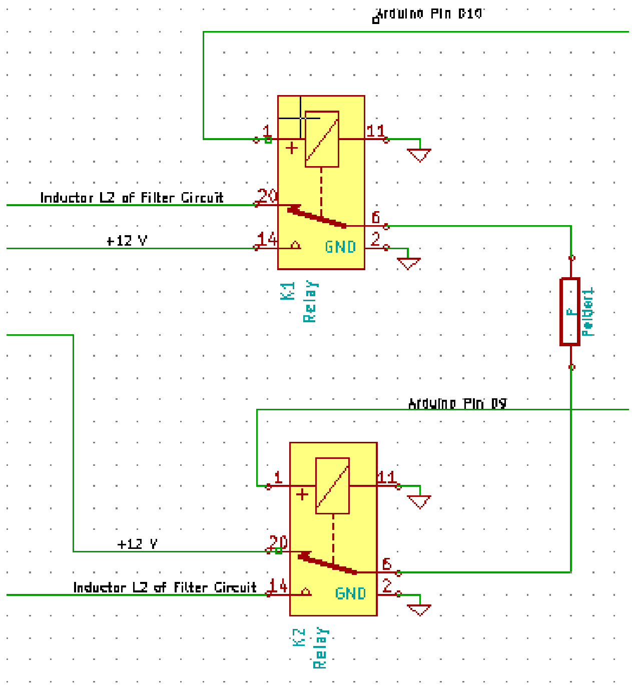

Figure 3 Relay Circuit

G. Power Supply Arrangement

We have used a Switch Mode Power Supply that connects to the AC Mains supply and provides a constant 12 V DC voltage.

The SMPS can supply current ranging from 0 to 10 A which makes is suitable for our project.

A 7805 Voltage Regulator is used to obtain 5V supply for the Arduino Board.

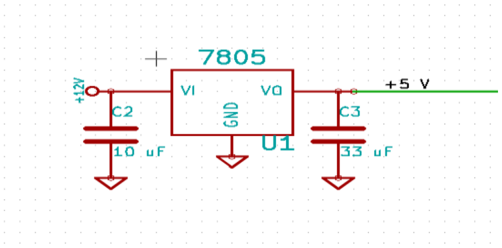

Figure 4 Voltage Regulator 7805

III. Simulation Results

Following are the simulation results for LTSpice Simulation of the Low Pass Filter circuit.

When PWM duty cycle is zero, no current flows through the resistor.

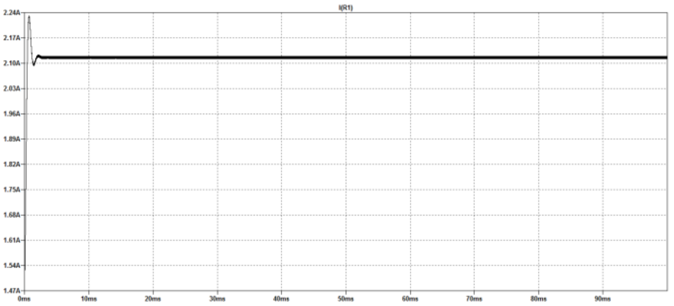

Figure 5 Duty Cycle 25%

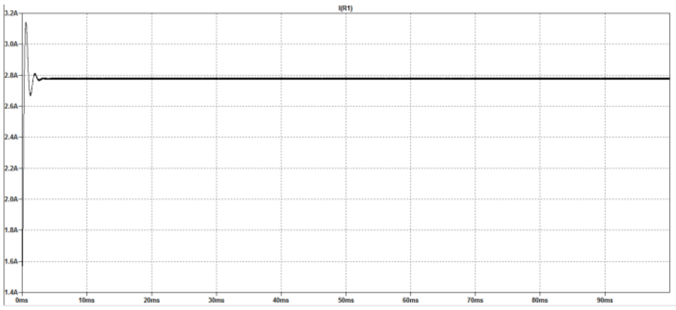

Figure 6 Duty Cycle 50%

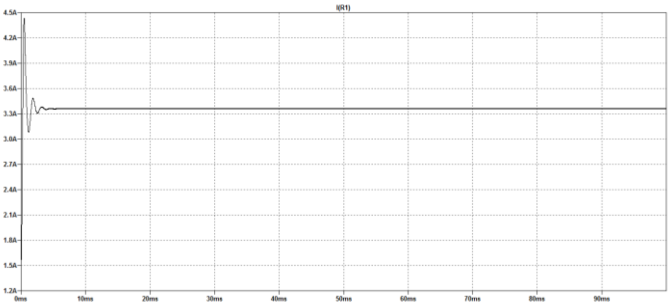

Figure 7 Duty Cycle 75%

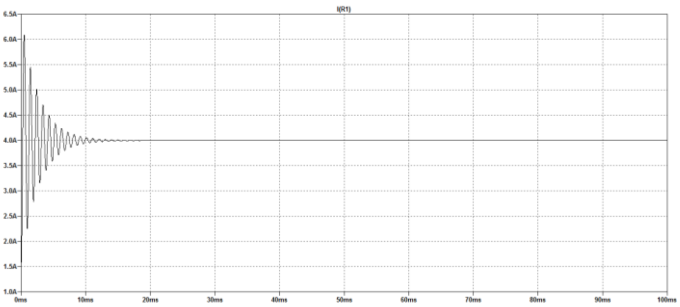

Figure 8 Duty Cycle 100 %

The output of the filter smoothens after a transience that lasts no more than 10ms, and the magnitude of the output current changes in proportion with the PWM duty cycle.

IV. SCHEMATIC

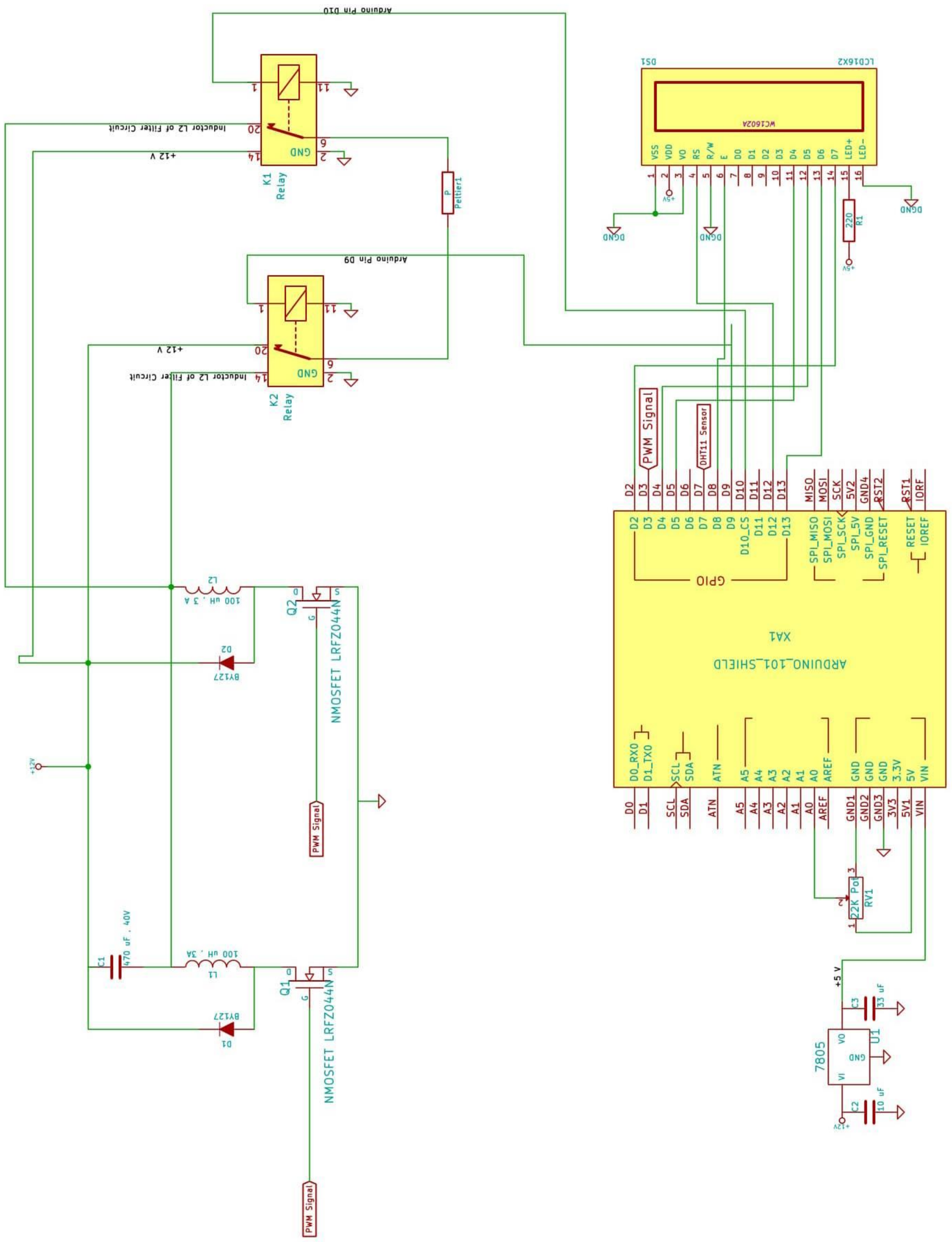

Figure 9 Final Schematic

V. PROGRAM

```
#include <LiquidCrystal.h>
#include <dht.h>
dht DHT;
#define DHT11_PIN 7
const int rs = 12, en = 8, d4 = 5, d5 = 4, d6 = 13, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int temp=0,temp1=0;
float potchange=0;
int sensorPin=A0;
int tempPin = A4;
int kp=0,kd=0,ki=0;
int temperature_read=0 , set_temperature=0;
int PID_error=0;
float previous_error = 0;
float PID_p=0,PID_d=0,PID_i=0,PID_value=0;
float elapsedTime, Time, timePrev;
int RoomTemp=0;
void setup() {
            Time = millis();
            pinMode(10,OUTPUT);
            pinMode(9,OUTPUT);
            pinMode(3, OUTPUT);
            TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM20);
            TCCR2B = _BV(CS20);
            OCR2B = 0;
            Serial.begin(9600);
            int chk = DHT.read11(DHT11_PIN);
            for(int i=0; i<=10; i++) {
              chk = DHT.read11(DHT11_PIN);
              lcd.begin(16, 2);
              lcd.setCursor(2, 0);
              lcd.print("Initial Setup");
              lcd.setCursor(5, 1);
              lcd.print(i);
              RoomTemp=DHT.temperature;
              lcd.setCursor(9, 1);
              lcd.print((int)RoomTemp);
              delay(1200);
              Serial.println(RoomTemp);
            }|
            delay(3000);
            RoomTemp=DHT.temperature;
            lcd.setCursor(2, 0);
            lcd.print("RoomTemp");
            lcd.setCursor(9, 1);
            lcd.print((int)RoomTemp);
            Serial.println("RoomTemp");
            Serial.println(RoomTemp);
}
void loop() {
  // LCD Setup
  lcd.clear();
  lcd.begin(16, 2);
  lcd.print("Set Temp: ");
  lcd.setCursor(10, 0);
  // DHT11 Reading
  int chk = DHT.read11(DHT11_PIN);
  // Potentiometer Reading
  temp=analogRead(sensorPin);
  potchange=temp;
```

```
 temp1=15+(temp/1023.0)*41.0;
 // LCD Printing
 lcd.print(temp1);
 lcd.setCursor(13, 0);
 lcd.print("C");
 lcd.setCursor(0,1);
 lcd.print("Actual Temp:");
 lcd.print((int)DHT.temperature);
 lcd.setCursor(15,1);
 lcd.print("C");
 Serial.println((int)DHT.temperature);
 Serial.println(temp1);
 set_temperature=temp1;
 temperature_read= DHT.temperature;
 // ------------------------------ HEATING ----------------------------------------------------------
 if(set_temperature>=RoomTemp){
       digitalWrite(9,HIGH);
       digitalWrite(10,HIGH);
       lcd.setCursor(15, 0);
       lcd.print("H");
       kp=50;
       kd=50;
       ki=0.001;
       PID_error = -(temperature_read - set_temperature);
       PID_p = kp * PID_error;
       PID_i = (PID_i + (ki * PID_error));
       timePrev = Time;
       Time = millis();
       elapsedTime = (Time - timePrev) / 1000;
       PID_d = kd*((PID_error - previous_error)/elapsedTime);
       PID_value = PID_p + PID_d + PID_i ;
        if(PID_value < 0 || PID_error<0 || PID_p < 0)
         {    PID_value = 0; }
       if(PID_value > 255)
         {    PID_value = 255;  }
       OCR2B=PID_value;
       previous_error = PID_error;
 }
 // ------------------------------ COOLING ----------------------------------------------------------
 else if(set_temperature<RoomTemp){
       digitalWrite(9,LOW);
       digitalWrite(10,LOW);
       lcd.setCursor(15, 0);
       lcd.print("C");
       kp=30;
       kd=12;
       ki=1;
       PID_error = (temperature_read - set_temperature) ;
       PID_p = kp * PID_error;
       PID_i = (PID_i + (ki * PID_error));
       timePrev = Time;
       Time = millis();
       elapsedTime = (Time - timePrev) / 1000;
       PID_d = kd*((PID_error - previous_error)/elapsedTime);
       PID_value = PID_p + PID_d + PID_i ;
        if(PID_value < 0 || PID_error<0 )
         {    PID_value = 0; }
       if(PID_value > 255)
         {    PID_value = 255;  }
       OCR2B=PID_value;
       previous_error = PID_error;
}
 delay(1500); }
```

VI. BILL OF MATERIALS

TABLE I

Component	Cost (Rs.)	Quantity	Total (Rs.)
Arduino UNO	250	1	250
16 x 2 LCD	150	1	150
Peltier Element	200	1	200
DHT11	150	1	150
Relays	30	2	60
MOSFET LRFZ044N	20	2	40
Diode BY127	10	2	20
Inductor 100uH , 3A	20	2	40
Capacitors	5	3	30
IC 7805	15	1	15
GPB	50	1	50
Total (Rs.)			**1015**

VII. CONCLUSION

The implemented project is thus able to regulate a user desired temperature of Peltier Element with a tolerance of ±1°C. The temperature can be controlled between 22 °C to 60°C.

The project can be further improved by the use of a better temperature sensor that has a quick thermal response and faster sampling. A better steady state response along with low overshoot and faster settling time can be achieved by using computer aided system modelling and PID tuning on soft wares such as Simulink and Matlab.

VIII. APPLICATIONS

A. Cooling Operations in Consumer Products

The Peltier device can be used to create a portable Coolers and Refrigerators that are compact and have no circulating fluid or moving parts. Such devices are useful in applications where their advantages outweigh the disadvantage of their very low efficiency.

Peltier coolers can be used to cool computer components to keep temperatures within designed limits, or to maintain stable functioning when overclocking. A Peltier cooler with a heatsink can cool a chip to well below ambient temperature.

B. Applications in the process'es of Biomedical Research

Thermal cyclers (thermo-cycler, PCR machine or DNA amplifier) are used for temperature-sensitive reactions in laboratories like live cell imaging. Peltier elements allow a great temperature range, high temperature stability and precision in combination with precise TEC controllers.

Micro scale Thermophoresis is used to quantify bio molecular interactions, measuring the motion of molecules along temperature gradients. Peltier elements are used to keep the base plate at a constant temperature or stabilize the infrared-laser.

C. Fiber Optic Applications

In fiber optic applications, where the wavelength of a laser or a component is highly dependent on temperature, Peltier coolers are used along with a thermistor in a feedback loop to maintain a constant temperature and thereby stabilize the wavelength of the device.

D. Industrial Processes

Countless applications in industrial processes are facilitated using thermoelectric modules. Examples are metallurgy, semiconductor lithography and hardening / curing needs.