The Temperature Box: An Introductory Control Systems Project

Introduction:

The purpose of this project is to build and understand the function of the closed-loop temperature control system. The primary components of this project are a lightbulb, temperature sensors (TMP36, DS18B20), a cardboard box that serves as a plant, and a relay that functions as an actuator. Additionally, an Op-Amp-based comparator circuit is present. A scale and shift circuit is also included for handling signal conditions. The primary feature, which is the dynamic control of a lightbulb's on/off state within a specified temperature range, is made possible by the two temperature sensors (TMP36 and DS18B20). These sensors compare the measured temperature to pre-established low and high-temperature thresholds using digital logic and op-amps to provide accurate temperature comparisons.

Objective:

The purpose of this project is to investigate temperature control system design, implementation, and evaluation, with an emphasis on both analog and digital controller designs. We will create and evaluate two different control strategies using Proteus software and two different temperature sensors, the TMP36 and the DS18B20. By combining components like digital logic circuits, op-amps, and Arduino microcontrollers, we aim to evaluate the effectiveness, economy, and efficiency of each control strategy in the closed-loop Temperature Box experiment. As we consider real-world constraints and considerations, especially about analog and digital control paradigms, we strive to identify the optimal control strategy using rigorous testing and comparative analysis of observed outcomes against anticipated results. This helps to advance our understanding of practical control system applications.

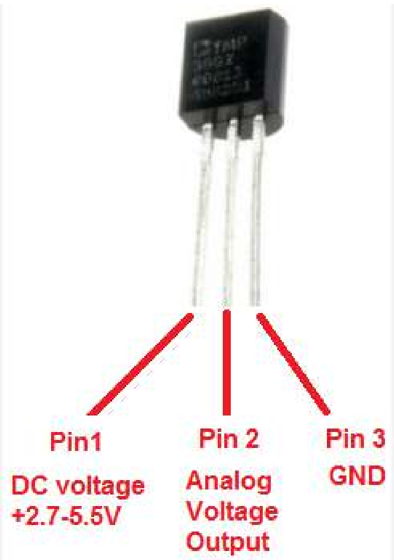
Components:

- Senros: TMP36, DS18B20
- Operational Amplifier (741)
- Resistors
- Voltmeter
- Voltage Source
- Relay & Bulb module
- Logic Gates (NAND, NOT)
- Potentiometer

Sensor Specification:

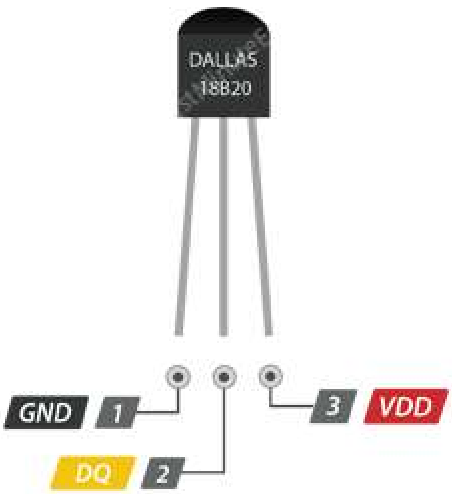
TMP36:

The TMP36 is a low-voltage, precision centigrade temperature sensor. It provides a voltage output that is linearly proportional to the Celsius temperature. It also doesn't require any external calibration to provide typical accuracies of ±1°C at +25°C and ±2°C over the −40°C to +125°C temperature range. The TMP36 temperature sensor employs a solid-state technique for temperature determination, eschewing traditional mercury, bimetallic strips, or thermistors. Instead, it capitalizes on the inherent property that as temperature rises, the voltage across a diode increases at a known rate, specifically the voltage drop between the base and emitter (Vbe) of a transistor. By accurately amplifying this voltage change, the TMP36 delivers a linear voltage output corresponding to Celsius temperature, obviating the need for external calibration.

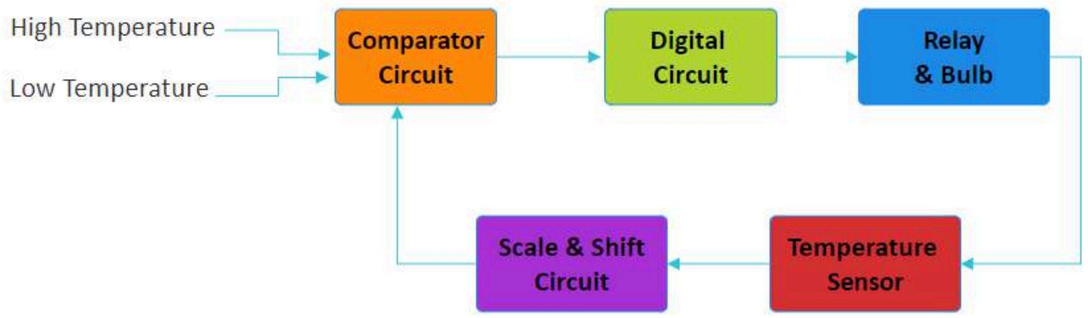


DS18B20:

The DS18B20 is one of the simplest digital temperature sensors. Operating within a wide temperature range from -55°C to +125°C, this sensor offers temperature readings with selectable resolutions ranging from 9 to 12 bits. Its adoption of a single-wire communication protocol simplifies wiring and reduces the required number of pins for seamless integration into systems. Additionally, the DS18B20 features a distinct 64-bit serial code for precise sensor identification, enabling efficient multi-sensor network addressing. Its applicability spans various domains, including consumer electronics, industrial automation, and environmental monitoring, thanks to its low power consumption and diverse packaging options. With its digital interface, the DS18B20 seamlessly interfaces with microcontrollers and digital systems, eliminating the need for analog-to-digital conversion and enhancing compatibility across platforms.



Flow Chart:



High Temperature
Low Temperature
Comparator Circuit
Digital Circuit
Relay & Bulb
Temperature Sensor
Scale & Shift Circuit

Calculation:

Required calculations for Dallas and TMP36:

Let,

the minimum voltage V_L is 30 ℃

the maximum voltage V_H 35℃

Let,

T_c= temperature in Celsius

So,

$V_S = 0.5 + \frac{T_c}{100} = (0.5 + 0.01T_c)V$

This 0.5 is the voltage output when the temperature is 0℃

The scale & shift circuit takes V_s as input and gives V_T as output,

So,

the V_T is designed in such a way that it converts V_S = 0.8V for 30°C and V_S= 0.85V for 35°C to a different voltage value.

Let,

We want to convert V_S = 0.8V to V_T = 3.6V and V_S = 0.85 to V_T = 4.5V

Given,

$V_T = \alpha + \beta V_s$

$3.6 = \alpha + \beta(0.80)$ ....(i)

$4.5 = \alpha + \beta(0.85)$.......(ii)

Solving equation (i) & (ii),

$\alpha = -10.8$

$\beta = 18$

We know,

$\alpha = -\frac{5R}{R_2}$

$\beta = \frac{R}{R_1}$

So,

$R_2 = 4.62 \approx 4.7\Omega$

$R_1 = 0.55 \approx 560\Omega$

Subsystems of the Analog Circuit:

Scale and Shift:

Mapping the TMP36 temperature sensor's output to a desired range requires the scale and shift subsystem. The output voltage V_S of the TMP36 sensor is 0.80V to 0.85V within a temperature range of 30 to 35 °C. The Comparator Circuit cannot precisely detect this 50 mV difference. V_S undergoes scaling and shifting in order to become V_T (Temperature Voltage), which fixes this limitation. This guarantees that at 30 degrees Celsius and 35 degrees Celsius, respectively, V_T will reach a value of 3.6 V and 4.5 V. This mapping is controlled by the transfer function $V_T = \alpha + \beta V_S$. An op-amp-based inverting amplifier and an adder circuit are used to perform this transformation, establishing the necessary adjustments to V_S for accurate temperature monitoring and control.

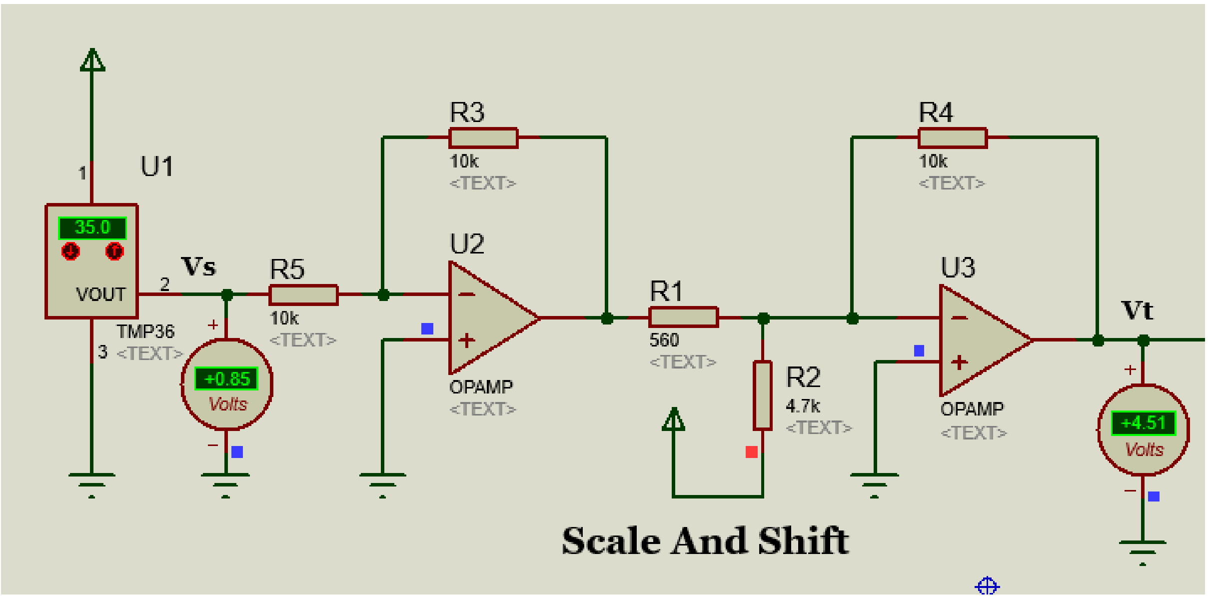


Fig: Scale and Shift Circuit

Comparator Circuit:
A vital component of the system, the comparator circuit determines whether the current temperature is above, below, or in between the designated boundaries of the desired range. This comparison is necessary to understand where the current voltage level is and to trigger the Digital part of the circuit. The first comparator determines whether the current temperature is higher than 35 °C or not by checking whether Vt is higher than 4.5 V. When this condition is satisfied, it gives an output of 15V theoretically and practically around 13.75V. The output of the first comparator is denoted as A. If not, A stays at 0V (OFF). Similarly, the second comparator checks to see if the temperature is lower than 30°C or not, by checking if the Vt is more than 3.6V. Likewise, it also gives an output of 15V theoretically, and 13.75V practically if the condition is met. And it is labeled as B. Two 100-kilo ohm potentiometers are used to generate reference values of 4.5 V and 0.5 V for the comparators to carry out this comparison process.

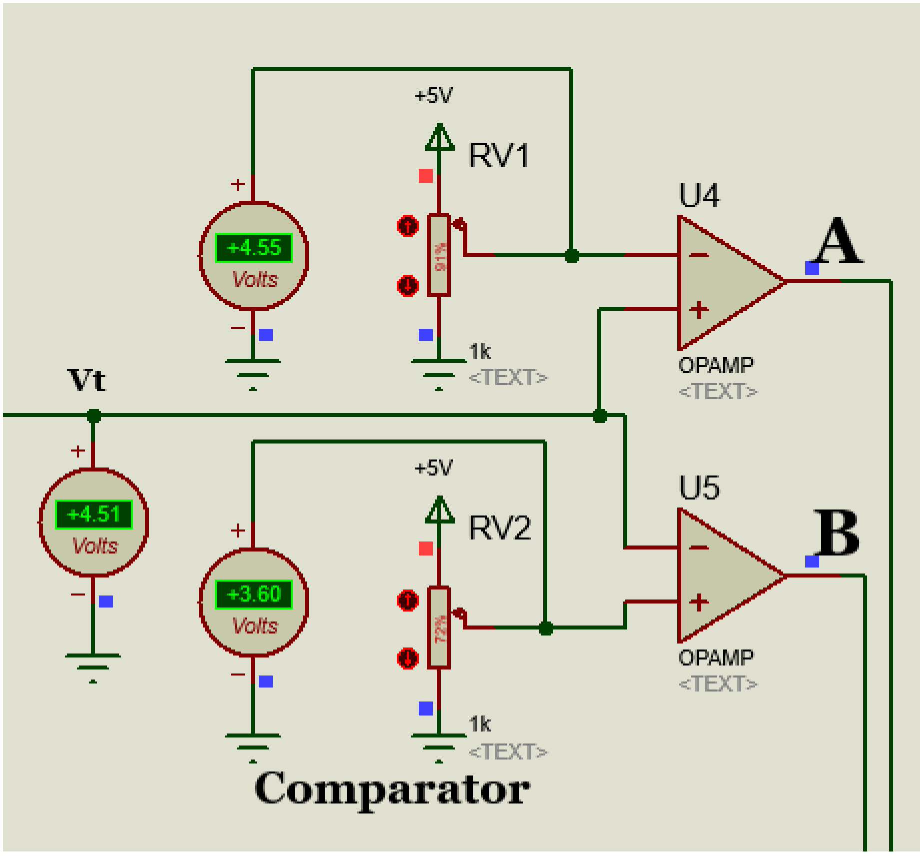


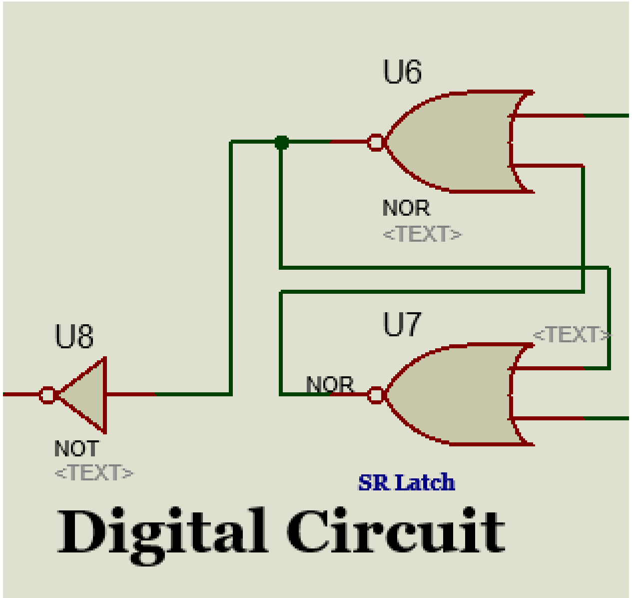
Fig: Comparator Circuit

Digital Circuit:

The digital circuit plays a very important role in determining whether the heating element (Light Bulb) should be activated or deactivated based on the current temperature condition. By analyzing the output of the comparator, we can get 3 possible outcomes,

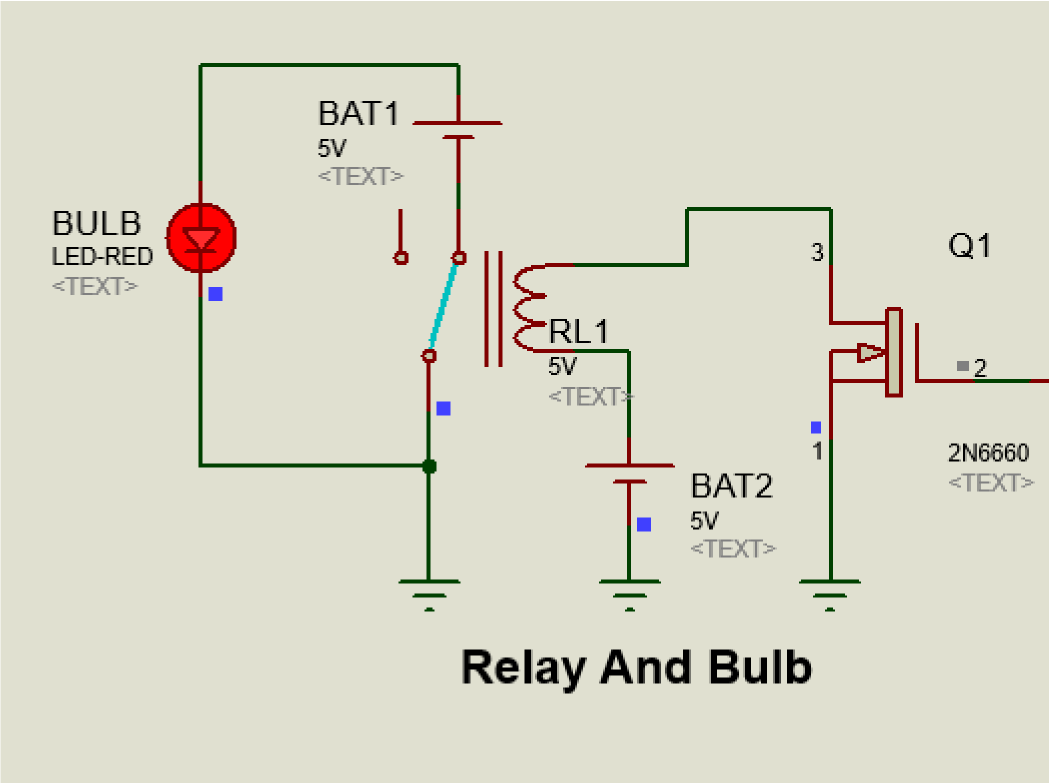
State	A	B	Q	Q'
$V_T<V_L<V_H$	0	1	1	1
$V_L<V_T<V_H$	0	0	1	No change
$V_T<V_H<V_L$	1	0	0	0

From the table we can see that, The logic part follows the SR latch type output method. So we have implemented a SR latch. We needed the Q' output to our Relay to operate. But, in hardware implementation we found out that we are getting comparatively low voltage from Q' output. That is what we have taken the output from the Q side, and through a not gate we will gate the final output, which is required for the relay.



Relay and Bulb:

To regulate the relay which is controlling the heating element, we need to invert the relay's state: turning ON the relay disables the heating element while turning OFF the relay activates it. Nevertheless, to turn on, the relay needs a current of 70 mA across 5V, which the SR-latch is unable to provide directly. Consequently, to supply the required voltage and current, a 5V DC supply is introduced. A MOSFET regulates this power supply, turning the relay ON and OFF in turns. The MOSFET operates utilizing the SR latch's inverted output. The MOSFET must be biased in the saturation region to accomplish this, and this can be done by connecting it to the SR latch's non-inverted output.



Methodology:

Analog Part: Both the TMP36 and Dallas DS18B20 analog circuit works similarly. For simplicity let's describe the methodology of the circuit for TMP36 first. The output of the TMP36 V_S first goes to the scale and shift circuit to be scaled up. From the scale and shift circuit, for 30 ℃ we will get an output of around 3.6V, and for 35 ℃ we will get around 4.5V. Then, this output will go to the comparator circuit. In this part, the comparator circuit will compare the temperature of the sensor (voltage output) of the sensor with the reference value. If the condition is matched, the comparator circuit will give a high output. As there are two comparators to check the sensor output voltage, we will get 4 possible output combinations, and using the SR-latch, and not get we will get the desired feedback from the light bulb.

In the case of the DS18B20 temperature sensor, the overall circuit remains largely unaltered. But because the analog circuit needs an analog signal, the DS18B20's digital output is first sent to an Arduino microcontroller for processing. The Arduino functions as a converter between digital and analog, transforming the digital signal into a suitable analog representation. The scale shift circuit receives this analog signal, which reflects the sensor's temperature data. The remaining parts, which include the relay, MOSFET, and heating element control mechanism, keep working as before, guaranteeing the smooth incorporation of the DS18B20 sensor data into the current setup.

Digital Part: To set up the digital part, we need to connect the TMP36 temperature sensor to an Arduino analog input pin for temperature sensing and link the DS18B20 temperature sensor to a digital pin using the OneWire protocol. We have to connect the heating element (the bulb) to a digital output pin on the Arduino. In the Arduino code, first, we have to read temperature values separately from the TMP36 and DS18B20 sensors. Based on the temperature readings from each sensor, digitally assess heating needs. Activate the heating element by setting the corresponding digital output pin to Low if the temperature reading from one sensor requires heating. Conversely, set the digital output pin to High if heating is not needed. This approach allows for independent temperature-based control of the heating element using the Arduino's built-in capabilities and sensor inputs.

Arduino Codes:

For TMP36:

```
const int TEMP_PIN = A5;
const int RELAY_PIN = 4;

void setup() {
  Serial.begin(9600);
  pinMode(RELAY_PIN, OUTPUT);
}

void loop() {
  int temp = analogRead(TEMP_PIN);
  float mV = temp * 4.9;
  float tempInC = (mV - 500) / 10.0;

  Serial.print("Temperature is: ");
  Serial.print(tempInC);
  Serial.println("Celsius!");

  if (tempInC < 30) {
    digitalWrite(RELAY_PIN, LOW);
    Serial.println("Relay is ON");
  } else if (tempInC > 35) {
    digitalWrite(RELAY_PIN, HIGH);
    Serial.println("Relay is OFF");
  }
  delay(1000);
}
```

For DS18B20:

```
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 4

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

const int relayPin = 7;

void setup() {
  Serial.begin(9600);
  sensors.begin();
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);
}
void loop() {
  sensors.requestTemperatures();
  float temperatureC = sensors.getTempCByIndex(0);

  Serial.print("Temperature: ");
  Serial.print(temperatureC);
  Serial.println("°C");

  if (temperatureC < 30) {
    digitalWrite(relayPin, LOW);
    Serial.println("Relay ON");
  } else if (temperatureC > 35) {
    digitalWrite(relayPin, HIGH);
    Serial.println("Relay OFF");
  }
  delay(1000);
}
```

Design and Simulation:

Circuit Diagrams:

DS18B20 Digital:

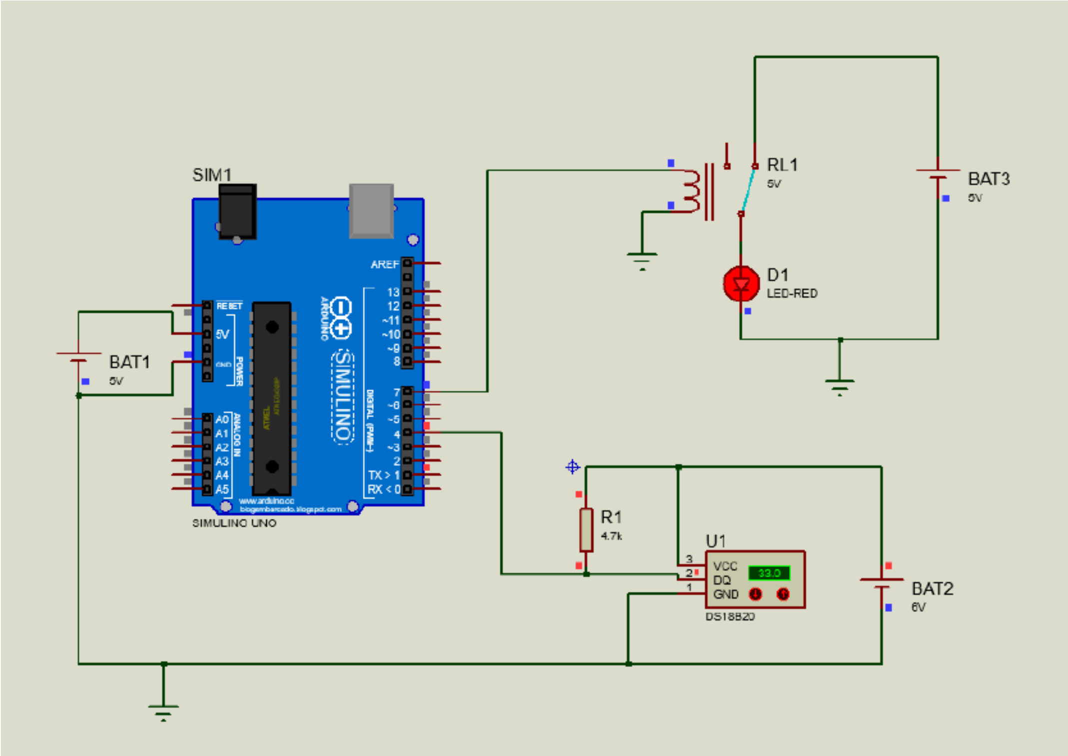


Fig: DS18B20 Digital Softaware.

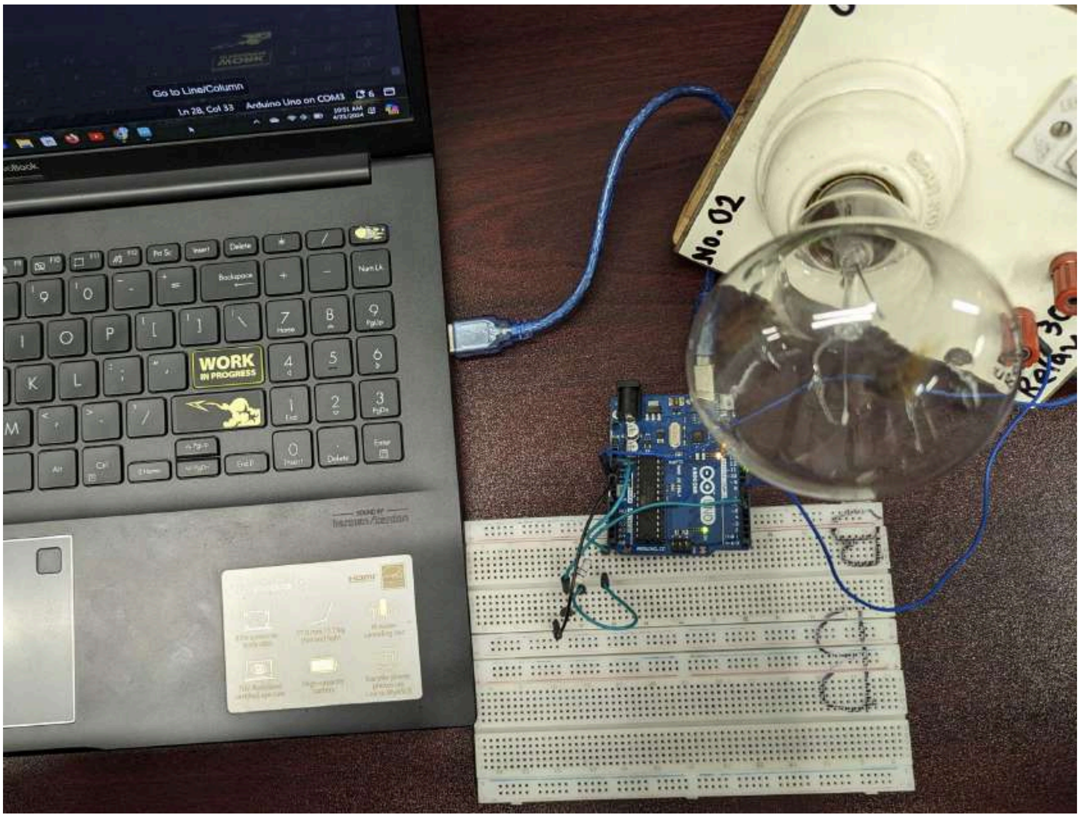


Fig: DS18B20 Digital gardware.

TMP36 Digital:

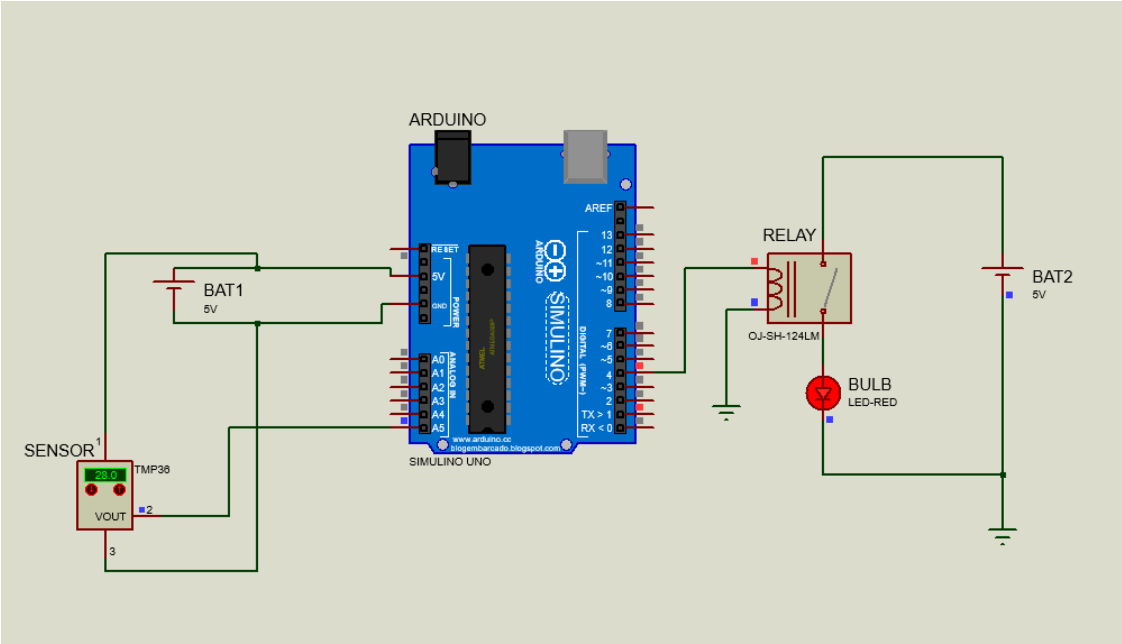


Fig: TMP36 Digital Softaware.

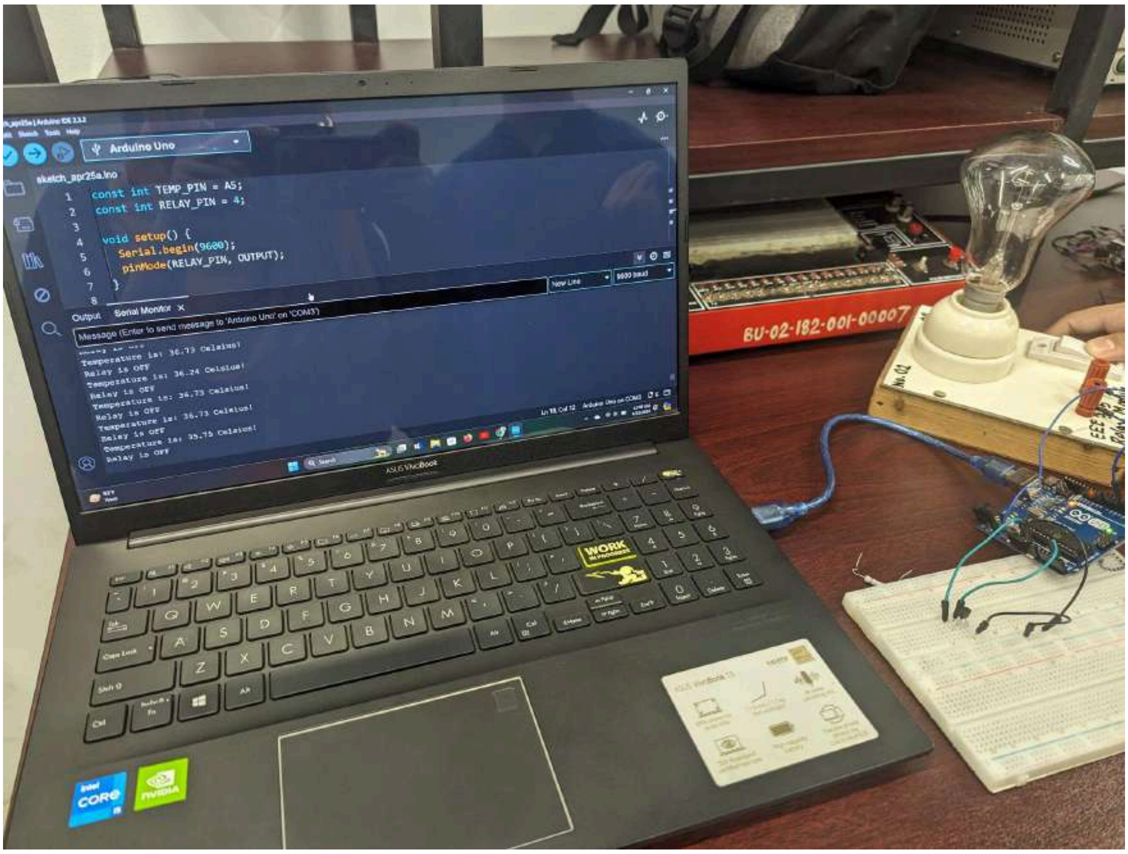


Fig: TMP36 Digital Hardware.

TMP35 Analog:

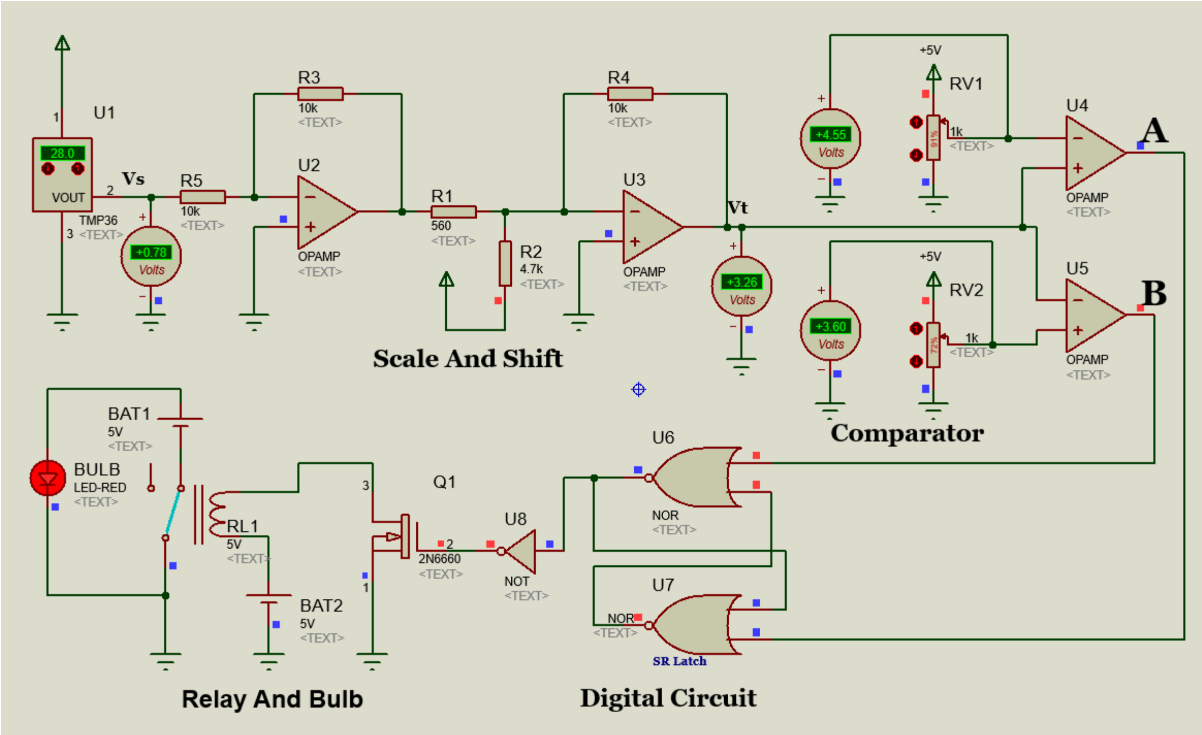


Fig: TMP36 Analog Software.

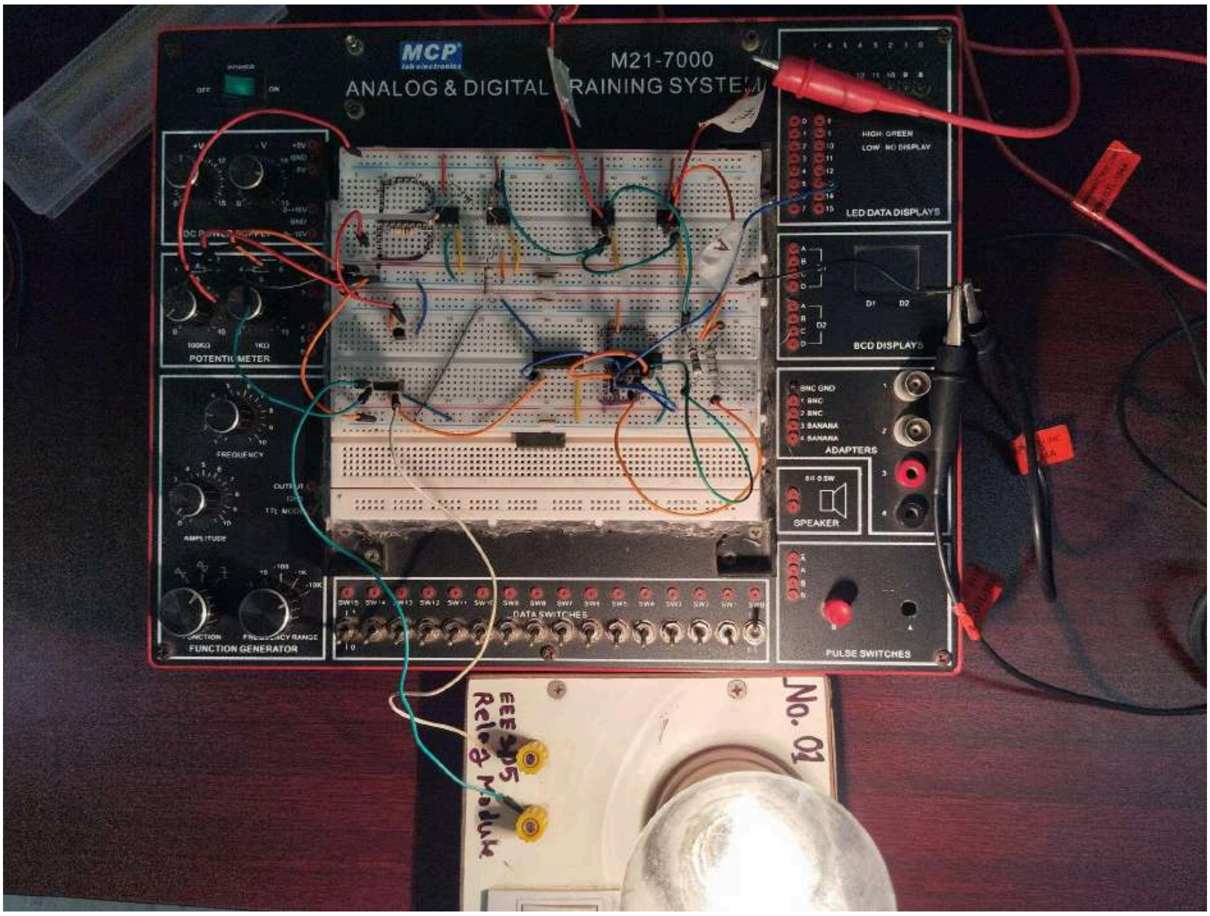


Fig: TMP36 Analog Hardware.

Dallas Analog:

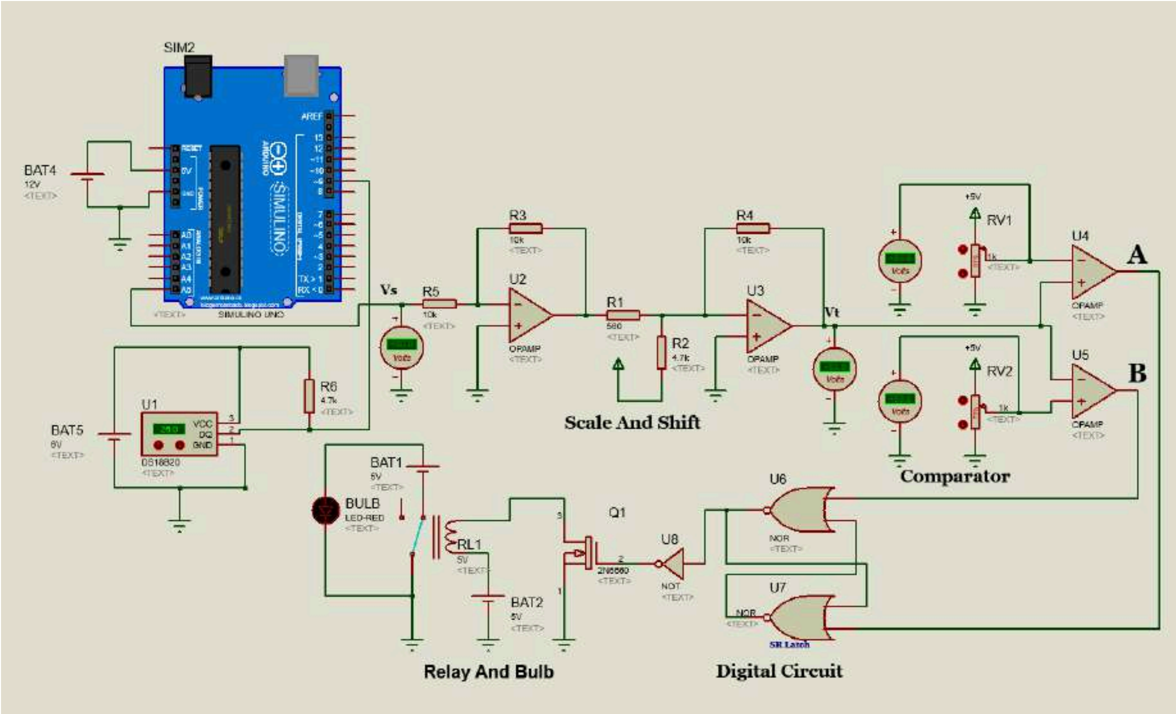


Fig: DS18B20 Analog Software.

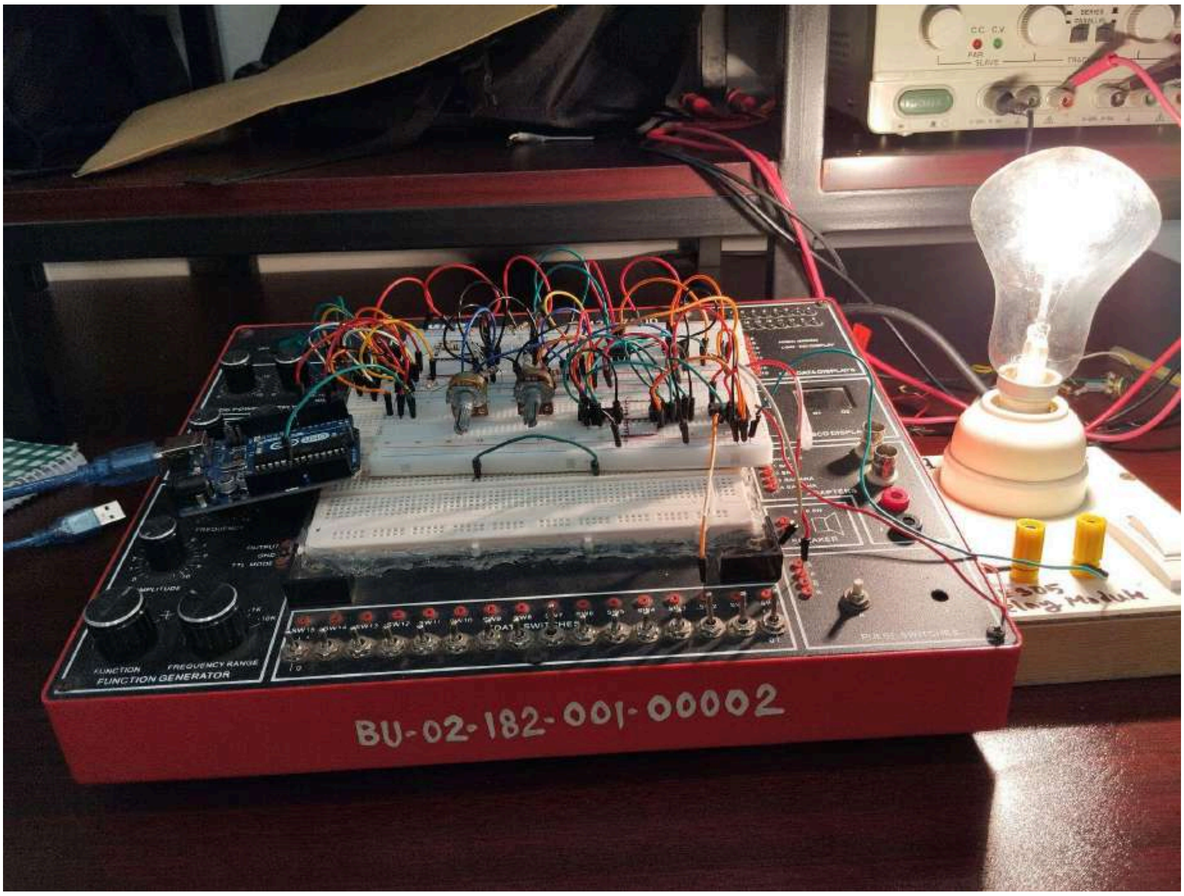


Fig: DS18B20 Analog Hardware.

Result and Analysis:

TMP36 Digital:
According to the input codes, the relay will stay on until the temperature reaches 35° Celsius which can be seen in the figure below. In this circuit, the main part is the Arduino Uno, where the hex of the code is being uploaded, in which the condition for the on/off stage change is given. Here using the TMP36, the temperature is varied and the LED remains on until the temperature hits 35 degrees Celsius, after that the state changes, LED turns off.

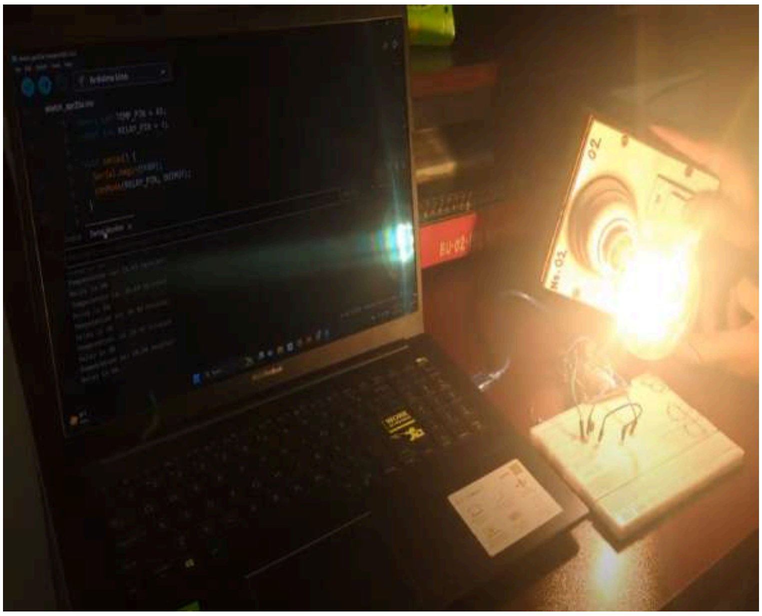


Fig: TMP36 Hardware setup below 30° C

As soon as the temperature hits 35° Celsius, the bulb turns off and it remains unchanged until the temperature goes below 30° Celsius. This can be seen in the figure below:

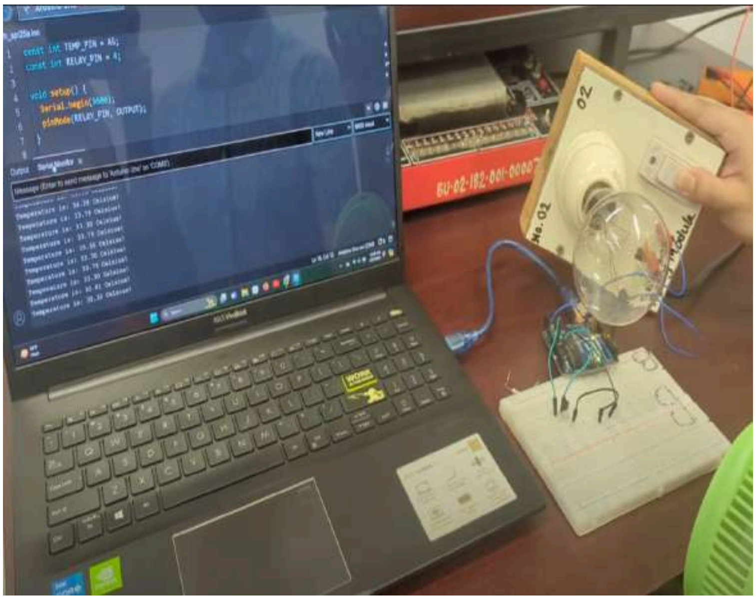


Fig: TMP36 Hardware setup above 35° C

Here V_L = 30° Celsius and V_H = 35° Celsius. When V_T (Scale and shift output) is in between the range of V_L, and V_H the circuit state will remain unchanged and the Active low voltage will light up the bulb connected to the relay.(while we are increasing the temperature from low 25 to high 40° C)

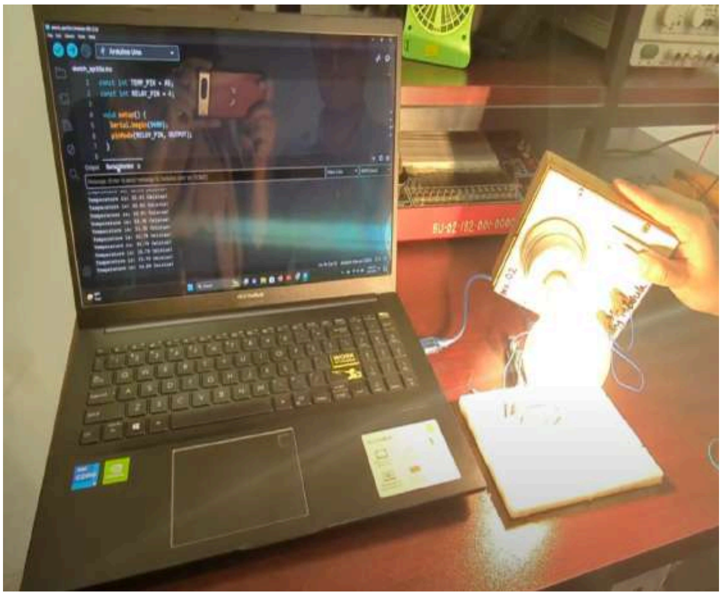


Fig: TMP36 Hardware setup in between 30 to35° C , while rising temperature from low to high.

The same things happen in proteus simulation as well.

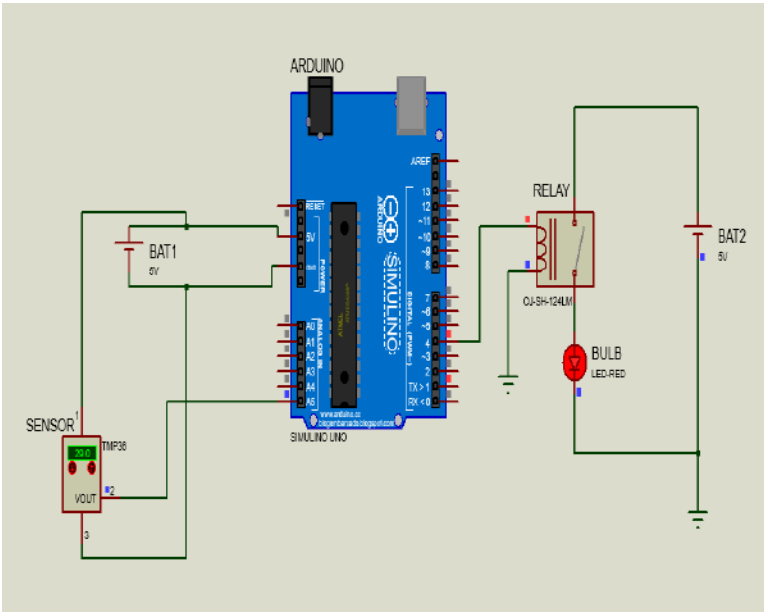


Fig: TMP36 software setup below 30° C

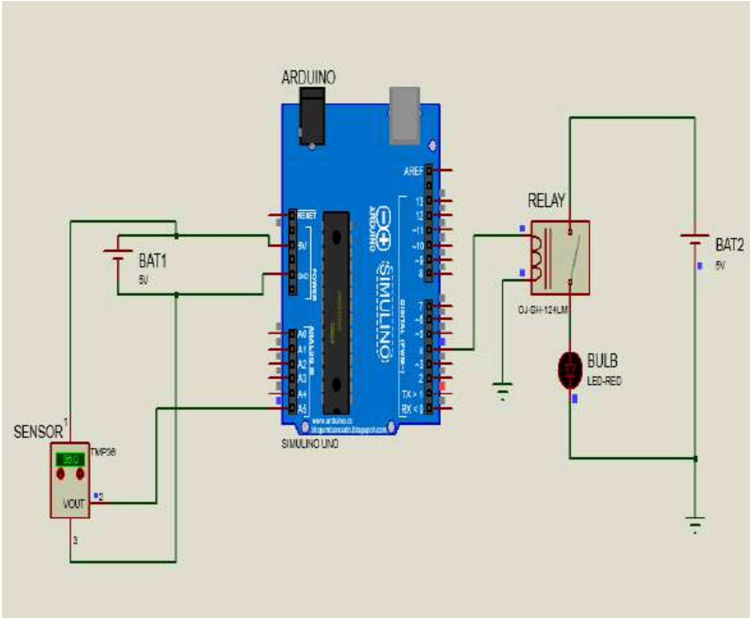


Fig: TMP36 software setup above 35° C

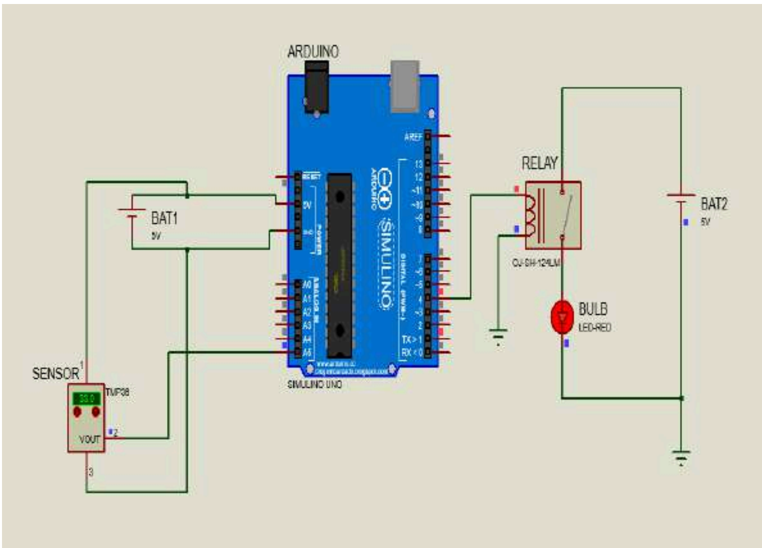


Fig: TMP36 software setup in between 30 to35° C, while rising temperature from low to high.

Dallas DS18B20 Digital:
Similarly, as shown in the figure below, the light will remain on until the temperature reaches 35° Celsius.

Fig: DS18B20 Hardware setup below 30° C

Again, the bulb turns off at 35 degrees Celsius and stays that way until the temperature drops below 30 degrees. This is seen in the following figure:

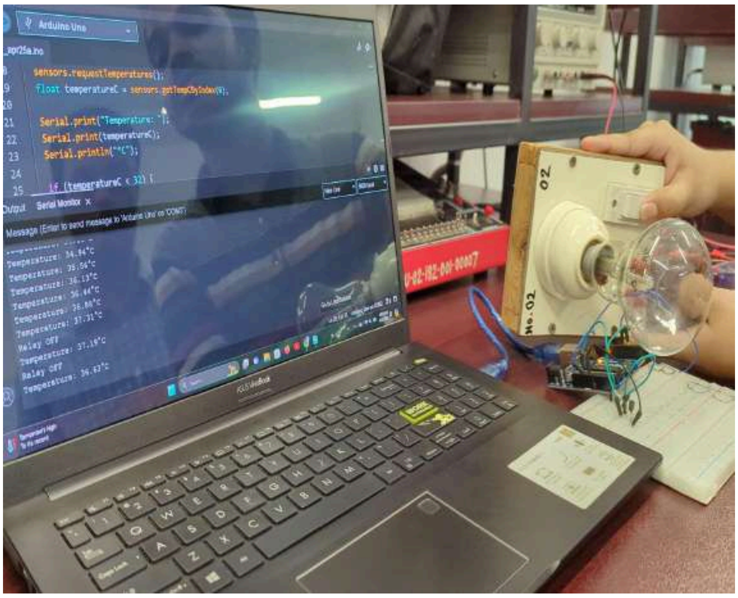


Fig: DS18B20 Hardware setup above 35° C

However, we can observe that the same thing occurs in the Proteus simulation.

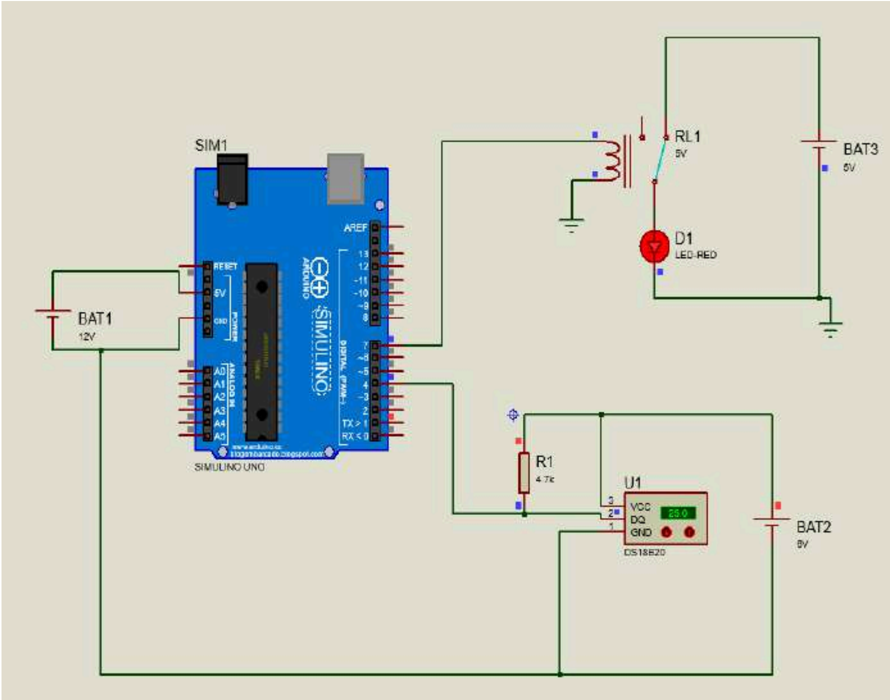


Fig: DS18B20 software setup below 30° C

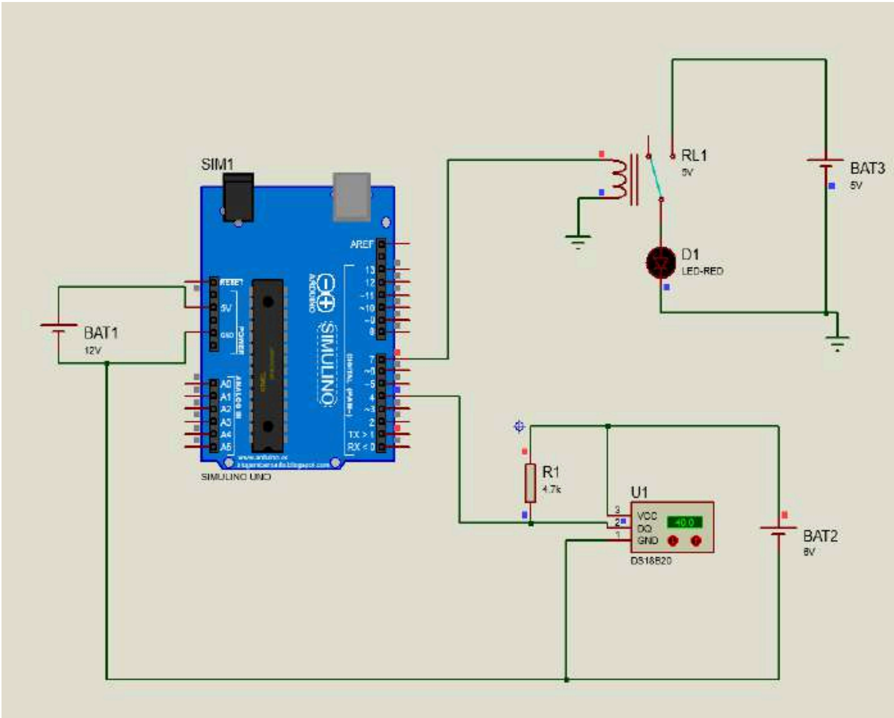


Fig: DS18B20 software setup above 35° C

Tmp36 Analog:

The output of the TMP36 V_S first goes to the scale and shift circuit to be scaled up. From the scale and shift circuit, for 30 ℃ we will get an output of around 3.6V, and for 35 ℃ we will get around 4.5V. Then, this output will go to the comparator circuit. In this part, the comparator circuit will compare the temperature of the sensor (voltage output) of the sensor with the reference value. If the condition is matched, the comparator circuit will give a high output. As there are two comparators to check the sensor output voltage, we will get 4 possible output combinations, and using the SR-latch, and not get we will get the desired feedback from the light bulb. The relay will stay on until the temperature reaches 35° Celsius which can be seen in the figure below.

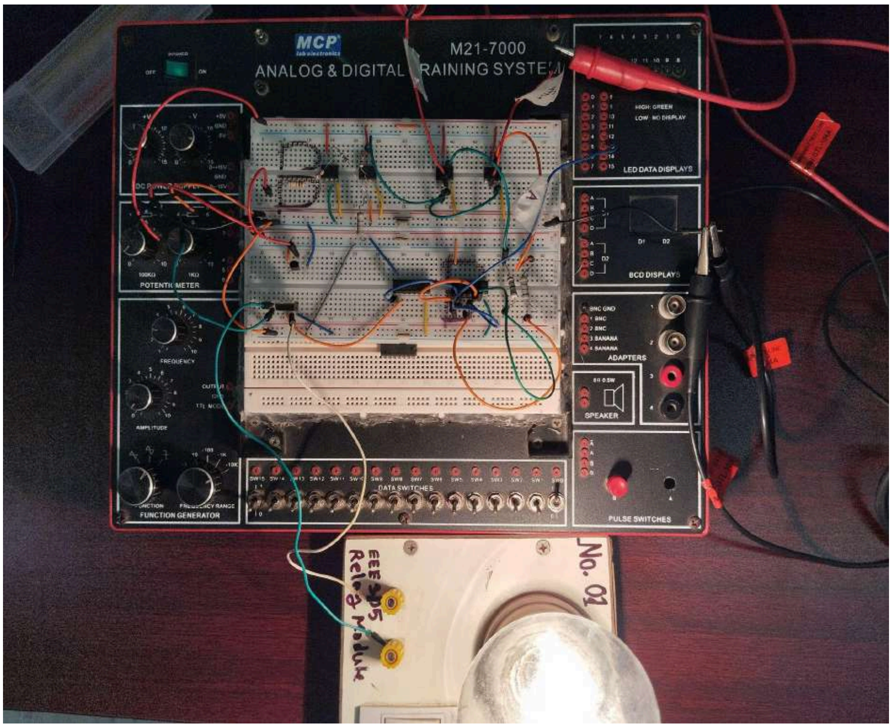


Fig: TMP36 analog Hardware setup below 30° C

Again, the bulb turns off at 35 degrees Celsius and stays that way until the temperature drops below 30 degrees. This is seen in the following figure:

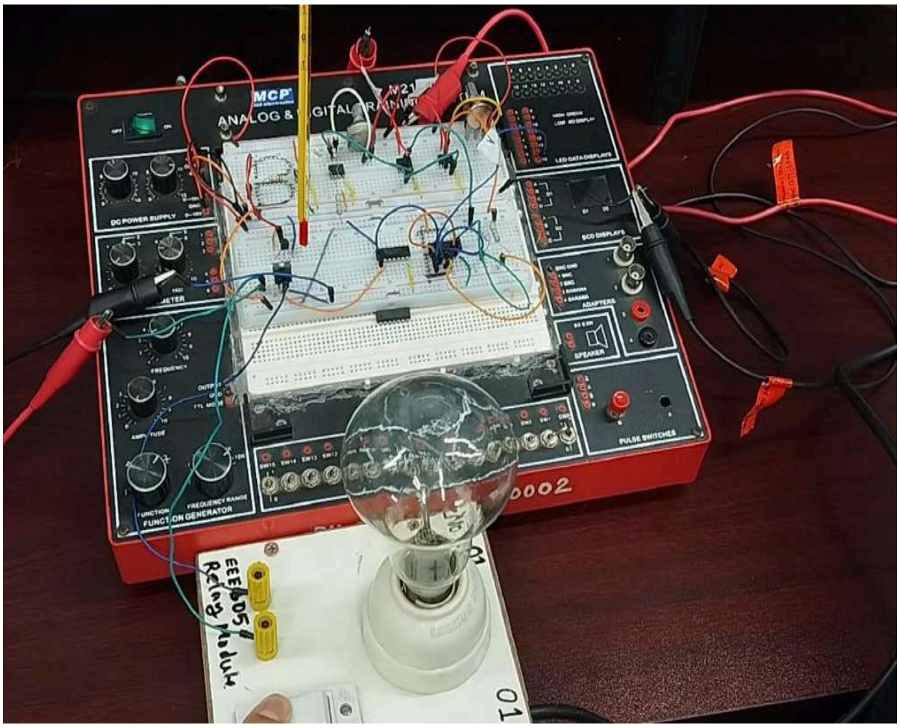


Fig: TMP36 analog Hardware above 35° C

In between 30 to 35° C, the output remains unchanged by the circuit we have made. So, after reaching peak more than 35° C, when it starts to drop .

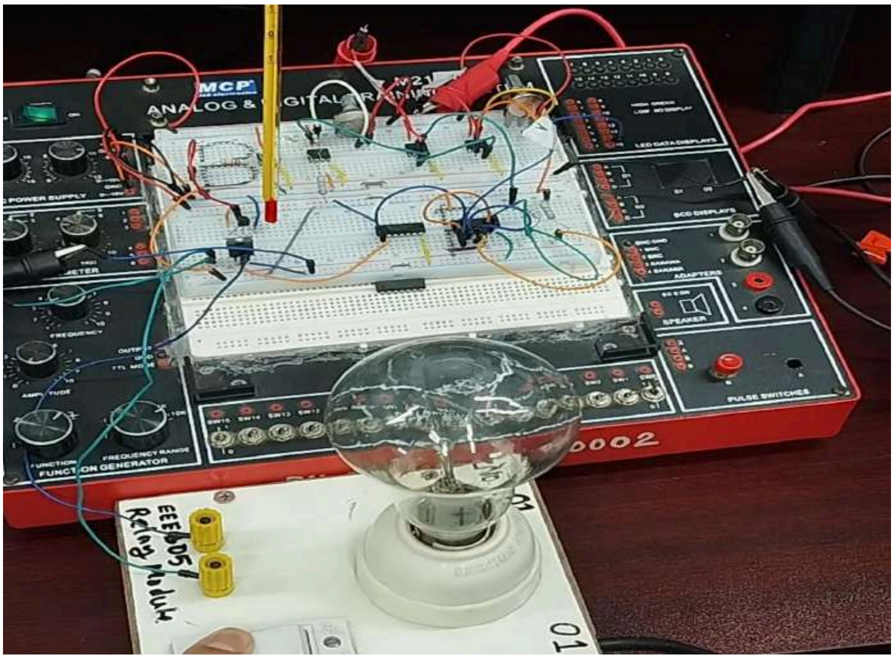


Fig: TMP36 analog Hardware between 30 to 35° C, unchanged condition.

The same thing goes for the proteus simulation:

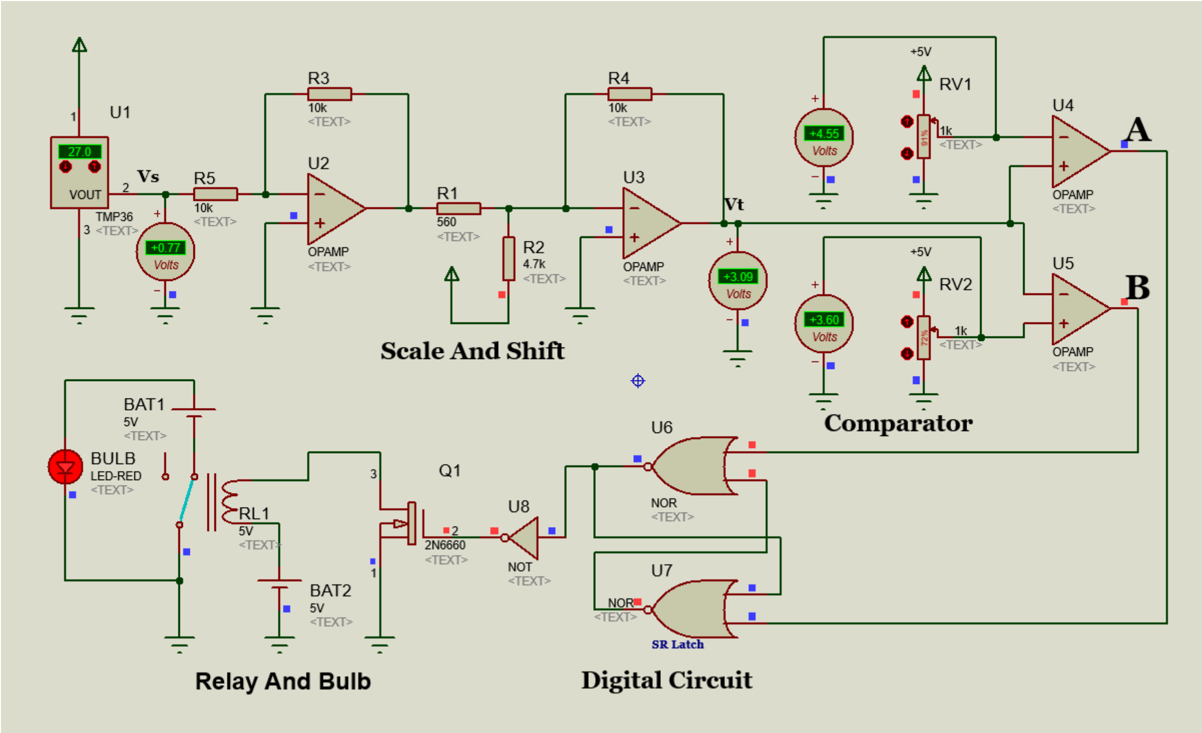


Fig: TMP36 analog Hardware setup below 30° C

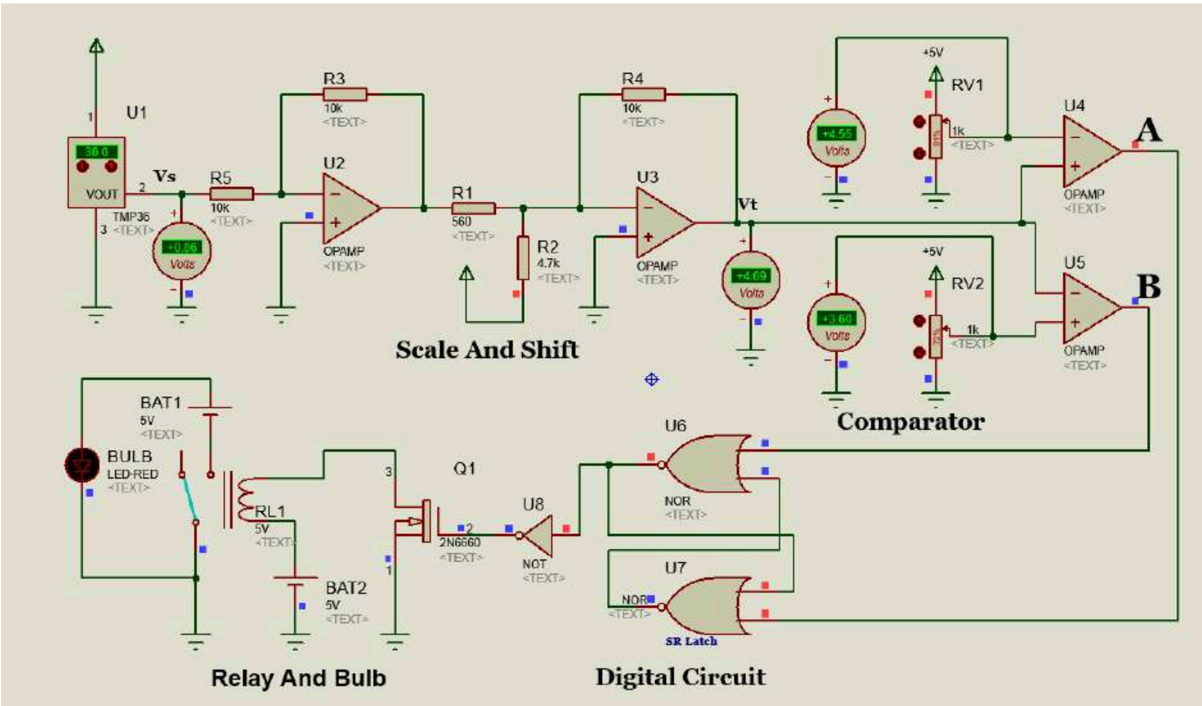


Fig: TMP36 analog Hardware above 35° C

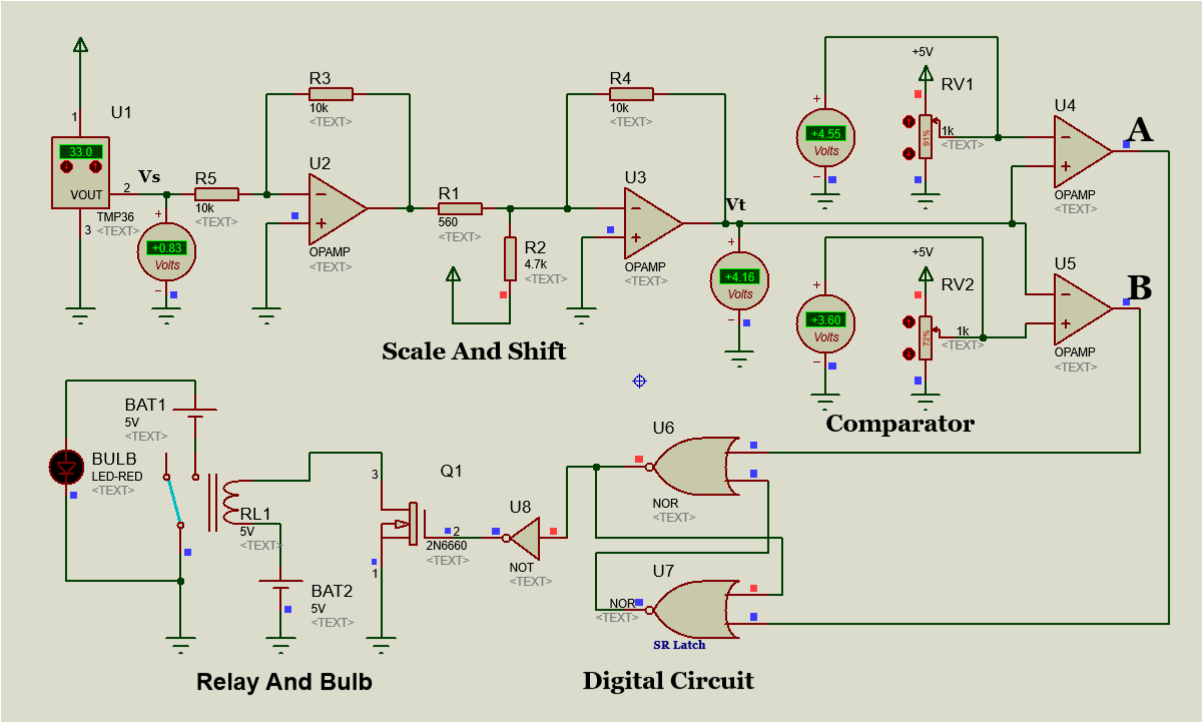
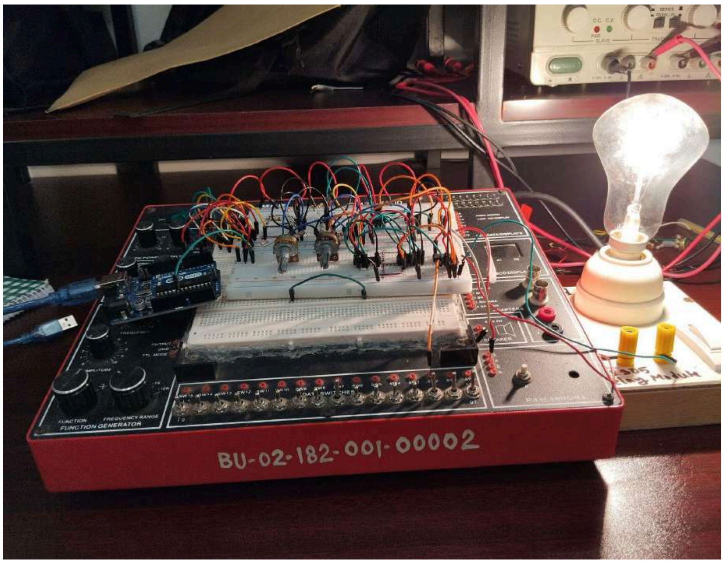


Fig: TMP36 analog Hardware between 30 to 35° C, unchanged condition.

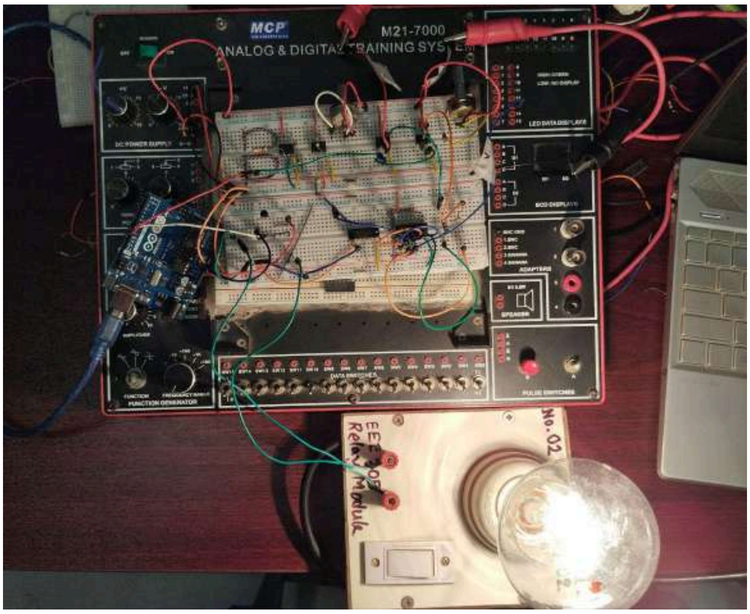
Dallas DS18B20 Analog:

We know that Dallas is a digital sensor, and we have to implement it in the analog circuit. That is why, we have to map the digital signal of the Dallas sensor to an analog output by using an Arduino. We have written the code, and it is working on the software simulation, But when we tried to implement the same thing in the hardware circuit, we started to get -6.2v from the scale and shift circuit. We tried to debug it, but we could not find any solution to it.

Trial 1:



Trial 2:



In the first trial, we tried it using a potentiometer, then we thought that, it might be causing the problem as the comparator circuit was not working, then we tried applying a direct reference voltage to the comparator circuit with a DC source. But, this time we figured out that our scale and shift circuit is not working.

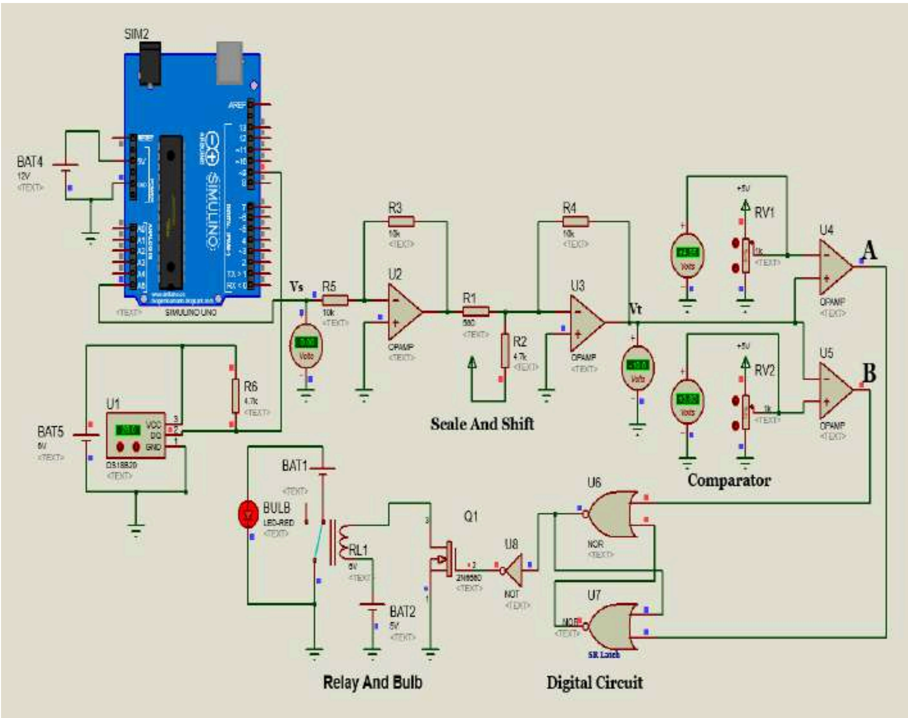


Fig: Dallas analog Hardware setup below 30° C

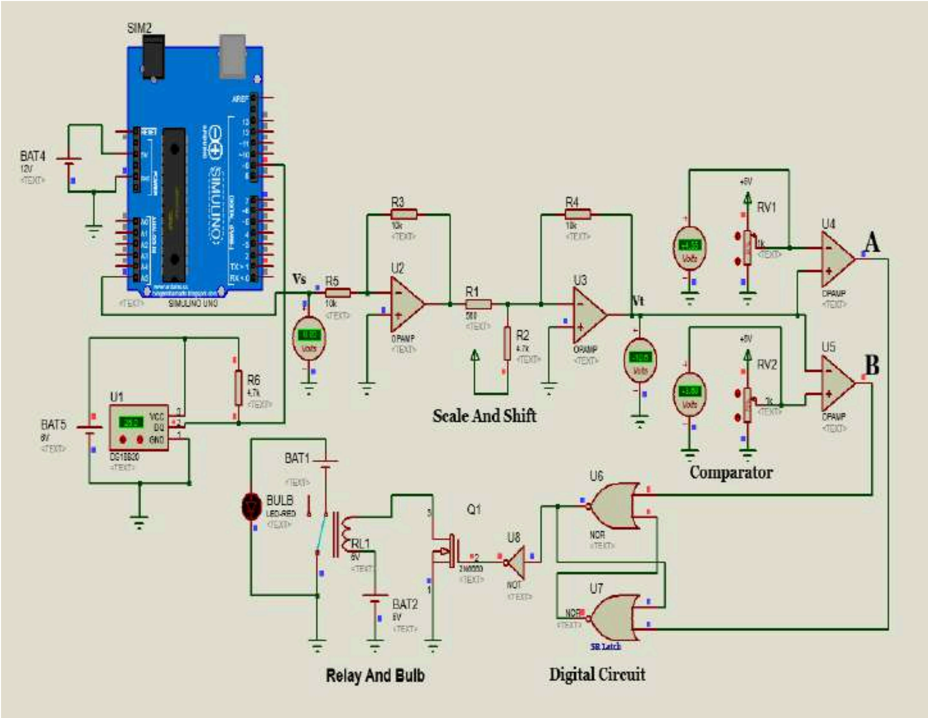


Fig: Dallas analog Hardware setup above 35° C

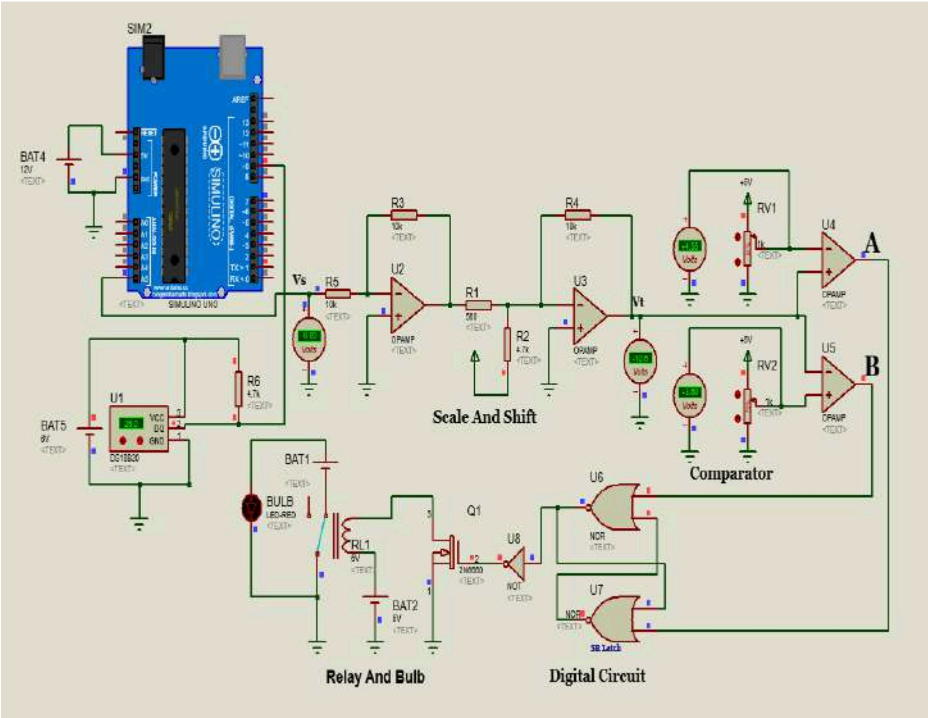


Fig: When dropping the temperature from 35° C, between 30 to 35° C condition.

Cost analysis:

Components	TMP36 Analog	TMP36 Digital	DS18B20 Analog	DS18B20 Digital
TMP36	1*199	1*199	~	~
DS18B20	~	~	1*87	1*87
ARDUINO UNO	~	1*790	1*790	1*790
Potentiometer	2*24	~	2*24	~
Op-amp	4*20	~	4*20	~
Resistor	8*2	~	8*2	1*2
Relay	1*45	1*45	1*45	1*45
Bulb	1*60	1*60	1*60	1*60
NOT Gate	1*28	~	1*28	~
NOR Gate	1*25	~	1*25	~
Mosfet	1*45	~	1*45	~
Jumper Wire	50	50	50	50
total	**596**	**1144**	**1274**	**1034**

In conducting a cost analysis between temperature sensing circuits utilizing the TMP36 and DS18B20 sensors in both analog and digital configurations, several key considerations emerge. The TMP36 Analogue Circuit stands out as a cost-effective solution, providing simplicity in design and lower overall setup expenses. However, its limitations in precision and susceptibility to noise may impact its suitability for applications requiring higher accuracy. On the other hand, the DS18B20 Analogue Circuit is hard to build, difficult to operate and costly.

Transitioning to digital circuits, the TMP36 Digital Circuit introduces advantages in terms of accuracy and adaptability. Its cost implications are influenced by the additional components required for digital processing. Although more expensive than its analogue counterpart, the TMP36 Digital Circuit strikes a balance between performance and cost. The DS18B20 Digital Circuit emerges as the optimal choice, combining the precision of a digital sensor with the advantages of straightforward implementation. While marginally pricise than the TMP36 Analogue Circuit, the DS18B20 Digital Circuit justifies its cost through superior accuracy, reliability, and robustness. In scenarios where precision and performance are paramount, the DS18B20 Digital Circuit proves to be a cost-effective investment, outperforming both analog and digital alternatives.

Performance Analysis:

Analog Circuit with TMP36:

Positive sides:

- Cost effective: Analog circuits, like the one with the TMP36, are very cost effective.
- Real-time Output: The TMP36 provides continuous analog voltage output proportional to temperature, enabling real-time monitoring.

Negative sides:

- Limited Precision: Analog circuits may offer limited precision compared to digital alternatives, potentially leading to less accurate temperature measurements.
- Susceptibility to Noise: Analog signals, including those from the TMP36, are vulnerable to noise interference, which can affect temperature accuracy.

Analog Circuit with DS18B20:

Positive sides:

- Higher Precision: The DS18B20 typically provides higher precision compared to the TMP36 sensor.

Negative sides:

- Complexity: Implementing a digital circuit, such as the one with DS18B20, can be more complex than TMP36 analog setup. The DS18B20 sensor provides temperature readings in a digital, so an analog to digital converter is required here. Which makes it more complex, and hard to debug.
- Cost: The DS18B20 sensor may incur higher costs compared to the TMP36. And additional requirement of an analog to digital converter makes it more expensive.

Digital Circuit with TMP36:

Positive sides:

- Digital Processing: Integrating the TMP36 with digital processing enables improved noise filtering and potentially increased temperature accuracy.

Negative sides:

- **Inaccurate:** TMP36 sensor is very noisy, hard to get precise output like DS18B20.
- **Cost:** The additional components and processing capabilities required for digital processing may increase the overall cost.

Digital Circuit with DS18B20:

Positive sides:

- Digital Precision: The DS18B20 provides digital temperature readings with high precision, ensuring accurate temperature monitoring.
- Noise Immunity: Digital signals are generally more immune to noise interference, resulting in more reliable temperature measurements.

Negative sides:

- Complexity and Cost: As with other digital circuits, implementing a digital solution with the DS18B20 may be more complex and costlier than analog alternatives.

Configuration	Performance	Cost	Complexity
TMP36 Digital	Low	High	Low
TMP36 Analog	Low	Low	High
DS18B20 Digital	High	High	Low
DS18B20 Analog	Low	Low	High

Summary:
In summary, after assessing all configurations, it's evident that the DS18B20 digital configuration emerges as the optimal choice for constructing this type of temperature control system. The TMP36 sensor, while simple to implement, proves to be noisy and less accurate, compromising its suitability for precise temperature monitoring. Even its digital configuration is hindered by the inherent limitations of the sensor. Conversely, while the DS18B20 sensor offers higher precision, its analog configuration presents challenges in construction complexity. Thus, the DS18B20 digital configuration stands out as the preferred solution, offering superior accuracy, reliability, and ease of implementation for temperature control applications.

Conclusion:
In conclusion, our discussion has covered various aspects of temperature sensing circuits, focusing on the utilization of the TMP36 and DS18B20 sensors in both analog and digital configurations. We explored the advantages and disadvantages of each circuit type, considering factors such as simplicity, precision, susceptibility to noise, complexity, and cost. Analog circuits with the TMP36 offer simplicity and real-time output but may lack precision and be susceptible to noise. On the other hand, analog circuits with the DS18B20 provide digital output and higher precision but are more complex and costly. Transitioning to digital circuits introduces benefits such as improved accuracy and noise filtering, with the DS18B20 digital circuit emerging as the optimal choice for precision-critical applications despite its increased complexity and cost. Ultimately, the selection of a temperature sensing circuit depends on specific project requirements, balancing factors such as accuracy, cost, and ease of implementation.

References:

- https://wiki.mchobby.be/index.php?title=ENG-CANSAT-TMP36&mobileaction=toggle_view_desktop#:~:text=25%C2%B0%20C%20%2D%2D%3E%20output,%C2%B0%20C%20%2D%2D%3E%20output%20%3D%20500mV

- https://www.elprocus.com/ds18b20-temperature-sensor/

- https://ieeexplore.ieee.org/document/832887

- https://docs.arduino.cc/hardware/uno-rev3