INMOVIDU IOT, ROBOTICS AND EMBEDDED SYSTEM INDUSTRIAL TRAINING B1 MAY 2021

Project Report on:

"Arduino Based Automatic Temperature Control System"

SUBMITTED BY: KUNAL VERMA

SECOND YEAR

EE UNDERGRADUATE

INDIA INSTITUE OF TECHNOLOGY ROPAR

EMAIL: 2020EEB1182@iitrpr.ac.in

INDEX

ABSTRACT

This project involves the design and implementation of an intelligent temperature control system. The objective is to design a system that can maintain temperature in a specified range using a heater and a fan without any external influence. Based on what the temperature is with respect to the specified range the system decides weather to use the fan or the heater while printing the status on an LCD screen.

The implementation is done using an Arduino, a DC motor, TMP36 temperature sensor, a 16X2 LCD, a 250kohm potentiometer, red LED, L293D motor driver and a 9V battery.

The Arduino acts as the controller of the system. It processes the sensor data and controls all other processes.

The TMP39 temperature sensor is used to measure the temperature.

The DC motor with the L239D motor driver and 9V battery acts as a fan for the system.

The red LED acts as heater for the system.

INTRODUCTION

With the advancement of technology, intelligent system is introduced every day. Everything is getting sophisticated day by day. There is an increase demand of technology and smart electronic system. Electric fan is one of the most popular electronic devices due to its cost effectiveness and low power consumption advantages. It is one of the most sensible to offer a comfortable and energy efficient. The fan has been long used and still available in the market.

Demand for the accurate temperature control has conquered many of industrial domains. Automatic temperature control is important in order to maintain a comfortable environment. Automation system aims to further the cause of automation to achieve the goal of simplicity. In primitive way of using a fan by button the switch and by controlling the speed controller, there is no chance that the temperature can change the speed of fan except changing the speed of fan by manually. So, it is needed to create an automatic temperature control system the electric fan to automatically change the speed level according to temperature changes and develop an automatic fan system that can change the speed level due to the environment temperature changes.

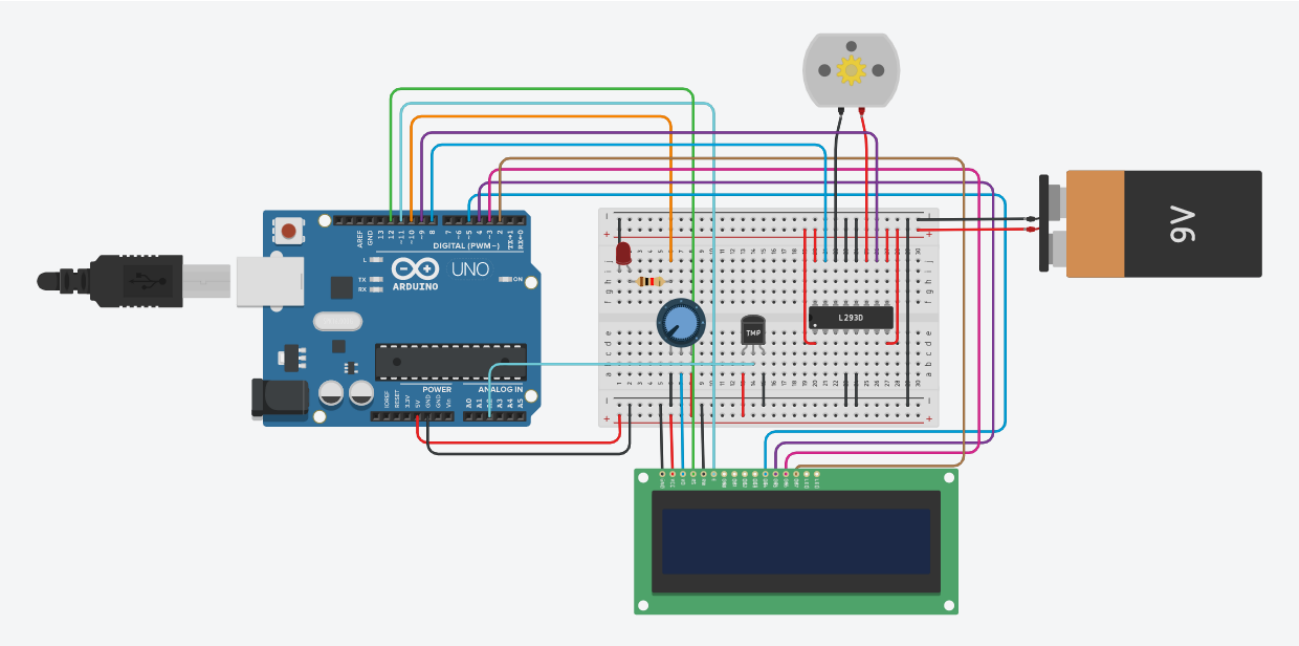
Automatic temperature-controlled fan leverages the power of Arduino to provide an automation control system of fan which speed is controlled by measuring temperature by itself. Using Arduino, motor, jumper wires, sensors and other hardware the fan measure the surrounding temperature and on depending that control the speed of fan automatically. It is one practical use is to integrate a microcontroller in a temperature control system that can be used for automatically controlling a room temperature by controlling the fan's speed automatically.

METHODOLOGY

In this project we have created a working simulation of the said project on tinker cad. This involves two main parts, making the hardware connections and coding the Arduino. Let's start by discussing how the hardware connections are made.

Hardware Connections

We are using an Arduino, breadboard, a DC motor, L239D Motor Driver, a 9V dc battery, a TMP36 temperature sensor, a red LED, a 16X2 LCD, 1Kohm resistor and a 250Kohn potentiometer.



Connections for the automatic temperature control system

Let us first connect the LCD. Connect the ground and VCC to the ground and 5V of the Arduino respectively. Connect the Vo (contrast pin) to the wiper of the potentiometer while connecting the power and ground of the potentiometer to the 5V and ground of the Arduino respectively. Connect the RS (register select) and E (enable pin) to the digital pins. Connect the read/write to the ground of the Arduino. Connect the DB4, DB5, DB6 and DB7 to the digital pins.

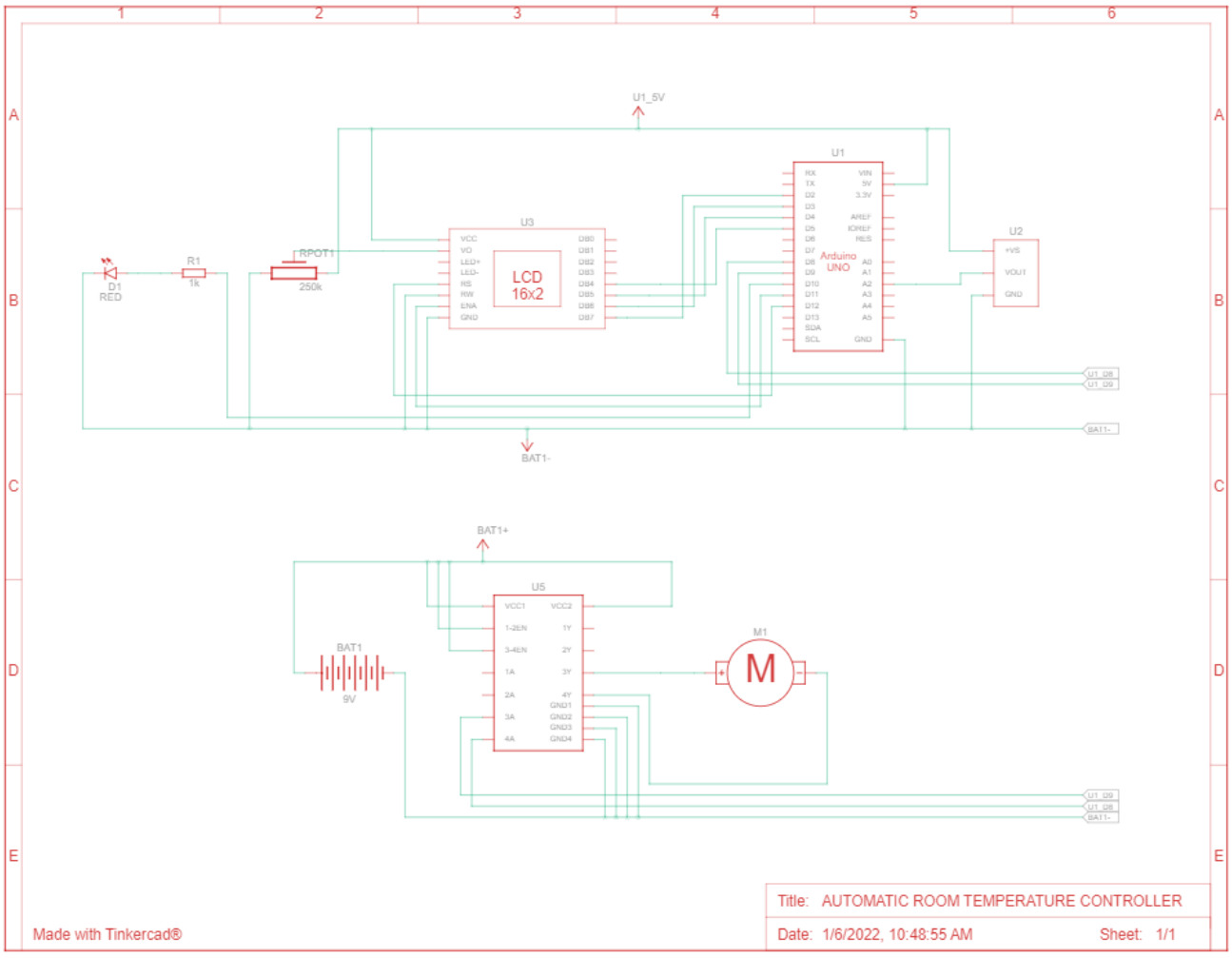
Now, let's connect the L293D and the DC motor with the 9V battery. Connect the grounds of the L239D, the negative terminal of the 9V battery and the ground of the Arduino together to create a common ground. Connect the power and enable terminal of both sides of the motor

driver with the 9V battery. Connect the output pin to the DC motor and the input pins to the digital pins. This completes the motor connection.

Now connect the led cathode to the ground and the cathode to a digital pin through a 1k ohm resistor.

Let's connect the TMP36 temperature sensor now. Connect its ground and power terminal to the ground and 5V of the Arduino and the vout to the analog pin.

With this the connections are complete. Here is the circuit diagram of the model:



Coding the Arduino

First, we include libraries and declare all the useful variables we are going to use in the code like pin names, delay times and some variables to store values.

```
#include <LiquidCrystal.h>

int rs = 12;
int en = 11;
int db4 = 5;
int db5 = 4;
int db6 = 3;
int db7 = 2;
int led = 10;
int Mt1 = 9;
int Mt2 = 8;
int Mdelay = 10;
int dt = 3000;
int tempread = A2;
float sensorread;
float eqvolt;
float temp;
float Maxtemp = 25.00;
float Mintemp = 20.00;
int fanstat = 0;
int ledstat = 0;

LiquidCrystal Lcd(rs,en,db4,db5,db6,db7);
```

First, we include the library for using the LCD called LiquidCrystal.h.
Now , rs, en, db4, db5, db6 and db7 represents the register select, the enable pin , DB4, DB5, DB6 and DB7 of the LCD.
Mt1 and Mt2 are for L239D input terminals and tempread is for vout of the temperature sensor.
Maxtemp and Mintemp are the max temperature and min temperature of the specified range which is 20-25 here.
Fanstat and ledstat are used to keep track of the fan and led being on or off.
Rest are used to store useful data and delay times.
In the end we declare a variable Lcd of the class LiquidCrystal with rs ,en, db4 ,db5, db6, db7 pins.

Now we need to configure our pins

```
void setup(){
 Lcd.begin(16,2);
 pinMode(led,OUTPUT);
 pinMode(tempread,INPUT);
 pinMode(Mt1,OUTPUT);
 pinMode(Mt2,OUTPUT);
 digitalWrite(Mt2,LOW);
}
```

We configure the LCD using the Lcd variable and begin function with its dimensions as the argument. We configure the led, Mt1 and Mt2 pins as output with PWM and the tempread pin as input.
Then we make the Mt2 pin 0V for the rest of the code.

Now we code the loop of the Arduino

```
void loop(){
 sensorread = analogRead(tempread);              //measuring temperature
 eqvolt = sensorread*(5.00/1023.00);
 temp = 100.00*eqvolt-50.00;

 Lcd.clear();                                    //printing temperature readings on LCD
 Lcd.print("Sensor Reading:");
 Lcd.setCursor(2,1);
 Lcd.print(temp);
 Lcd.print(" C");
 delay(dt);

 if(temp >= Maxtemp){                            // temp greater than max temperature
  if(ledstat == 1){
   Lcd.clear();
   Lcd.print("It's too Hot!!");
   Lcd.setCursor(1,1);
   Lcd.print("Turn off Heater");
   digitalWrite(led,LOW);
   ledstat = 0;
   delay(dt);
  }
  Lcd.clear();
  Lcd.print("It's too Hot!!");
  Lcd.setCursor(2,1);
  if(fanstat == 0){
```

```
   Lcd.print("Turn on Fan");
   for(int i=0; i<=255; i++){
    analogWrite(Mt1,i);
    delay(Mdelay);
   }
   fanstat = 1;
  }
  else if(fanstat == 1){
   Lcd.print("Fan is On");
  }
  delay(dt);
 }
 else if(temp <= Maxtemp && temp >= Mintemp){                    //temp in specified range
  Lcd.clear();
  Lcd.print("Temp is Normal");
  Lcd.setCursor(2,1);
  if(fanstat == 1){
   Lcd.print("Turning off Fan");
   fanstat = 0;
   for(int i=255; i>=0; i--){
        analogWrite(Mt1,i);
        delay(Mdelay);
   }
  }
  else if(ledstat == 1){
   Lcd.setCursor(1,1);
   Lcd.print("Turn off Heater");
   ledstat = 0;
   digitalWrite(led,LOW);
  }
  else{
   Lcd.print("All are Off");
  }
  delay(dt);
 }
 else if(temp <= Mintemp){                                  //temp less than minimum temperature
  if(fanstat == 1){
   Lcd.clear();
   Lcd.print("It's too Cold!!");
   Lcd.setCursor(2,1);
   Lcd.print("Turn off Fan");
   fanstat = 0;
   for(int i=255; i>=0; i--){
        analogWrite(Mt1,i);
        delay(Mdelay);
   }
   delay(dt);
```

```
   }
   Lcd.clear();
   Lcd.print("It's too Cold!!");
   Lcd.setCursor(2,1);
   if(ledstat == 0){
    Lcd.print("Turn On Heater");
    ledstat = 1;
    digitalWrite(led,HIGH);
   }
   else if(ledstat == 1){
    Lcd.print("Heater is On");
   }
   delay(dt);
  }
  else{                                                    //error
   Lcd.clear();
   Lcd.print("Something Went Wrong");
   Lcd.setCursor(2,1);
   Lcd.print("Error in the Circuit");
   delay(dt);
  }
}
```

So first we measure the temperature using the TMP26 temperature sensor. We use the analog read to read the input from the vout of the sensor and convert it into equivalent voltage. Based on the data sheet and the equivalent voltage obtained we calculate the temperature from the sensor.
Now we clear the screen and print "Sensor Reading" then set the cursor at (2,1) and print the temperature and its unit. Then we wait for dt time.

Now if the temperature is greater than maximum temperature of the specified range and the led (heater) is already on, print "It's too Hot!!", set the cursor at (1,1) and print "Turn off Heater" and turn the led off. Now set the ledstat to 0 to signify that the led if off.
If the fan is off then print "It's too Hot!!", set the cursor at (2,1) and print "Turn on Fan" then slowly start the motor using analogwrite. Make fanstat as 1 to signify that the fan is on now.
If the fan is already on then print "It's too Hot!!", set the cursor at (2,1) and print "Fan is on".

Now if the temperature is in the specified range then print "Temp is Normal" and set the cursor at (2,1). If the fan is on print "Turning off Fan", make the fanstat 0 and turn off the motor slowly. If the led is on then, print turn off Heater and set the ledstat as 0. If both led and fan are off print "All are Off".

Now if the temperature is less than the minimum temperature and the fan is on print "Its too Cold!!", set the cursor ate (2,1) and print "Turn off Fan" and make fanstat as 0 then slowly turn off the motor. If fan is not on, print "Its too cold!!" and set the cursor at (2,1) now

if led is off print "Turn on Heater" and set ledstat as 1, then turn on the led. If the led is already on print "Heater is On".
If there is some unknow error it should print "Something went wrong" and "Error in the circuit" in the next line.

This concludes the code.

CODE

```
#include <LiquidCrystal.h>

int rs = 12;
int en = 11;
int db4 = 5;
int db5 = 4;
int db6 = 3;
int db7 = 2;
int led = 10;
int Mt1 = 9;
int Mt2 = 8;
int Mdelay = 10;
int dt = 3000;
int tempread = A2;
float sensorread;
float eqvolt;
float temp;
float Maxtemp = 25.00;
float Mintemp = 20.00;
int fanstat = 0;
int ledstat = 0;

LiquidCrystal Lcd(rs,en,db4,db5,db6,db7);

void setup(){
 Lcd.begin(16,2);
 pinMode(led,OUTPUT);
 pinMode(tempread,INPUT);
 pinMode(Mt1,OUTPUT);
 pinMode(Mt2,OUTPUT);
 digitalWrite(Mt2,LOW);
}

void loop(){
 sensorread = analogRead(tempread);          //measuring temperature
 eqvolt = sensorread*(5.00/1023.00);
 temp = 100.00*eqvolt-50.00;

 Lcd.clear();                                //printing the temperature readings on LCD
 Lcd.print("Sensor Reading:");
 Lcd.setCursor(2,1);
 Lcd.print(temp);
 Lcd.print(" C");
```

```
delay(dt);

if(temp >= Maxtemp){                          //temp is greater than max temperature
 if(ledstat == 1){
  Lcd.clear();
  Lcd.print("It's too Hot!!");
  Lcd.setCursor(1,1);
  Lcd.print("Turn off Heater");
  digitalWrite(led,LOW);
  ledstat = 0;
  delay(dt);
 }
 Lcd.clear();
 Lcd.print("It's too Hot!!");
 Lcd.setCursor(2,1);
 if(fanstat == 0){
  Lcd.print("Turn on Fan");
  for(int i=0; i<=255; i++){
   analogWrite(Mt1,i);
   delay(Mdelay);
  }
  fanstat = 1;
 }
 else if(fanstat == 1){
  Lcd.print("Fan is On");
 }
 delay(dt);
}
else if(temp <= Maxtemp && temp >= Mintemp){          //temp in specified range
 Lcd.clear();
 Lcd.print("Temp is Normal");
 Lcd.setCursor(2,1);
 if(fanstat == 1){
  Lcd.print("Turning off Fan");
  fanstat = 0;
  for(int i=255; i>=0; i--){
        analogWrite(Mt1,i);
        delay(Mdelay);
  }
 }
 else if(ledstat == 1){
  Lcd.setCursor(1,1);
  Lcd.print("Turn off Heater");
  ledstat = 0;
  digitalWrite(led,LOW);
 }
 else{
```

```
    Lcd.print("All are Off");
   }
   delay(dt);
  }
  else if(temp <= Mintemp){                    //temp less than minimum temperature
   if(fanstat == 1){
    Lcd.clear();
    Lcd.print("It's too Cold!!");
    Lcd.setCursor(2,1);
    Lcd.print("Turn off Fan");
    fanstat = 0;
    for(int i=255; i>=0; i--){
          analogWrite(Mt1,i);
          delay(Mdelay);
    }
    delay(dt);
   }
   Lcd.clear();
   Lcd.print("It's too Cold!!");
   Lcd.setCursor(2,1);
   if(ledstat == 0){
    Lcd.print("Turn On Heater");
    ledstat = 1;
    digitalWrite(led,HIGH);
   }
   else if(ledstat == 1){
    Lcd.print("Heater is On");
   }
   delay(dt);
  }
  else{                              //error
   Lcd.clear();
   Lcd.print("Something Went Wrong");
   Lcd.setCursor(2,1);
   Lcd.print("Error in the Circuit");
   delay(dt);
  }
}
```

CONCLUSION

At the end of this project we are able to create an automatic temperature controller which can turn on/ off a fan or heater based on the temperature read by the temperature sensor with respect to the specified range.

In this project if the temperature is higher than the maximum temperature of the specified range then we turn on the fan a turn off the heater if it is on. Unless the temperature is back to normal the fan keeps blowing. Once it reaches the normal temperature range everything is turned off.

Similarly, if the temperature is less than the minimum specified temperature then we turn off the fan if its on and turn on the heater. If the heater is already on we keep it on unless temperature is back to normal. Once it reaches the normal temperature range everything is turned off.

The temperature, fan and led status is constantly displayed on the LCD.

In case there is some error, an error message is printed on the LCD.

The link of the project is given below:
https://www.tinkercad.com/things/lXlovESZ4Zt

END OF REPORT