END SEMESTER ASSESSMENT (ESA)
B.TECH. (CSE)
IV SEMESTER

UE18CS256 – MICROPROCESSOR AND COMPUTER ARCHITECTURE LABORATORY

PROJECT REPORT

ON

SMART REFRIGERATOR

SUBMITTED BY

NAME	SRN
1) Raeesa Tansen	PES2UG19CS310
2) Ramit Bathula	PES2UG19CS319
3) Ria Singh	PES2UG19CS326
4) R Shailesh	PES2UG19CS307

JANUARY – MAY 2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ELECTRONIC CITY CAMPUS,
BENGALURU – 560100, KARNATAKA, INDIA

ABSTRACT OF THE PROJECT:

We have created a smart refrigerator microcontroller using Arduino UNO. It is simulated on Tinkercad. It is designed like the mini fridges in hotels and is a prototype for water bottle units as of now. It detects the number of water bottles kept inside and displays the subsequent output to the user so that they know if it needs to be restocked or not.

It consists of four Force Sensitive sensors, a PIR sensor, a Temperature sensor, a LCD 16x2 display , a DC motor, a potentiometer, one LED light, and one RGB LED light.

The PIR sensor is used to detect whether the fridge has been opened or not, and if it detects any motion, it switches on the LED in the fridge. If the fridge is empty, the LED is switched off irrespective of whether the PIR detects any motion. The force sensors act as pressure sensors and give us an inventory of how many items are present in the fridge. If no force is detected then the fridge is empty

The DC motor is connected to an internal fan which is used for ventilation and to remove heat from the inside, and the speed is based on the temperature. The LCD display displays the temperature and the number of items present in the fridge at all times detected by the temperature and force sensors respectively. The RGB LED light shines either red, green or blue depending on the temperature inside the fridge. (RED - extreme, GREEN - optimum, BLUE - very low). The potentiometer controls the brightness of the LCD screen by varying the resistance.

TECH STACK:

Force Sensor

The force sensor or FSR (Force-sensitive) resistor is used because it changes the value of current passing through the sensor according to the pres vc sure on the sensor. Depending on the weight, the sensor's resistance changes accordingly. Weight is directly proportional to the resistance of the sensor in the circuit.

We have used FSR to detect the quantity of water bottles based on the presence of application of their weight. We have included these so that the user can view their inventory on the LCD without having to actually open the refrigerator and manually count.

When the sensor detects force/pressure on it, the value is incremented by one, determining the presence of one unit bottle. This is how we calculate the number of water bottles present.

PIR Sensor

The Passive Infrared Motion sensor is used to detect the presence of the user in front of the refrigerator to regulate the powering of internal lights, which is depicted by an LED light in our tinkercad prototype.

Subsequently, if the fridge is empty, the lights remain switched off in order to conserve energy.

Temperature Sensor

The Temperature Sensor LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature. Depending on the temperature of the refrigerator, the cooling and ventilation system represented by an internal fan connected to a DC motor is induced.

We also have a general RGB indicator for the temperature and the DC motor.

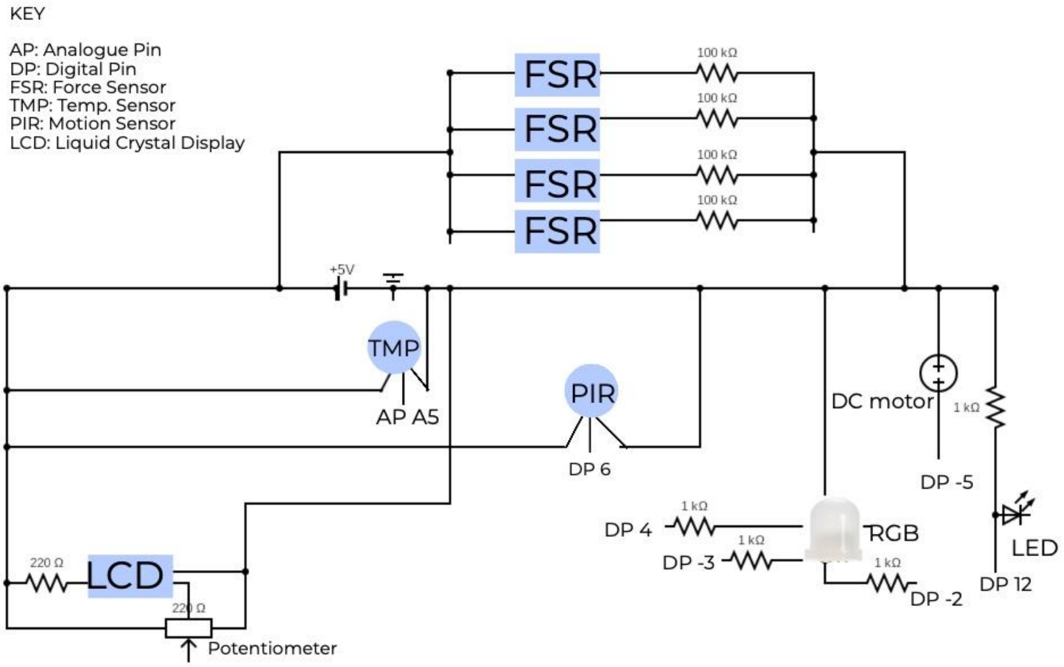
- If the temperature is too low, the RGB turns Blue, and the DC motor is reduced to ~1000 rpm.
- For optimum temperature, the RGB is green and the DC motor runs at ~ 2000rpm-3000 rpm.
- If the temperature has risen dramatically, and it is too hot, the RGB turns Red and the DC motor runs at ~4000-5000 rpm.

We have also used a potentiometer which is used to control the brightness of the LCD screen by varying the resistance.

COMPONENT LIST:

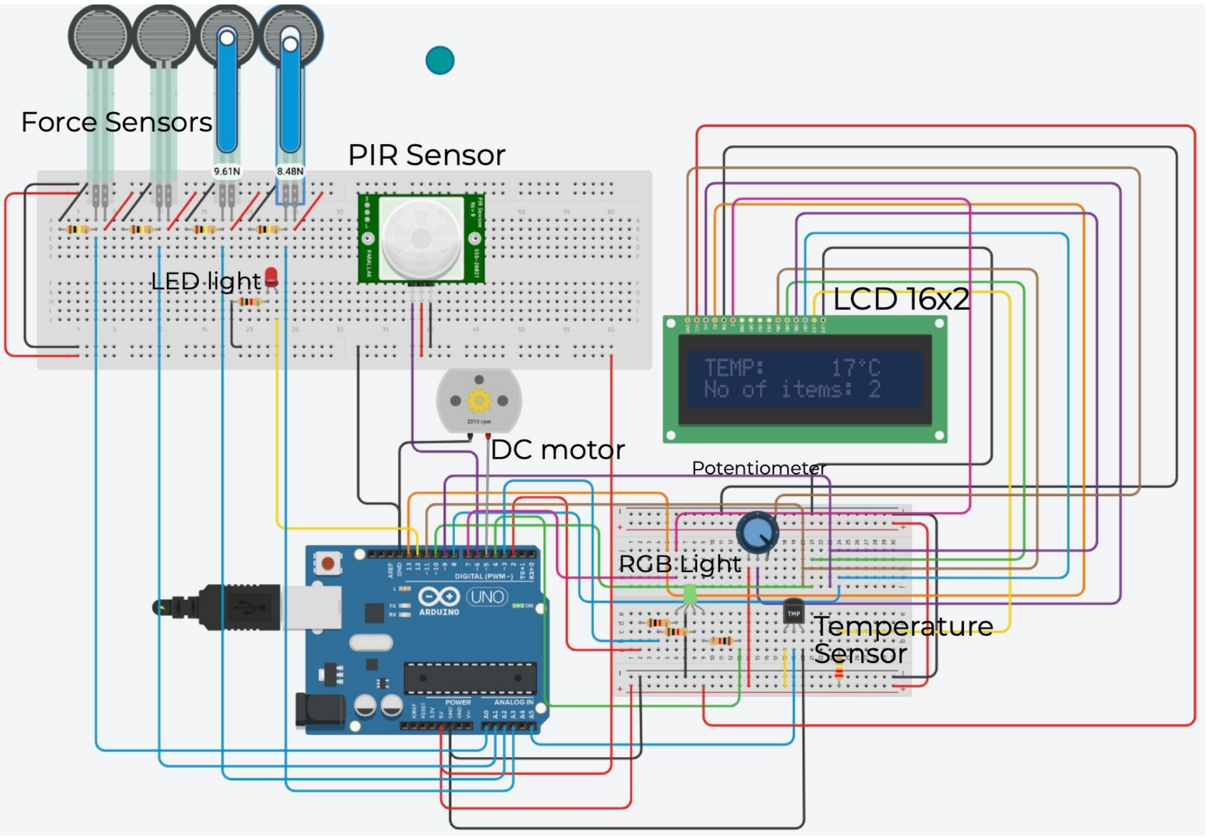
Name	Quantity	Component
U1	1	Arduino Uno R3
D1	1	LED RGB
D3	1	Red LED
PIR1	1	31.610301581960243 , -187.9571324839402 , -179.57963437464946 , -142.78336820641243 PIR Sensor
U3	1	Temperature Sensor [TMP36]
R2 R3 R4 R5	4	1 kΩ Resistor
U2	1	LCD 16 x 2
Rpot1	1	250 kΩ Potentiometer
R1	1	220 kΩ Resistor
M1	1	DC Motor
R6 R7 R10 R11	4	Force Sensor
R12 R14 R15 R16	4	100 kΩ Resistor

CIRCUIT DIAGRAM:



KEY

AP: Analogue Pin
DP: Digital Pin
FSR: Force Sensor
TMP: Temp. Sensor
PIR: Motion Sensor
LCD: Liquid Crystal Display



ARDUINO CODE:

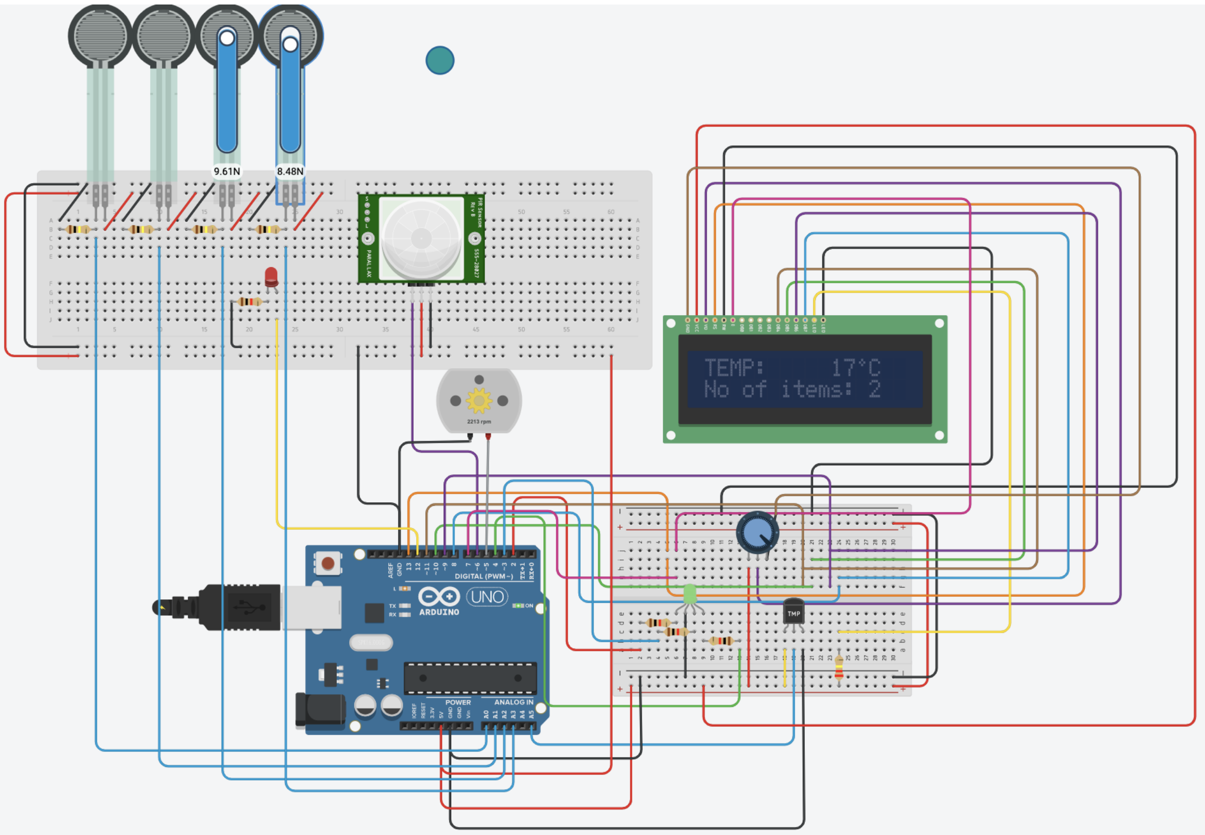
```
#include <LiquidCrystal.h>
//Initialise all the digital and analog pin values
LiquidCrystal lcd(13, 7, 11, 10, 9, 8);
int fridgeSensor[4]={A0,A1,A2,A3};
float fridgeForceValue[4]={0,0,0,0};
String lcdBuffer;

int LED = 12;//Digital pin 12 for door LED
//Digital Pin 2,3,4 for temp optimization
int LEDR = 2;
int LEDB = 3;
int LEDG = 4;

int PIR = 6;
int sensor = 0;
int temp = 0;
int FAN = 5;
int speed = 0;

void setup()
{
  lcd.begin(16, 2);  //Initializes the interface for the LCD screen
  pinMode(LED, OUTPUT); // Setting the pin for the LED
  // Setting the pins for the RGB LED
  pinMode(LEDR, OUTPUT);
  pinMode(LEDB, OUTPUT);
  pinMode(LEDG, OUTPUT);
  pinMode(PIR, INPUT);// Setting the pin for the PIR sensor
  pinMode(A5, INPUT);//Setting the pin for the temperature sensor
  pinMode(FAN, OUTPUT);// Setting the pin for the DC motor
  // Setting the pins for the Force sensors
  pinMode(fridgeSensor[0],INPUT);
  pinMode(fridgeSensor[1],INPUT);
  pinMode(fridgeSensor[2],INPUT);
  pinMode(fridgeSensor[3],INPUT);
  lcd.print("TEMP: ");
}

void loop()
{
  int count = 0;
  for(int i=0; i<4; i++)//to read input from the four Force sensors
  {
    fridgeForceValue[i]=analogRead(fridgeSensor[i]);
    //if sensor is squeezed, force value is >0 and  item count incremented
    if(fridgeForceValue[i]>0)
    {
        count++;
    }
  }

  if(count>0)   //if fridge empty, do not turn LED lights on
    {
    sensor = digitalRead(PIR); //reading PIR sensor value
    if(sensor==HIGH) //if motion detected, fridge lights on
      digitalWrite(LED, HIGH);
    else
      digitalWrite(LED, LOW);
    }
    temp = analogRead(A5); //reading temperature sensor value
    //To map the input from the temperature sensor to the appropriate temperature values
    temp=map(temp, 20, 358, -40, 125);
    if(temp>30)
    {
        //red LED for high temperature
        digitalWrite(LEDR, HIGH);//Red indicator on
        digitalWrite(LEDB, LOW);
        digitalWrite(LEDG, LOW);
        analogWrite(5, 150); //fan speed slightly high
    }
    else if(temp<12)
    {
        //blue LED for low temperature
        digitalWrite(LEDR, LOW);
        digitalWrite(LEDB, HIGH);//Blue indicator on
        digitalWrite(LEDG, LOW);
        analogWrite(5, 50); //fan speed low
    }
    else
    {
      //green LED for optimum temperature
        digitalWrite(LEDR, LOW);
        digitalWrite(LEDB, LOW);
        digitalWrite(LEDG, HIGH);//Green indicator on
        analogWrite(5, 100); //fan speed optimal
    }
    if(temp > 90 )
    {
        analogWrite(5, 255); //fan speed very high
    }
    else if(temp > 60 )
    {
        analogWrite(5, 200); //fan speed reasonably high
    }
    // print temp detected in the temperature sensor
    String output = String(temp)+ String((char)178) + "C ";
    lcd.setCursor(10,0);
    lcd.print(output);
    lcd.setCursor(0,1); //set cursor to line 2
    //print number of items inside the fridge
    lcd.print("No of items: ");
    lcd.setCursor(13,1);
    lcd.print(count);
    delay(10); //delay after which loop() function is executed again.
}
```

SCREENSHOTS OF THE OUTPUT:

Optimum Case:

There are 2 water bottles detected in the fridge, and the temperature is optimum (17°C), thus the RGB is green. The inventory and temperature is printed on the LCD. The cooling system works at normal range, depicted by a DC motor running at 2213 rpm.

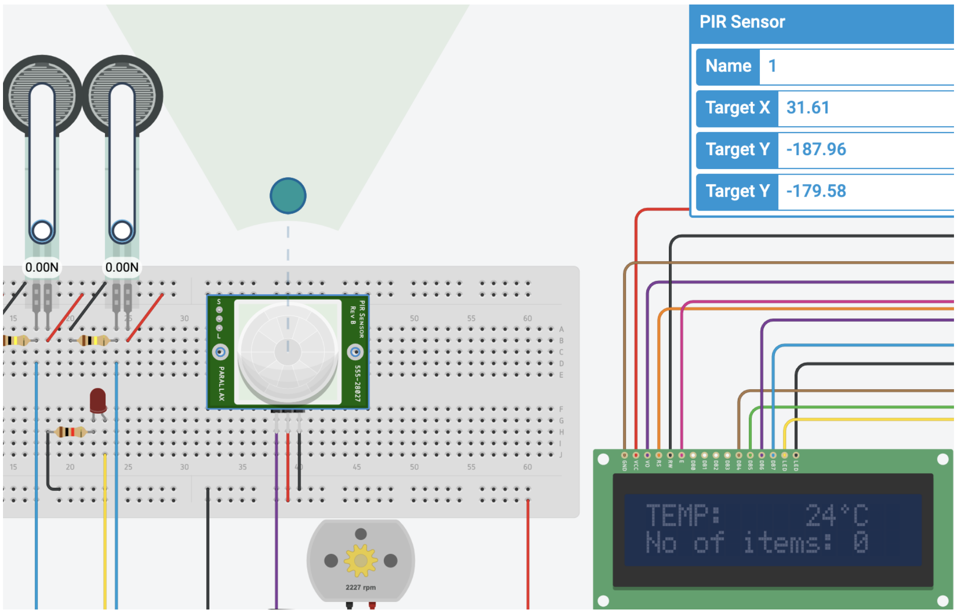
The user's motion has been detected and as the refrigerator is non-empty, the red light (indicating the fridge's internal light) is switched on.



```
TEMP:        17°C
No of items: 2
```

Empty Fridge Case:

There are zero bottles detected inside, so in order to save power the fridge inner light wont turn on even if the PIR sensor detects motion. Thus the Red LED is switched off.



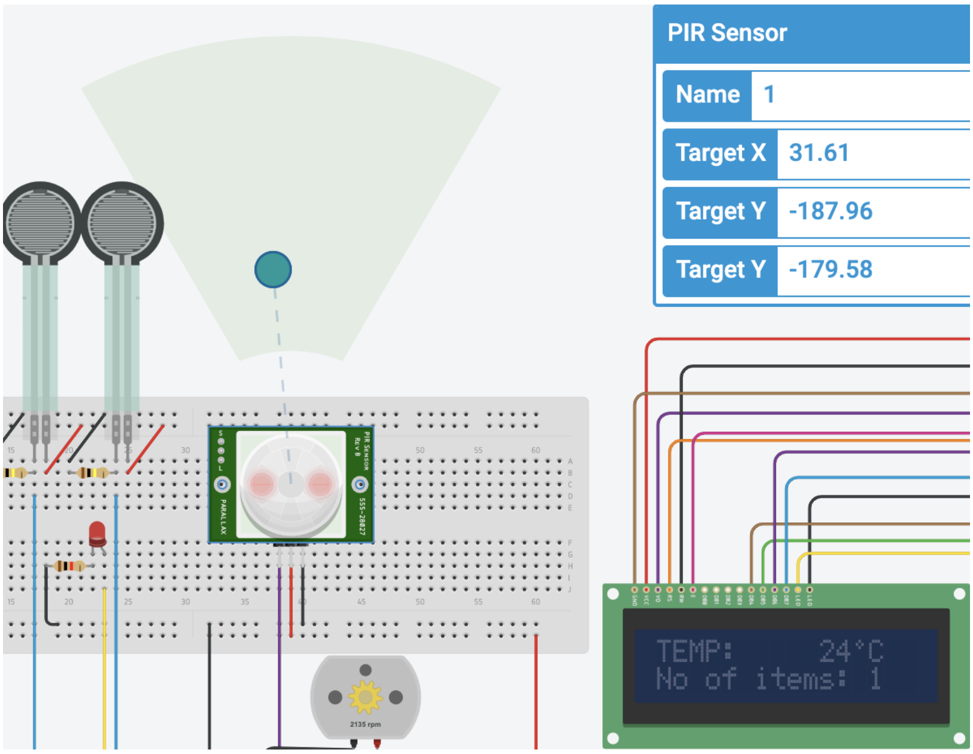
PIR Sensor

Name	1
Target X	31.61
Target Y	-187.96
Target Y	-179.58

```
TEMP:       24°C
No of items: 0
```

Non-Empty Fridge Case:

Bottles detected inside the fridge, so the lights will turn on. Red LED switched on.



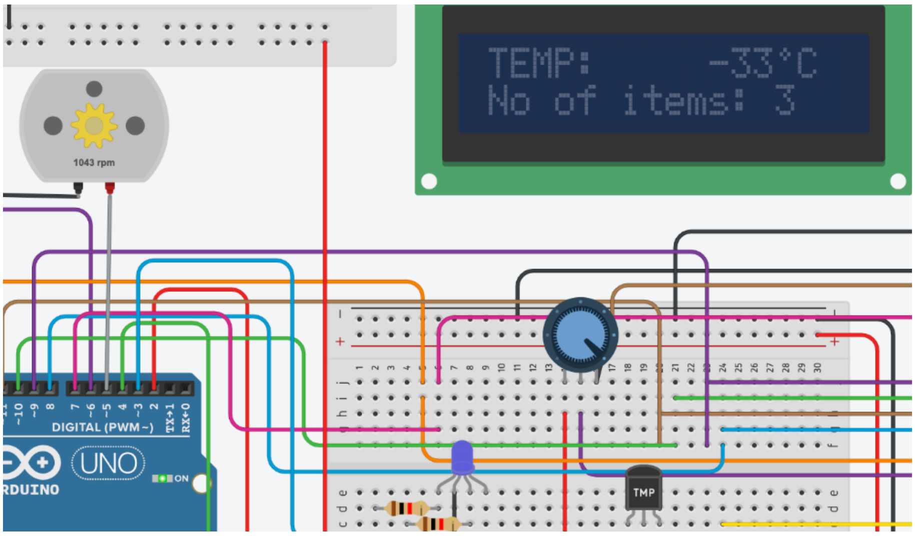
PIR Sensor

Name	1
Target X	31.61
Target Y	-187.96
Target Y	-179.58

```
TEMP:       24°C
No of items: 1
```

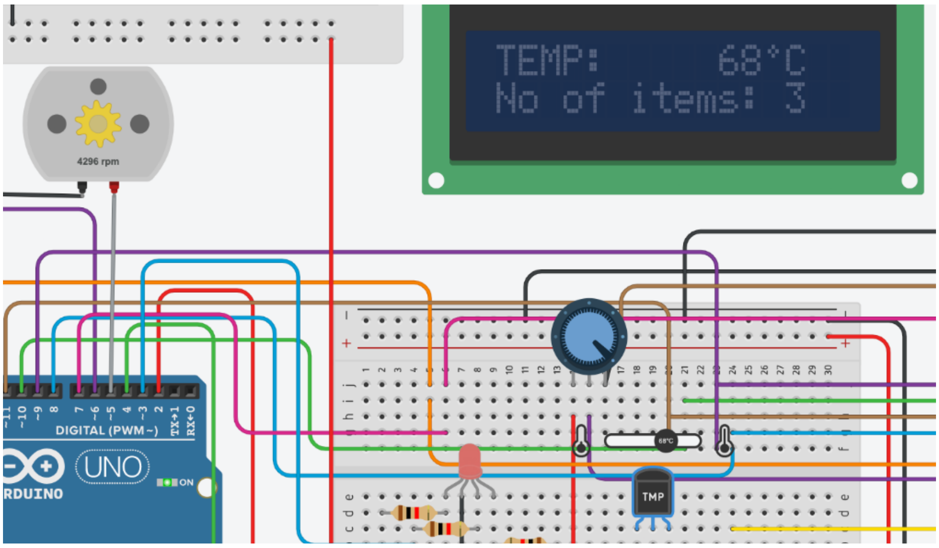
Low Temperature Case:

As the temperature is extremely low(-33°C), the RGB indicator turns Blue and the cooling system is reduced i.e. the DC motor speed has been decreased to 1043 rpm.



High Temperature Case:

As the temperature has risen drastically (68°C), the RGB indicator turns Red and the cooling system has been increased i.e. the DC motor speed is now 4296 rpm.



PROJECT LINK:

https://www.tinkercad.com/things/g0oQyuGMDjZ-mpcaprojectv1/editel?sharecode=k8Cn1roxKejHFDiixsEgRVtS1JB290ZQjMeyRHvaPhk

REFERENCES:

- https://arduinogetstarted.com/tutorials/arduino-force-sensor
- https://www.tutorialspoint.com/arduino/arduino_pir_sensor.htm
- https://www.tutorialspoint.com/arduino/arduino_temperature_sensor.htm
- https://arduinogetstarted.com/tutorials/arduino-lcd
- https://www.instructables.com/How-to-use-an-LCD-displays-Arduino-Tutorial/
- https://arduinogetstarted.com/tutorials/arduino-dc-motor
- https://create.arduino.cc/projecthub/Guitarman1/displaying-sensor-values-on-lcd-c0c44f
- https://www.instructables.com/Interfacing-Force-Sensitive-Resistor-to-Arduino/