**PHASE III PROJECT**

**(PRELIMINARY DESIGN)**

***On***

**Automatic Room Temperature Controller**

**Submitted for the requirement of**

**Project course**

BACHELOR OF ENGINEERING

**COMPUTER SCIENCE & ENGINEERING**

****

**Submitted to: Submitted By:**

**Project Teacher (Supervisor)**

Shuvendu Das (12496) **NAME UID**

Sahil Kaundal 21BCS8197

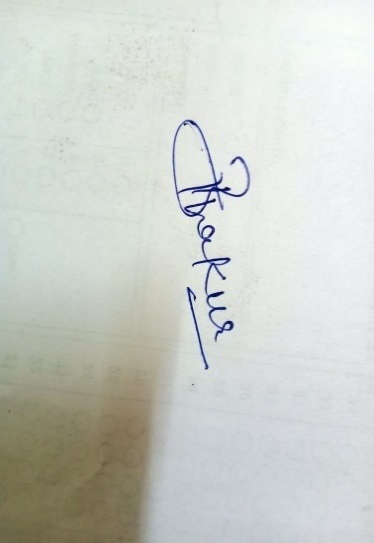
Rohit Kumar Goswami21BCS10083

Pratik Kumar Dey21BCS8183

Feroj Khan 21BCS8168

**Co Supervisor**

Bhanu Thakur (12251)

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CHANDIGARH UNIVERSITY, GHARUAN**

**June 2022**

**ABSTRACT**

Temperature control refers to the processes that are aimed at maintaining the

temperature in a given area at certain maximum/minimum level or within a certain range.

This process is commonly used in most areas of the world. Recently, globalization and

industrialization has further necessitated the need for Temperature Control applications in

various daily activities, especially with the advent of the greenhouse effect.

Many Homes and Industries among other areas maintain certain sections of operation

that must be maintained within a certain temperature for process to work successfully. In

research laboratories, the lack of use of Temperature Control Systems has led to the

purchase of chambers of various sizes where temperature specific research work would be

kept. This has also led to an increase in overhead cost. In areas that have electronic activities

or machinery functioning constantly, such as in server rooms and production plants. These

are places where heavy machinery and computers work continuously 24 hours every day.

During these processes, the temperature needs to be monitored frequently in order to ensure

that it doesn’t rise or fall below a value that would accelerate wearing out of the systems.

It is important also to monitor the level of temperature various other places such as

morgues, hospitals, aircrafts, living rooms, etc, to ensure that thermal comfort is maintained.

Thermal comfort is generally defined as that condition of mind or functionality which

expresses satisfaction with the thermal environment (e.g. in ISO 1984). Dissatisfaction may

be caused by the body / equipment being too warm or cold as a whole, or by unwanted

heating or cooling of a particular part of the body (local (functional) discomfort).

Automatic room temperature controller is certified as the best method in any application

because the temperature is usually controlled automatically throughout the process. The results obtained from various applications of the process across different regions and timelines shows the temperature is controlled effectively and more accurately. In addition, this finding also makes human work easier as an automatically controlled system worries about other contingent weather issues for you.

The major objective of this project would be to create a Automatic Room Temperature Controller System that would be able to automatically control the temperature of the room it is placed in

by the timely activation of the effector devices to influence the temperature in relation to the

set-point.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  | **TABLE OF CONTENT** | |  | |  |  |  |  |  | | Sr | Topic | | | Page No. | | no. |  |  |  |  | | 1 | Feature/characteristics identification | | | 5 - 18 | | 2 | Constraints Identification | | | 19 – 21 | | 3 | Analysis of features and finalization subject to constraints | | | 22 – 25 | | 4 | Design selection | | | 26 - 33 |   References | | |  |
|  | |  |  |
|  | |  |  |
|  |  | |  |

**Feature/Characteristics Identification**

The concept of this report is to create an automatic room temperature control system to control the temperature of a room. This circuit maintains the temperature of the room in a particular range. Automatic control plays an ever-increasing role in human way of life. During winter and summer season, room temperature must be maintained in order to ensure the human convenience. This project mainly includes the temperature control of surrounding in winter and summer. It consists of Temperature sensing unit, LCD module, a fan and heater. It will operate based on the value or range of Temperature in the system which is detected by Temperature of the sensor. The Temperature sensor detect the temperature of the room. The Temperature sensor is connected to the Arduino. It converts the analog input to digital value. It is use to switch on the heater and fan. The LCD module is also connected to the Arduino. The module displays the current temperature. The LCD display used is a 16x12 Alphanumeric display.

**1.1. Objectives**

The objectives of this project are:

1. To test its applicability in different temperature ranges as measured by the TMP36 temperature sensor.

2. To analyses the performance of the controller.

3. To develop a smart automatic room temperature controller system.

4. To design and construct an automatic temperature controller to regulate the temperature of a room.

5. To build an automatic room temperature controller making use of a temperature sensor and a microcontroller.

6. To achieve the design of an alert system for the temperature controller.

**1.2. Single entity**

In this project i.e., Automatic Room Temperature Controller there are two parts that are:

* Coding Part
* Hardware part

First of all, we write the code in small modules using a software Tinkercad. After writing the code we can make a circuit diagram for our project and implement them in the software Tinkercad. After this we can combine the coding of the project with the circuit and test the program.

**In coding part,** we have prior make modules of the code (in which different Arduino pin setup coding are present i.e., deciding that which pin of Arduino Uno perform what task. Then we define different functions that can reduce the code efforts and size of the code. After that we can combine these small modules and create a final code in C language.)

**In hardware part,** firstly we roughly make a circuit diagram of the project and then we test it on a software name Tinkercad. After final execution we will working on the hardware part i.e., we take an Arduino Uno, Temperature Sensor, Led, DC Motor, LCD Display, Potentiometer, Diode, Resistance, Jumper Wires, Transistors etc, and connected them according to the circuit diagram.

At last stage, we can combine both hardware and software part. We can upload the code in Arduino Uno with the help of Arduino IDE (Arduino Integrated Development Environment).

**1.3. Life Span**

In this project i.e., Automatic Room Temperature Controller there are five modules that are:

* Breadboard Supply Module
* LCD Display Module
* Temperature Sensor Module
* Heating Module
* Cooling Module

The total time span of this project is 2 and half months.

**Module 1: Breadboard Supply Module**

If we want to easily power the components on your breadboard, then a Breadboard Power Supply Module is the right choice. It provides dual output voltages: 3.3V and 5V on two different power rails.

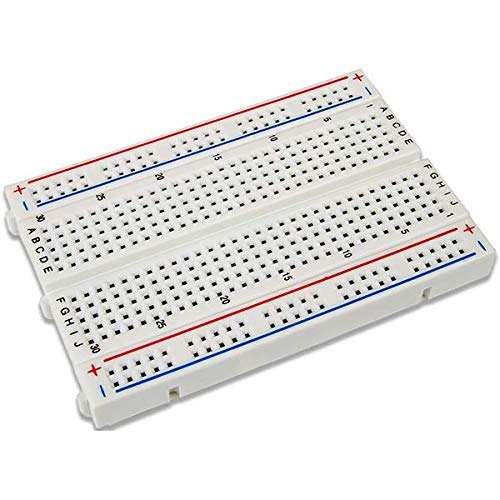


Fig1.1: Breadboard Power Supply Module

This module will perfectly fit on to 400-point and 800-point breadboards. The input range is 6.5V to 12V. The time span for this module can be completed in half month.

**Module 2: LCD Display Module**

The [Liquid Crystal library](https://www.arduino.cc/reference/en/libraries/liquidcrystal/) allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

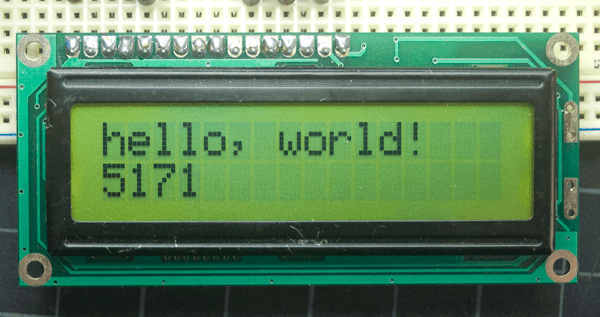
[](https://docs.arduino.cc/static/7a7f1f877f04d48236ab166814aab58f/0a47e/lcd_photo.png)

Fig 1.2: Output of the sketch on a 16x2 LCD

The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The time span for this module can be completed in half month.

**Module 3: Temperature Sensor Module**

The TMP36 temperature sensor is an easy way to measure temperature using an Arduino! The sensor can measure a fairly wide range of temperature (-50°C to 125°C), is fairly precise (0.1°C resolution), and is very low cost, making it a popular choice.

Unlike a thermistor, the TMP36 does not have a temperature sensitive resistor. Instead, this sensor uses the property of diodes; as a diode changes temperature the voltage changes with it at a known rate. The sensor measures the small change and outputs an analog voltage between 0 and 1.75VDC based on it. To get the temperature we just need to measure the output voltage and a little bit of math. The time span for this module can be completed in half month.

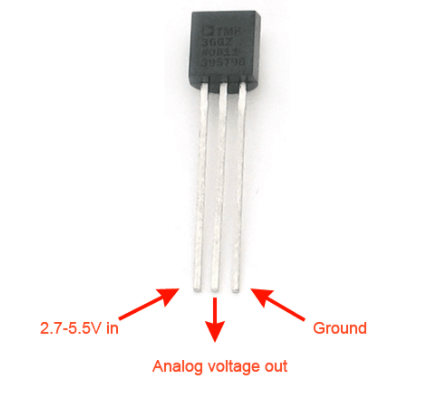


Fig 1.3: TMP36 Temperature sensor

**Module 4: Heating Module**

In this module we can use LED instead of heater for only temporary purpose. This module can work only when the temperature of the room decreases. It will start glowing the LED instead of heater that means our heater is started. This module can start when the temperature sensor gives the reading less than 20 degrees Celsius. The time span for this module can be completed in half month.

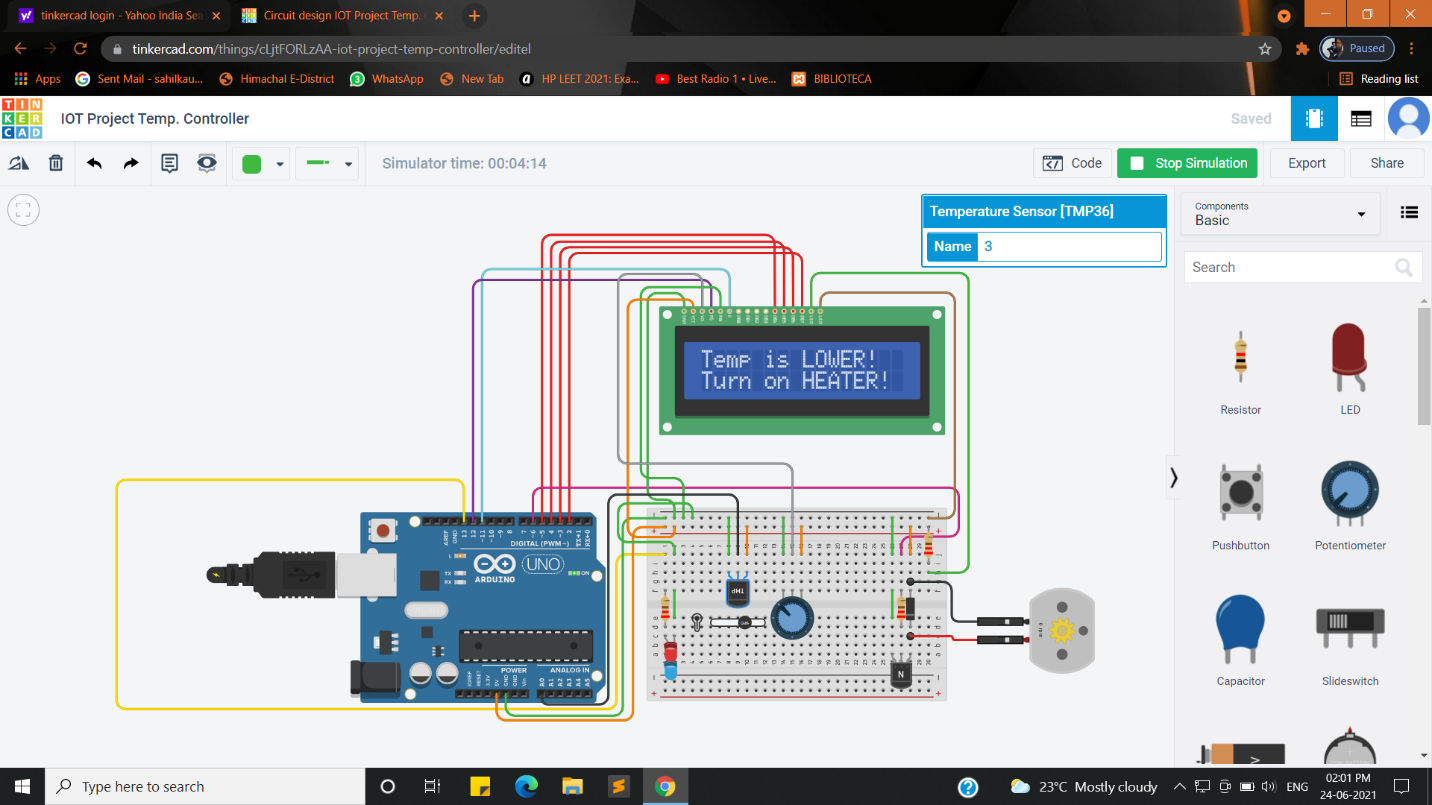


Fig 1.4: Heating Module

**Module 5: Cooling Module**

In this module we can use DC Motor instead of cooling for only temporary purpose. This module can work only when the temperature of the room increases. It will start rotating the DC Motor instead of Fan that means our cooling is started. This module can start when the temperature sensor gives the reading more than 25 degrees Celsius. The time span for this module can be completed in half month.

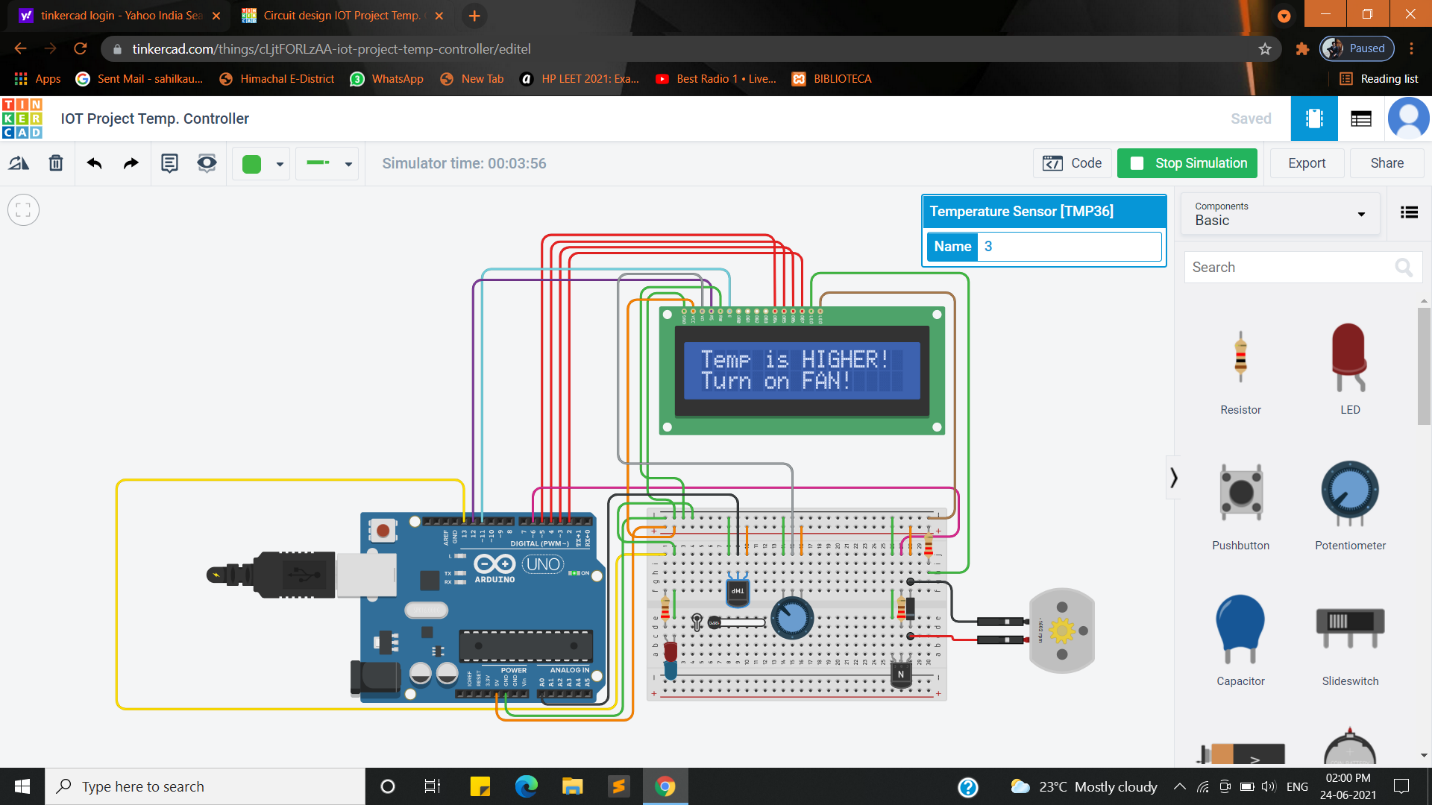


Fig 1.5: Cooling Module

**1.4. Require funds**

This project needs funds to reach the endpoint. Without adequate funds, this project cannot be successfully implemented. Cost estimation is one of the essential factors for any organization. So, calculating in advance the required funds for this project will be very impactful.

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Items** | **Estimated Cost (Rs.)** |
| 1. | Arduino Uno | 800 |
| 2. | Breadboard Small | 100 |
| 3. | DC Motor | 50 |
| 4. | NPN Transistor | 15 |
| 5. | Temperature Sensor [TMP36] | 30 |
| 6. | LCD 16\*2 | 200 |
| 7. | 250 Kilo Ohm Potentiometer | 30 |
| 8. | 221 Ohm Resistor | 10 |
| 9. | LED | 10 |
| 10. | Diode | 20 |
| 11. | Jumper Wires | 50 |

Table 1.1: Cost Estimation

This estimated cost for Automatic Room Temperature Controller.

**1.5. Life Cycle**

**Project Planning Phase:**

This project work presents the design of an automatic room temperature controller system. This system allows the user to set a desired temperature between (20’C to 25’C the normal temperature of room) which is then compared to the room temperature measured by TMP36 temperature sensor. With the help of Atmega 328P based on Arduino microcontroller, the system responds by turning ON and OFF the two (2) loads (fan or a heater) automatically depending on the temperature difference in the room. The fan is triggered ON when the room temperature is higher than the set temperature and the heater is triggered ON when the room temperature is lower than the set temperature. The system was designed and simulated using Tinkercad, circuit building software used for building electronics system. Tinkercad software was used to design and simulate the main circuit; Arduino IDE was used to write a program on how the system will operate which was then loaded to the microprocessor. An LCD was used to display temperature readings obtained by the TMP36 temperature sensor and the condition of the heater and the fan whether ON or OFF. When the system was simulated the heater and the fan were turned on automatically depending on the room temperature. The heater turned on when the temperature was below 20 degrees Celsius while the fan turned on when temperature fell below the threshold. In both cases, either of them was on at a specified time but not both. When temperature is between 20 to 25 degrees Celsius, the both heater and fan will OFF.

**Defining Phase:**

The problem happens when the air conditioning is still functioning although in the event of cold weather. The function is uncontrolled and must be manually turned on and off. Sometimes it can lead to high usage of electricity which in turn raises the electricity bill when the user forgot to switch it off. The system also does not have the capacity to adjust the room temperature regardless of the ambient temperature. To address the problem, the automatic room temperature controller that can control the temperature automatically is proposed. The advantages of such a system are less energy usage, and provides more convenient to the consumers.

All living organisms want comfort and do away with all aspects that try to bring discomfort to them. One of the common causes of discomfort is temperature fluctuations. Human beings are the most peculiar species known for its problem-solving skills and they have not been left behind in trying to solve these temperature fluctuations. Most of the systems present however are manual hence tedious to turn on and off. These expensive systems are also disadvantageous to the physically challenged people who find it difficult to access switching mechanisms. This makes it necessary to have an automatic room temperature controller system within the room that can be operated by both able-bodied and physically challenged. The air conditioner minimizes discomforts due to varying weather conditions. This system is efficient, cheaper, easier to use and cost effective.

Room temperature always changes and sometimes it occurs drastically (for example: rainy day at the night). Thus, uncomfortable environment will be happened to the human life. Secondly, human aided control is commonly use but in certain time there are some unexpected matters occur. For this reason, automatic control system is needed.

**Designing Phase:**

The objectives of the SDLC Design Phase are as follows:

Objectives:

1. To test its applicability in different temperature ranges as measured by the TMP36 temperature sensor.

2. To analyses the performance of the controller.

3. To develop a smart automatic room temperature controller system.

4. To design and construct an automatic temperature controller to regulate the temperature of a room.

5. To build an automatic room temperature controller making use of a temperature sensor and a microcontroller.

6. To achieve the design of an alert system for the temperature controller.

Here, we can make the architecture of the project and make the circuit diagram.

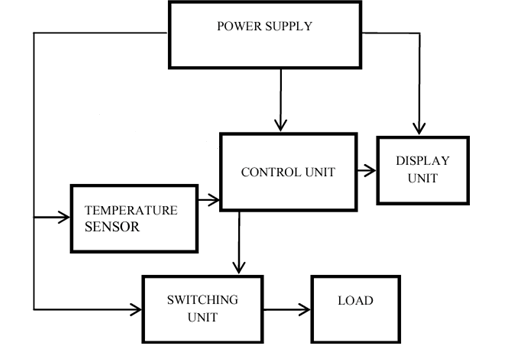


Fig 1.6: Architecture

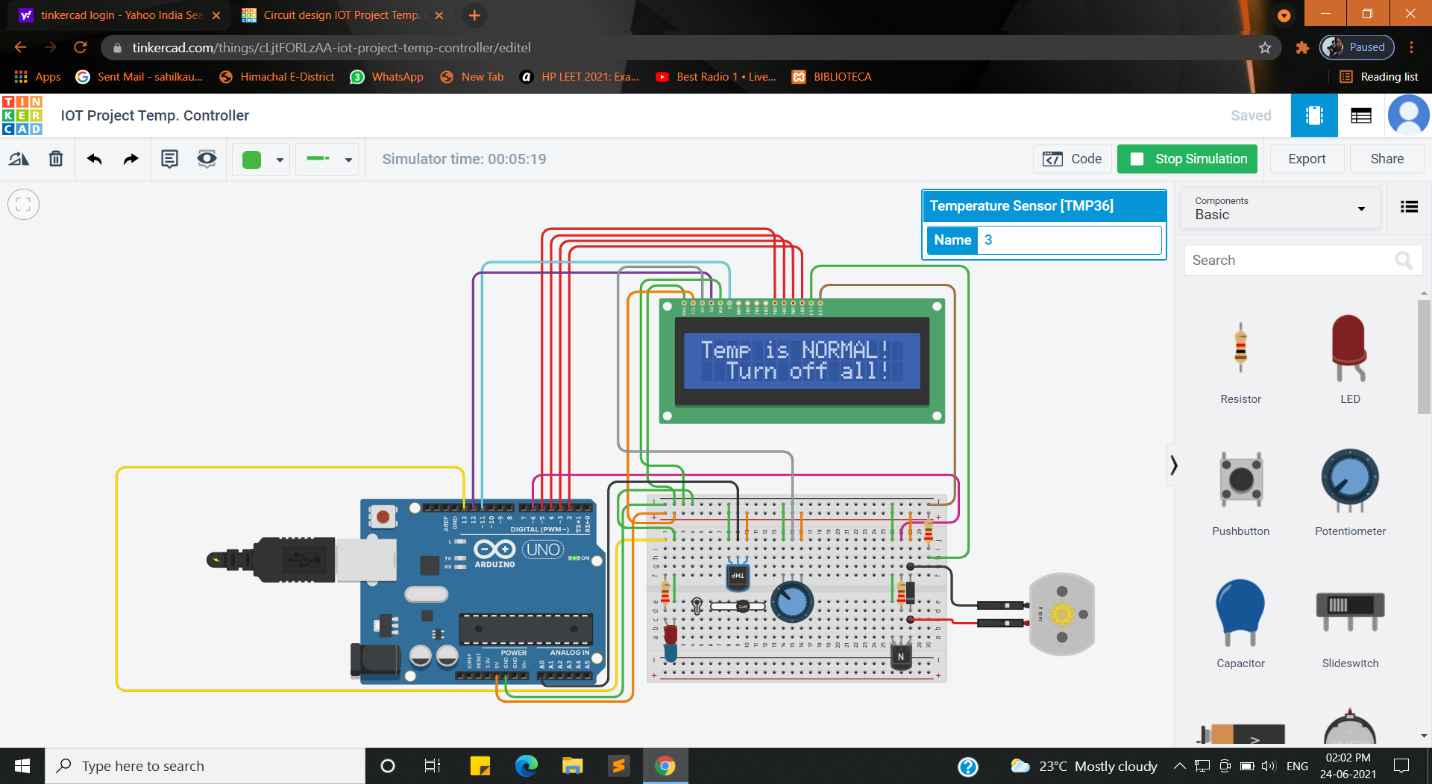


Fig 1.7: Architecture

**Building Phase:**

In this phase we start developing the code in small modules in C Language.

Below the sample code is mentioned which we can used in our project.

// Declaring Arduino IO-pins

const int temp\_trans\_pin = A0, Heater\_pin = 13, FAN\_pin = 6;

/\*For Fan:- DC Motor

For Heater:- LED\*/

// Setting the range of suitable temperature

float MinTemp = 20, MaxTemp = 25;/\*Room temperature is [20,25] degree C \*/

// Include the LCD library code

#include <LiquidCrystal.h>

// Initialize the library with the numbers of the interface pins

LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

void setup() {

// System initialization

LCD.begin(16, 2);

pinMode(Heater\_pin, OUTPUT);//LED in our case

pinMode(FAN\_pin, OUTPUT);// DC Motor in our case

// Displaying the suitable range of temperature

LCD.print("Room temp(C):");

LCD.setCursor(2,1);

LCD.print(MinTemp);

LCD.print("-");

LCD.print(MaxTemp);

delay(2000);

}

void loop() {

float Eqv\_volt, SensorTemp;

// Read voltage and convert to temperature (Celsius)

Eqv\_volt = analogRead(temp\_trans\_pin) \* 5.0 / 1023;

SensorTemp = 100.0 \* Eqv\_volt-50.0;

// Displaying the sensor reading

LCD.clear();

LCD.print("Sensor reading:");

LCD.setCursor(2,1);

LCD.print(SensorTemp);

LCD.print(" C");

delay(2000);

**Testing Phase:**

Testing is the last phase of the software development life cycle before the software is delivered to customers. During testing, experienced testers start to test the system against the requirements.

At this stage we can combine the coding part and hardware part and test the project against the objectives.

Below the results of testing are mentioned:

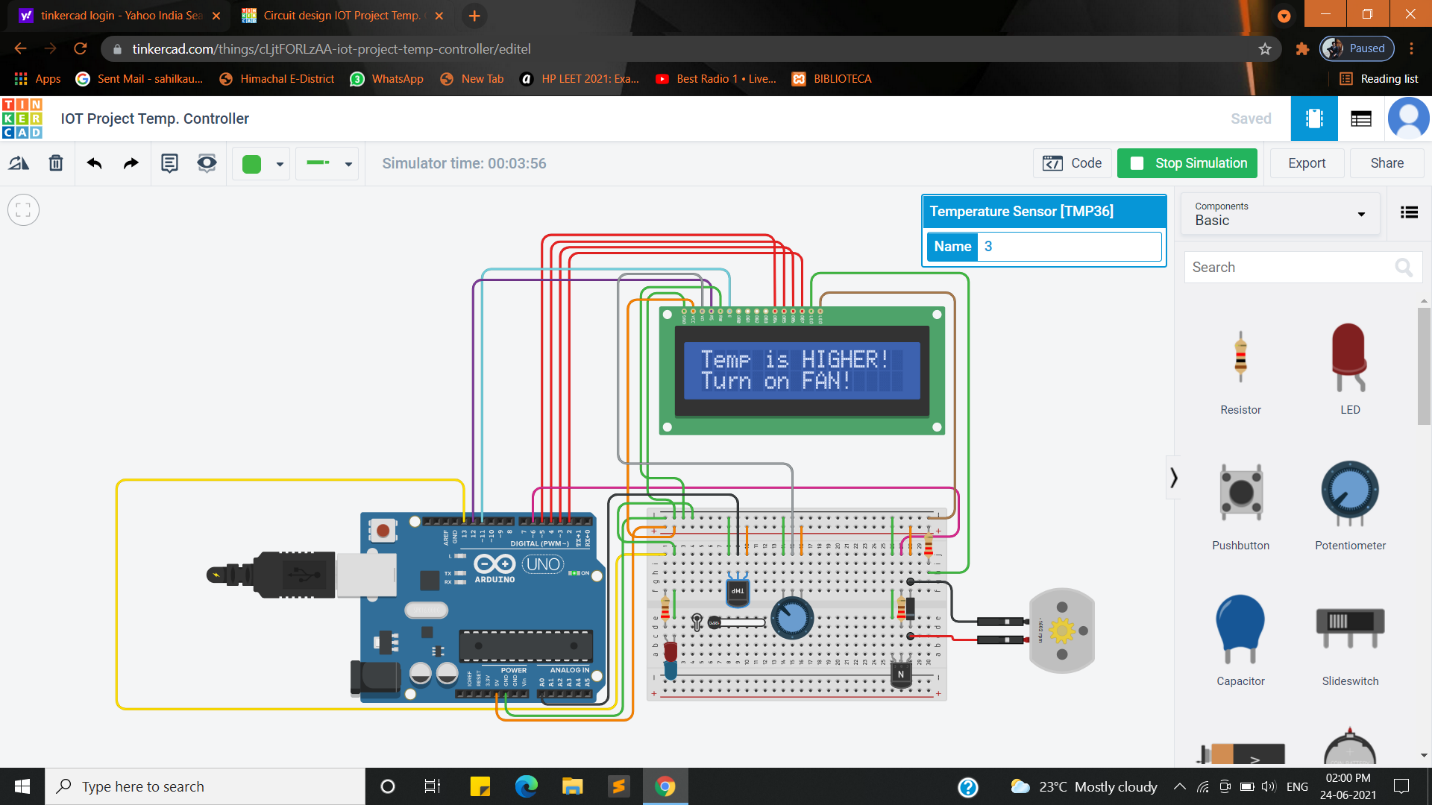
****

Fig 1.8: Testing

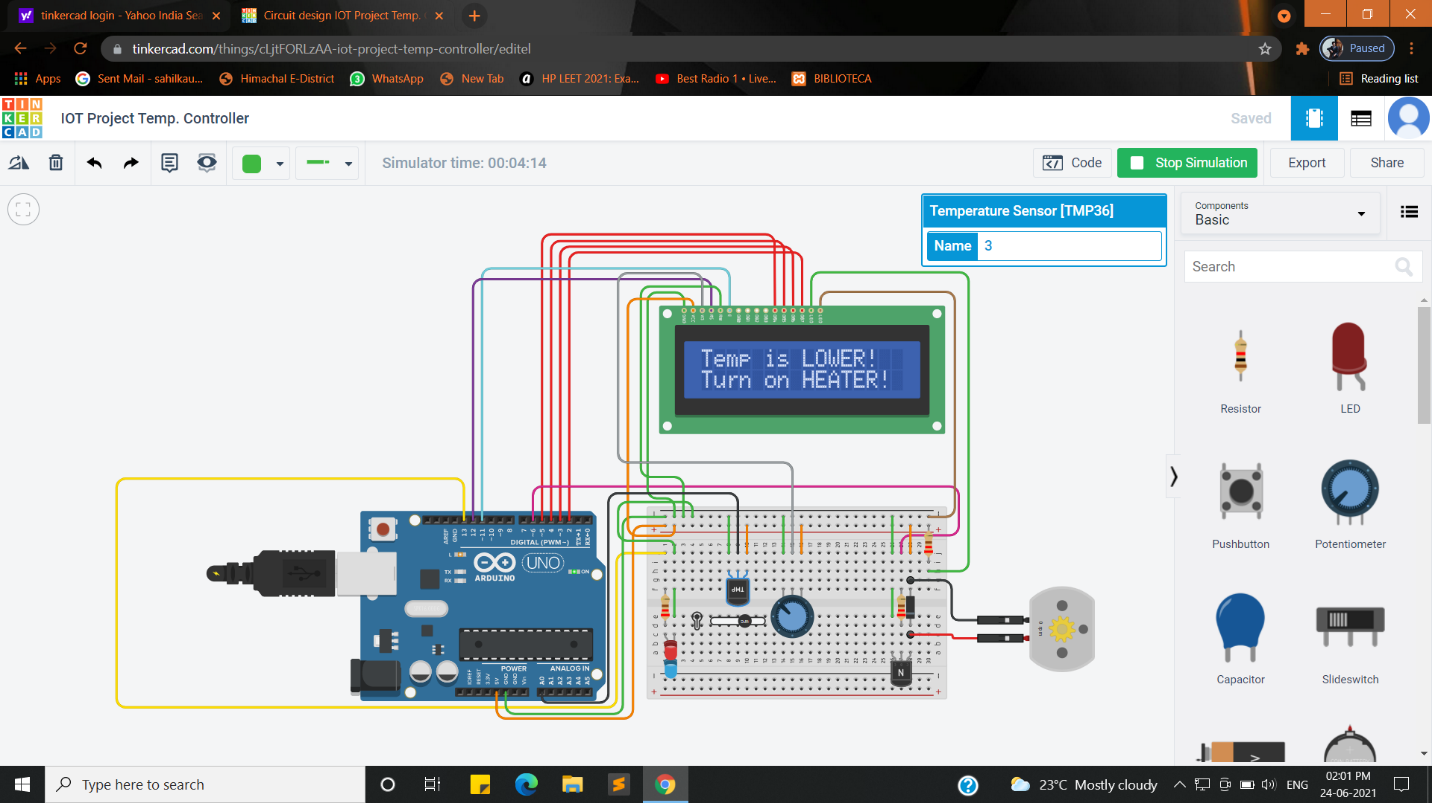
****

Fig 1.9: Testing

**Deployment Phase:**

Once the project has been fully tested and no high priority issues remain in the project, it is time to deploy to production where customers can use the system.

This project is ready for deploy in market. It will be tested successfully.

**1.6. Team Spirit**

Team spirit is required to get the project completed because the project constitutes different members having different characteristics and from various disciplines. But to achieve common goal harmony, missionary zeal, team spirit

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Member** | **UID** | **Roles** |
| 1. | Sahil Kaundal | 21BCS8197 | Coding Work |
| 2. | Rohit Kumar Goswami | 21BCS10083 | Hardware Work |
| 3. | Pratik Kumar Dey | 21BCS8183 | Hardware Work |
| 4. | Feroj Khan | 21BCS8168 | Coding Work |

Table 1.2: Team Members

Different members of the team perform different tasks and at the later stage we combine all their efforts and achieve our goal. All team members give their best efforts to this project. Due to their best effort, we can complete our objectives.

Due to team spirit this project is successfully completed.

**1.7. Risk and Uncertainty**

The project is generally based on forecasting. So, risk and uncertainty are always associated with projects. There will be a high degree of risk in the project which are not properly defined. Only the degree of control over risk and uncertainty varies with the project being conceived based on information available.

**1.8. Directions:**

Project is always performed according to the directions given by the customers with regard to time, quality and quantity, etc. The convenience of the supply sides of economics such as labor availability ore resources and managerial talent etc. are all secondary concerns, primary being the customer requirement.

**1.9. Uniqueness**

Each project is unique in itself, and it’s having own features. No two projects are similar even if the type of organization is the same. The uniqueness of the project can measure by considering the many factors like objectives, features of the project, application of the project, etc.

Here are some of the features of this project that makes it unique:

• The features of the system are less energy usage and it provides more convenience to the user

• It automatically manages temperature of the room.

• Less expensive then air-conditioned.

• It reduces electric power consumption.

• Easily install at any place.

**1.10. Flexibility**

Change and project are synonymous. This project sees many changes throughout its life span. These changes can make projects more dynamic and flexible.

**1.11. Sub-Contracting**

Sub-contracting is a subset of this project and without which this project can be completed unless it is a proprietary firm or tiny in nature. The more complexity of project the more will be the extent of contracting. This project needs the help of an outsider consultant, engineer, or expert in IOT field.

**1.12. Cost**

If the quality of the project is to be changed there could be an impact on the cost of the project. The cost could increase if more resources are required to complete the project quicker.

**Constraints Identification**

There are six major constraints in project management to consider.

* 1. **Time:** In this project i.e., Automatic Room Temperature Controller there are five modules that are:
* Breadboard Supply Module
* LCD Display Module
* Temperature Sensor Module
* Heating Module
* Cooling Module

The total time is taken by this project is 2 and half months.

**Breadboard Supply Module**

The time span for this module can be completed in half month.

**LCD Display Module**

The time span for this module can be completed in half month.

**Temperature Sensor Module**

The time span for this module can be completed in half month.

**Heating Module**

The time span for this module can be completed in half month.

**Cooling Module**

The time span for this module can be completed in half month.

* 1. **Cost:** This project needs funds to reach the endpoint. Without adequate funds, this project cannot be successfully implemented. Cost estimation is one of the essential factors for any organization. So, calculating in advance the required funds for this project will be very impactful.
  2. **Scope:** The project is making use of a fan and a heater with the circuit systems in it. The scope of this project is:
* To design a system that sets the desired temperature value range.
* Automatic temperature sensor monitoring the recent change of temperature within the range.
* Heater/fan functioning within desired temperature range(s).
* Automatically sets the room temperature within 20 to 25 degrees Celsius.
  1. **Quality:** This project work presents the design of an automatic room temperature controller system. This system allows the user to set a desired temperature between (20’C to 25’C the normal temperature of room) which is then compared to the room temperature measured by TMP36 temperature sensor. With the help of Atmega 328P based on Arduino microcontroller, the system responds by turning ON and OFF the two (2) loads (fan or a heater) automatically depending on the temperature difference in the room. The fan is triggered ON when the room temperature is higher than the set temperature and the heater is triggered ON when the room temperature is lower than the set temperature. The system was designed and simulated using Tinkercad, circuit building software used for building electronics system. Tinkercad software was used to design and simulate the main circuit; Arduino IDE was used to write a program on how the system will operate which was then loaded to the microprocessor. An LCD was used to display temperature readings obtained by the TMP36 temperature sensor and the condition of the heater and the fan whether ON or OFF. When the system was simulated the heater and the fan were turned on automatically depending on the room temperature. The heater turned on when the temperature was below 20 degrees Celsius while the fan turned on when temperature fell below the threshold. In both cases, either of them was on at a specified time but not both. When temperature is between 20 to 25 degrees Celsius, the both heater and fan will OFF.
  2. **Benefits:** Here are some benefits of this project:
* The benefit of the system is less energy usage and it provides more convenience to the user
* It automatically manages temperature of the room.
* Less expensive then air-conditioned.
* It reduces electric power consumption.
* Easily install at any place.
  1. **Risk:** The project is generally based on forecasting. So, risk and uncertainty are always associated with projects. There will be a high degree of risk in the project which are not properly defined. Only the degree of control over risk and uncertainty varies with the project being conceived based on information available.

**Analysis of features and finalization subject to constraints**

Here are some of the features of this project that makes it unique:

* The features of the system are less energy usage and it provides more convenience to the user
* It automatically manages temperature of the room.
* Less expensive then air-conditioned.
* It reduces electric power consumption.
* Easily install at any place.

### 3.1. The triple constraints of project management

1. **Time constraint:** The time constraint refers to the project’s schedule for completion, including the deadlines for each phase of the project, as well as the date for rollout of the final deliverable.
2. **Scope constraint:** The scope of a project defines its specific goals, deliverables, features, and functions, in addition to the tasks required to complete the project.
3. **Cost constraint:** The cost of the project, often dubbed the project’s budget, comprises all of the financial resources needed to complete the project on time, in its predetermined scope.
   1. **Time constraint**

**Planning:** This project work presents the design of an automatic room temperature controller system. This system allows the user to set a desired temperature between (20’C to 25’C the normal temperature of room) which is then compared to the room temperature measured by TMP36 temperature sensor. With the help of Atmega 328P based on Arduino microcontroller, the system responds by turning ON and OFF the two (2) loads (fan or a heater) automatically depending on the temperature difference in the room. The fan is triggered ON when the room temperature is higher than the set temperature and the heater is triggered ON when the room temperature is lower than the set temperature. The system was designed and simulated using Tinkercad, circuit building software used for building electronics system. Tinkercad software was used to design and simulate the main circuit; Arduino IDE was used to write a program on how the system will operate which was then loaded to the microprocessor. An LCD was used to display temperature readings obtained by the TMP36 temperature sensor and the condition of the heater and the fan whether ON or OFF. When the system was simulated the heater and the fan were turned on automatically depending on the room temperature. The heater turned on when the temperature was below 20 degrees Celsius while the fan turned on when temperature fell below the threshold. In both cases, either of them was on at a specified time but not both. When temperature is between 20 to 25 degrees Celsius, the both heater and fan will OFF.

**Scheduling:**

In this project i.e., Automatic Room Temperature Controller there are five modules that are:

* Breadboard Supply Module
* LCD Display Module
* Temperature Sensor Module
* Heating Module
* Cooling Module

The total time span of this project is 2 and half months.

Fig 3.1



Fig 3.2: Gantt Chart



Fig 3.3: Gantt Chart

#### **Scope constraint**

The project is making use of a fan and a heater with the circuit systems in it. The scope of this project is:

* To design a system that sets the desired temperature value range.
* Automatic temperature sensor monitoring the recent change of temperature within the range.
* Heater/fan functioning within desired temperature range(s).
* Automatically sets the room temperature within 20 to 25 degrees Celsius.

#### **Cost constraint**

This project needs funds to reach the endpoint. Without adequate funds, this project cannot be successfully implemented. Cost estimation is one of the essential factors for any organization. So, calculating in advance the required funds for this project will be very impactful.

**Design selection**

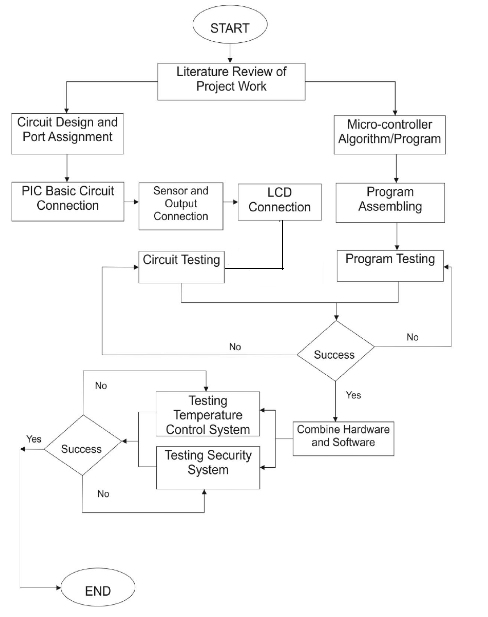


Fig 4.2. Flowchart for automatic room temperature controller

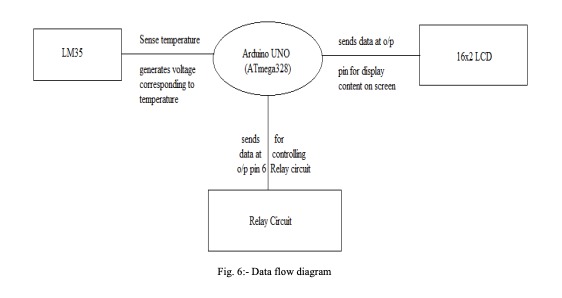
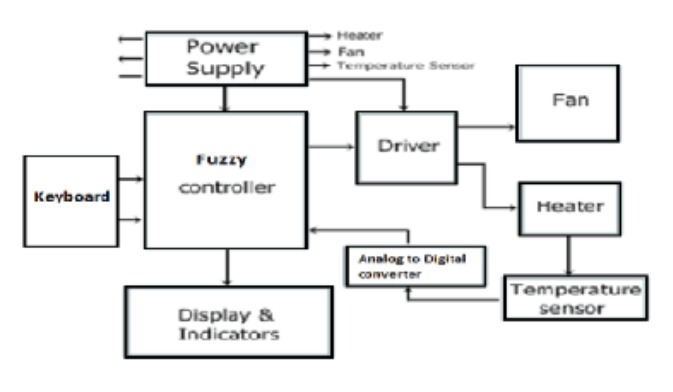


Fig 4.1. Data Flow Diagram



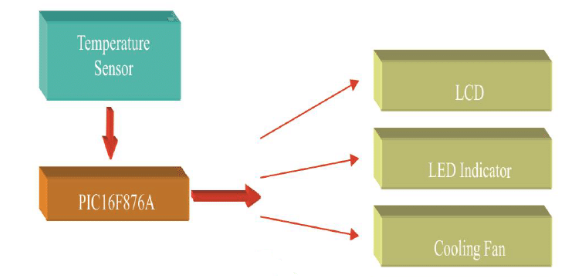


Fig 4.3. Block Diagram

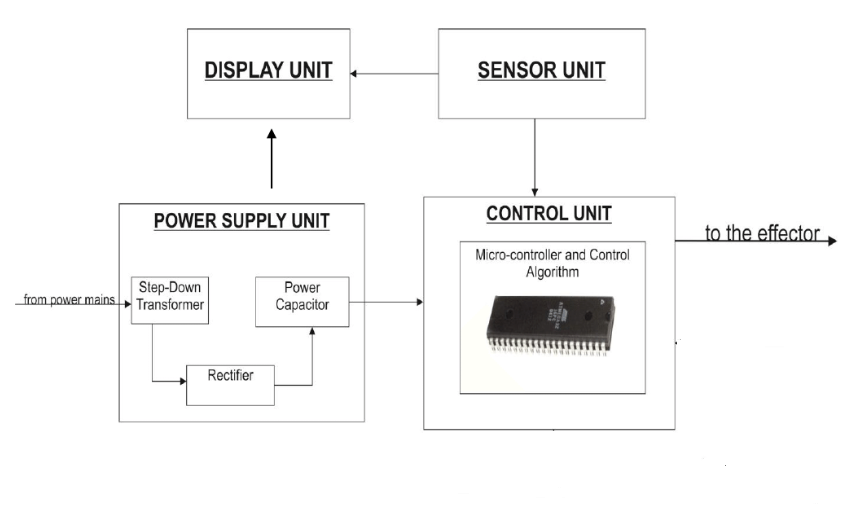


Fig 4.4. Project Channel for automatic room temperature controller

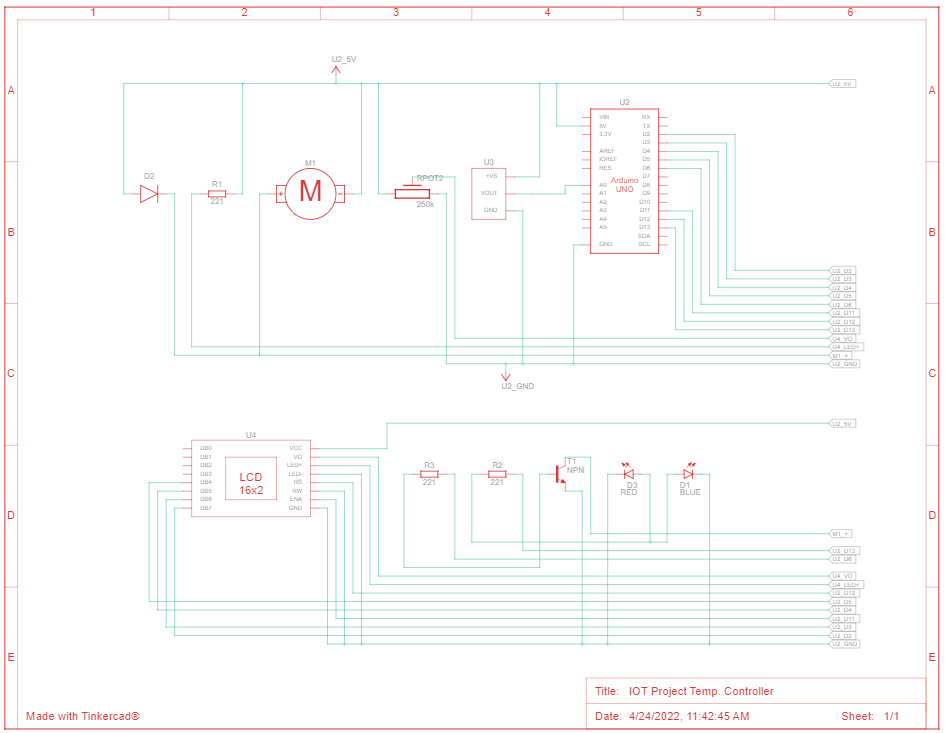


Fig 4.5. Circuit Diagram for automatic room temperature controller

**Final Outcome:**

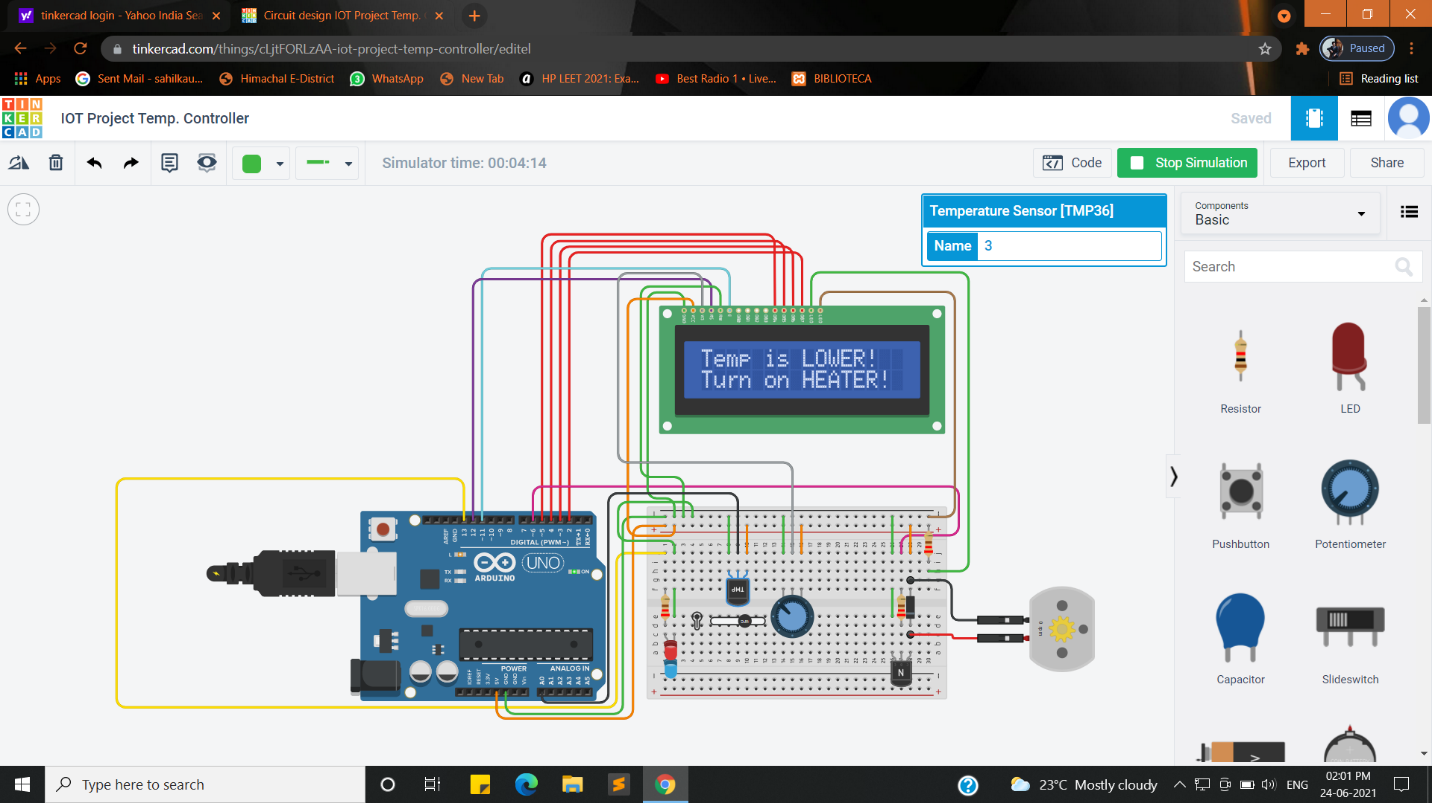
****

Fig 4.6. Circuit Implementation

**The Final Code for Automatic Room Temperature Controller:**

// Declaring Arduino IO-pins

const int temp\_trans\_pin = A0, Heater\_pin = 13, FAN\_pin = 6;

/\*For Fan:- DC Motor

For Heater:- LED\*/

// Setting the range of suitable temperature

float MinTemp = 20, MaxTemp = 25;/\*Room temperature is [20,25] degree C \*/

// Include the LCD library code

#include <LiquidCrystal.h>

// Initialize the library with the numbers of the interface pins

LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

void setup() {

// System initialization

LCD.begin(16, 2);

pinMode(Heater\_pin, OUTPUT);//LED in our case

pinMode(FAN\_pin, OUTPUT);// DC Motor in our case

// Displaying the suitable range of temperature

LCD.print("Room temp(C):");

LCD.setCursor(2,1);

LCD.print(MinTemp);

LCD.print("-");

LCD.print(MaxTemp);

delay(2000);

}

void loop() {

float Eqv\_volt, SensorTemp;

// Read voltage and convert to temperature (Celsius)

Eqv\_volt = analogRead(temp\_trans\_pin) \* 5.0 / 1023;

SensorTemp = 100.0 \* Eqv\_volt-50.0;

// Displaying the sensor reading

LCD.clear();

LCD.print("Sensor reading:");

LCD.setCursor(2,1);

LCD.print(SensorTemp);

LCD.print(" C");

delay(2000);

/\*Compare the sensor reading with the range of

acceptable temperatures\*/

if(SensorTemp > MaxTemp)

{

LCD.clear();

LCD.print("Temp is HIGHER!");//higher than the max

/\*Turn on FAN (dc motor)! to regulate the temp.

LCD.setCursor(0, 1);

LCD.print("Turn on FAN!");

for( int i = 0; i <= 255; i++ )

{

analogWrite(FAN\_pin, i);

}

delay(10000);

LCD.clear();

LCD.print("Temp is OK Now!");

LCD.setCursor(0, 1);

LCD.print("Turn off FAN!");

// Turn off FAN slowly

for( int i = 255; i >= 0; i-- )

{

analogWrite(FAN\_pin, i);

}

delay(2000);

}

else if(SensorTemp < MinTemp)

{

LCD.clear();

LCD.print("Temp is LOWER!");

LCD.setCursor(0, 1);

LCD.print("Turn on HEATER!");

//Turn the heater ON, LED in our case

digitalWrite(Heater\_pin, HIGH);

delay(10000);

LCD.clear();

LCD.print("Temp is OK Now!");

LCD.setCursor(0, 1);

LCD.print("Turn off HEATER!");

delay(1000);

digitalWrite(Heater\_pin, LOW);

LCD.clear();

}

else if(SensorTemp > MinTemp && SensorTemp < MaxTemp){/\*Now temperature is perfect.

That is,it is in the desired range. Hence no need of changes!!\*/

LCD.clear();

LCD.print("Temp is NORMAL!");

LCD.setCursor(2,1);

LCD.print("Turn off all!");

delay(1000);

LCD.clear();

}

else {

LCD.clear();

LCD.print("Something went");

LCD.setCursor(2,1); LCD.print("WRONG in the ckt");

delay(1000);

LCD.clear();

}

delay(1000);

}

**References**

The web sites that provide the information’s:

[www.microchip.co.uk](http://www.microchip.co.uk)

[www.quora.com](http://www.quora.com)

[www.geeksforgeeks.com](http://www.geeksforgeeks.com)

[www.wikipedia.com](http://www.wikipedia.com)

[www.engineeringproject.com](http://www.engineeringproject.com)

[www.electronicsworldforyou.com](http://www.electronicsworldforyou.com)

[www.tinkercad.com](http://www.tinkercad.com)

[www.arduinoide.com](http://www.arduinoide.com)

Arduino. (2018). Arduino - Software. <https://doi.org/28/12/2008s>