Design

Name: - Sanskruti Shimple

Div: - TY – IC – B

Roll No: - 41

PRN No: - 12011236

Subject: - Measurement Systems

Type: - Home Assignment

Topic: - IOT Based on Temperature Measurement System

IoT Based on Temperature Measurement System

Temperature Measurement and Control Systems

The temperature measurement and control system allows intelligent temperature sensor programming and operating study and controls temperature using a programmable logic controller. The sensor programming involves choosing the temperature unit, output function, and type, but also operating functions for switching outputs, temperature thresholds for both outputs, value domain of output current, etc. The sensor functioning for various settings can be verified through the manually controlled voltage supply of the power resistor and the cooler. As an application of an intelligent temperature sensor, an on-off temperature control system is implemented using a programmable logic controller. The experimental system presented in this paper has various benefits in engineering education: the students can work with a real temperature probe and an intelligent sensor, and they can implement and test an on-off temperature control system.

Introduction

It is crucial for many activities and tasks to be completed, such as in any business where heaters are used, heat up to a certain degree is necessary, and temperature measurement is the approach that is utilized. A temperature sensor that is put at the location whose temperature needs to be sensed is employed when it comes to temperature sensing. Through the internet of things, the temperature of that location may be tracked online.

Applications, where monitoring is used, include temperature, pressure, flow rate, capacity, acceleration, and many more. There

are various monitoring techniques to gather measurements depending on the sizes, distributions, and frequency of the observed items. When a room's temperature is monitored, several issues frequently arise. For instance, a server room needs to be monitored for temperature and kept between 15 and 20 degrees Celsius, or else the server can crash and result in a loss of hundreds of thousands of dollars. Management must decide to save money by creating a system to hire a person to keep track of the temperature or that can do so whenever necessary from other locations.

Components

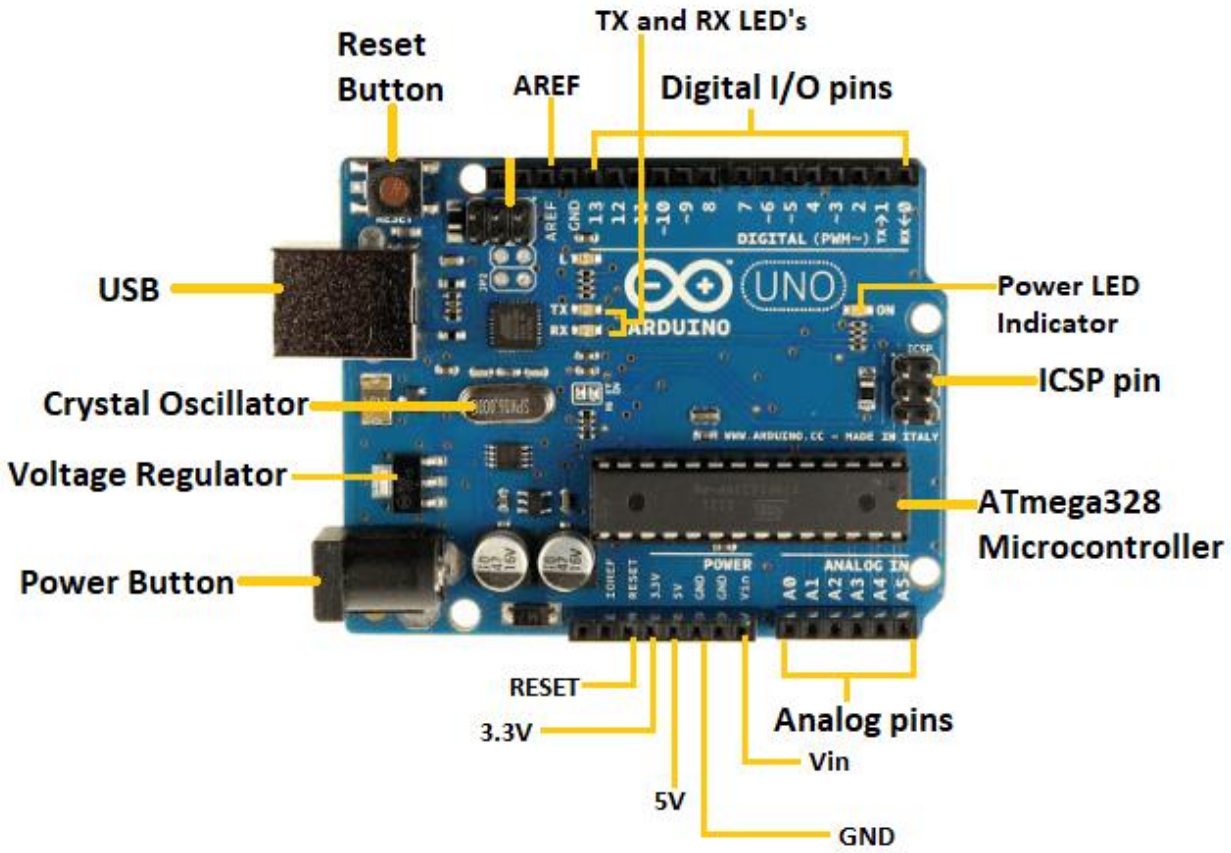
The components used in the design of the Temperature Measurement System are as follows,

1) Arduino Uno
2) L293D Motor Driver
3) LM35 Temperature Sensor
4) LM016L 16*2 Alphanumeric LCD
5) Simple Motor
    i. FAN
    ii. HEATER
6) LED-Blue
7) LED-Green
8) LED-Red
9) 1K 10WATT Resistor

Arduino Uno

The Arduino UNO is one of the company's standard boards. The Italian word UNO here is for "one." To identify the initial release of the Arduino Software, it was given the moniker UNO. It was also the first USB board that Arduino had ever released. It is regarded as a strong board that is employed in many projects. The Arduino UNO board was created by Arduino. cc.

The ATmega328P microprocessor is the foundation of the Arduino UNO. Compared to other boards, like the Arduino Mega board, etc., it is simple to use. The board is made up of shields, various circuits, and digital and analog Input/Output (I/O) pins.



The Arduino UNO has 14 digital pins, a USB port, a power jack, and an ICSP (In-Circuit Serial Programming) header in addition to 6 analog pin inputs. The programming language used is called IDE, or integrated development environment. It is compatible with offline and online platforms.

L293D Motor Driver

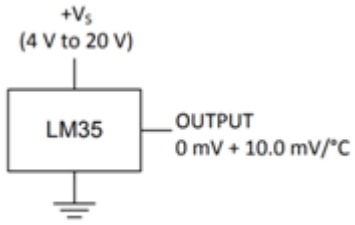
A motor driver is known as L293d IC. Like other ICs, it operates at low voltage. The other ICs might perform the same tasks as L293d, but they are unable to supply the motor with the required voltage. The Motor receives a constant, bidirectional direct current from L293d. Without affecting the entire IC or any other component in the circuit, the polarity of the current can change at any time. L293d has two motors and an internal H-bridge fitted.

An electrical circuit known as an H-Bridge allows the load to operate in both directions. Low voltage signals coming from outside sources regulate the L293d bridge. Despite being small, it has a larger power output capacity than we had anticipated. With a voltage range of 4.5 to 36 Volts, it could control the speed and direction of any DC motor. Additionally, its diodes protect the IC and controlling device from back EMF. It has a built-in "Darlington transistor sink" that can be used to control a large amount of current by supplying a small amount of current to regulate the maximum 600mA amount of current. Additionally, it has an internal "pseudo-Darlington source" that amplifies the input signal so that it can safely control the high voltage DC motor.

LM35 Temperature Sensor

The output voltage of the precession integrated circuit temperature sensor LM35 fluctuates according to the ambient temperature. It is a small, inexpensive IC that can sense temperatures between -55°C and 150°C. Any microcontroller with an ADC function or any type of development platform, such as Arduino, can easily be interfaced with it.

The IC is powered by connecting the ground pin to the circuit's ground and delivering a regulated voltage to the input pin, such as +5V (VS). Now, as seen below, you may measure temperature as voltage.



The output voltage will be 0V if the temperature is 0°C. For every degree Celsius that the temperature rises, there will be a rise of 0.01V (10mV). The formulae shown below can be used to convert voltage to temperature.

V_{OUT} = 10 mv/°C × T

where

- V_{OUT} is the LM35 output voltage
- T is the temperature in °C

LM35 Sensor Specifications

- 35V and -2V are the minimum and maximum input voltages, respectively. Normally, 5V.
- can record temperatures between -55°C and 150°C
- A rise of 10mV (0.01V) will occur for every 1°C increase in temperature since the output voltage is exactly proportional (linear) to temperature.
- ±0.5°C Drain current accuracy is less than 60uA
- cost-effective temperature sensor
- Small, making it a good fit for remote applications
- readily accessible in the TO-92, TO-220, TO-CAN, and SOIC packages

Simulation Of Temperature Measurement System

The prototype of the system is developed using the Proteus Software

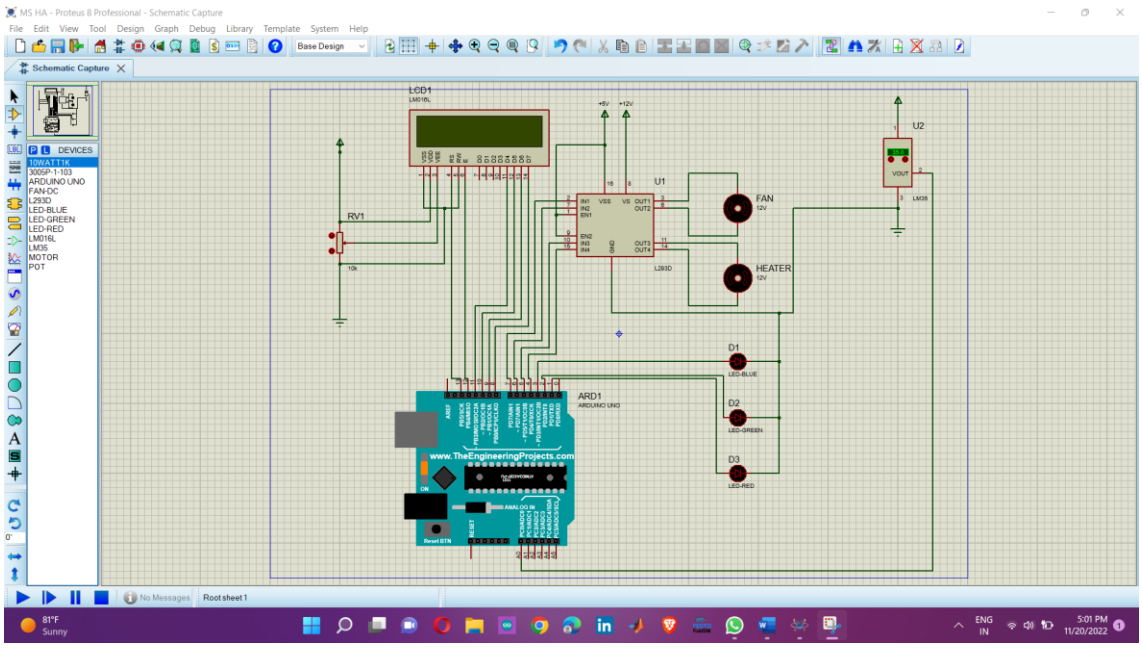


Fig.1

Condition 1

If the temperature range is between 20℃ – 35℃ the heater and fan are at a steady state and hence, the green LED will give an indication of a steady state.

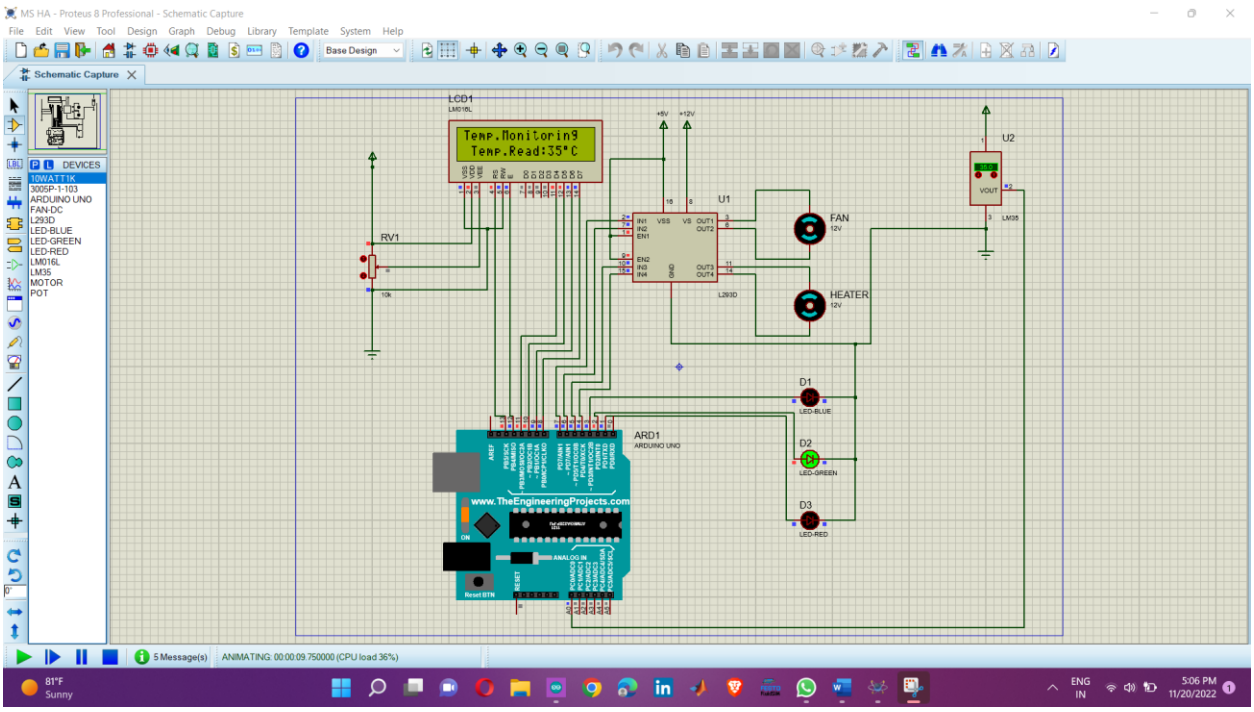


Fig.2

Condition 2

If the temperature range is below 20℃, then the heater will be in an active state and the fan is in a steady state hence, the blue LED will give an indication of an active state.

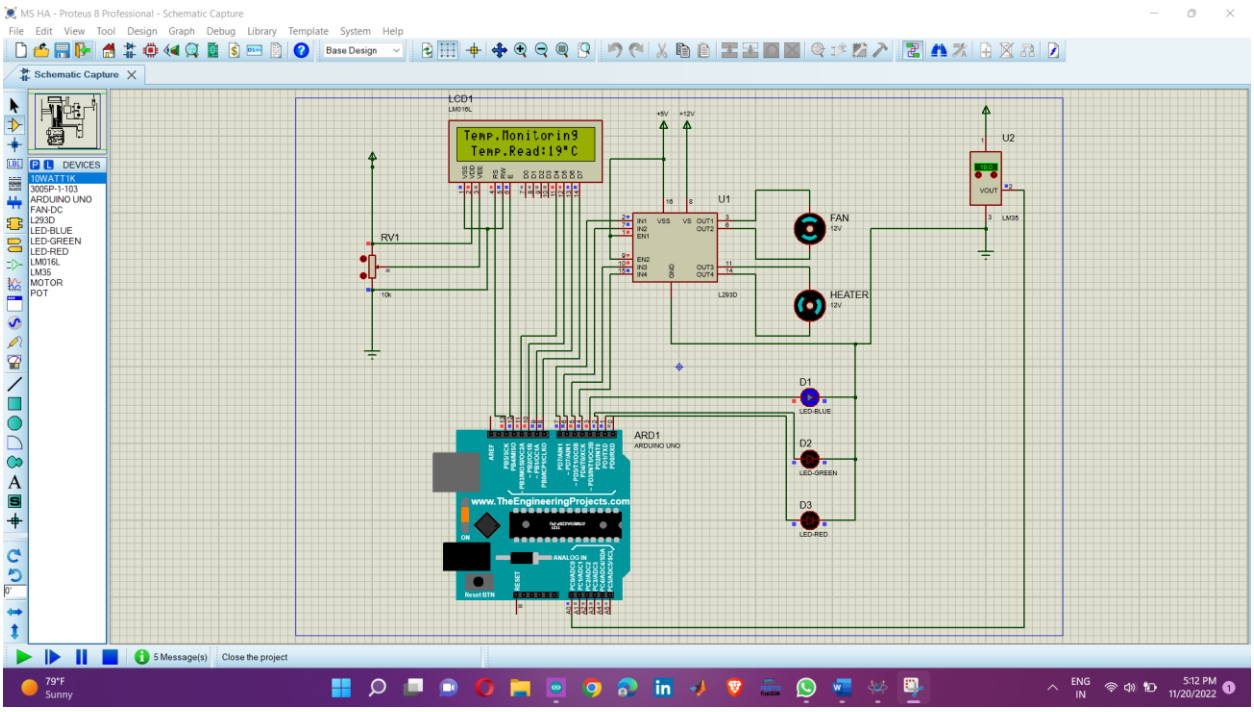


Fig.3

Condition 3

If the temperature range is above 35℃, then the heater will be in a steady state and the fan is in an active state hence, the red LED will give an indication of an active state.

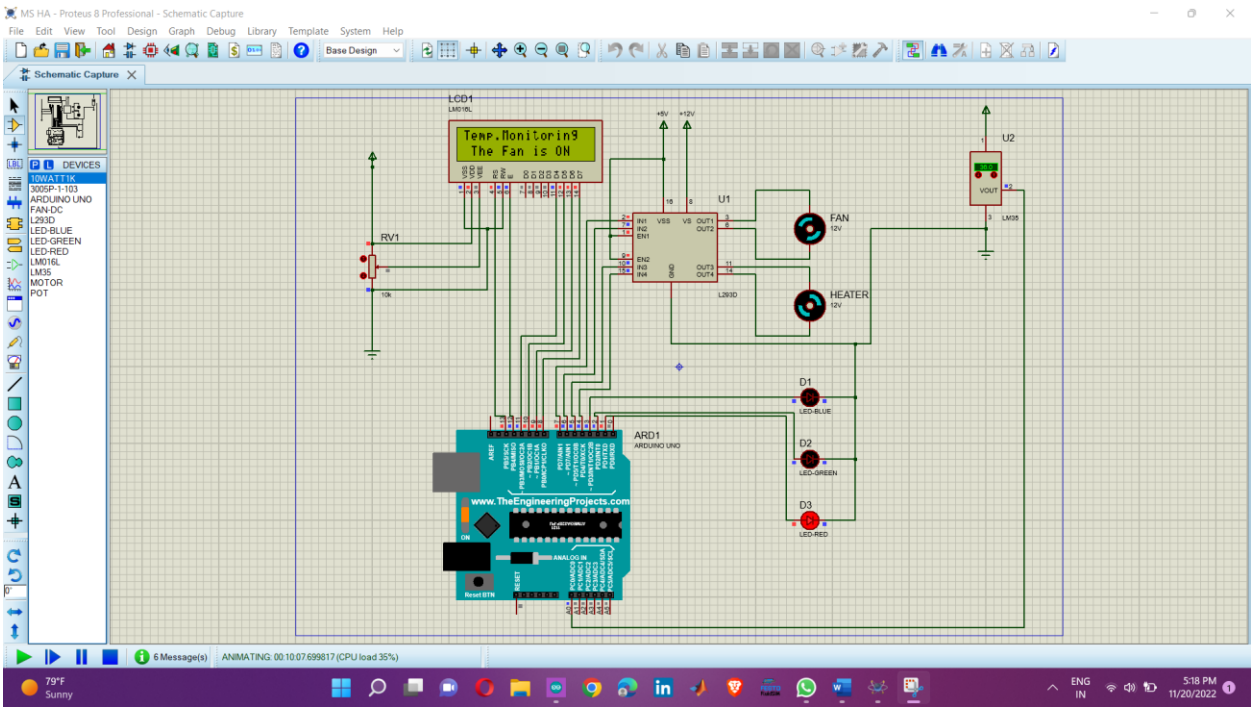


Fig.3

Code

The prototype of the system code is developed using the Arduino IDE Software.

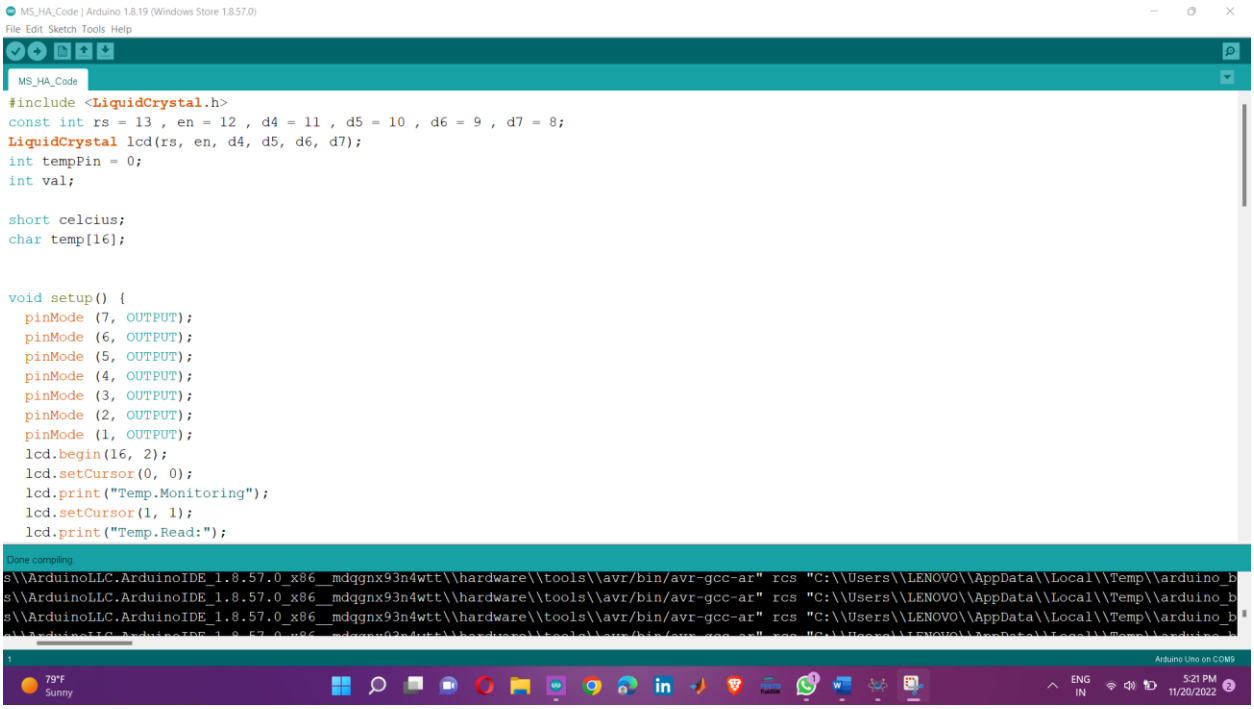


Fig.3

Code

```
#include <LiquidCrystal.h>
const int rs = 13 , en = 12 , d4 = 11 , d5 = 10 , d6 = 9 , d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int tempPin = 0;
int val;


short celcius;
```

```
char temp[16];


void setup() {
  pinMode (7, OUTPUT);
  pinMode (6, OUTPUT);
  pinMode (5, OUTPUT);
  pinMode (4, OUTPUT);
  pinMode (3, OUTPUT);
  pinMode (2, OUTPUT);
  pinMode (1, OUTPUT);
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("Temp.Monitoring");
  lcd.setCursor(1, 1);
  lcd.print("Temp.Read:");
}

void loop() {
  Readtemp();
  //NORMAL TEMP
```

```
if (celcius <= 35 && celcius >= 20) {
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(1, LOW);
  digitalWrite(7, LOW);
  digitalWrite(6, LOW);
  digitalWrite(5, LOW);
  digitalWrite(4, LOW);
}
//HIGH TEMP
else if (celcius > 35) {
  FanON();
  digitalWrite(1, HIGH);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(5, LOW);
  digitalWrite(4, LOW);
  delay(800);
  lcd.setCursor(0, 1);
  lcd.print("                ");
  lcd.setCursor(1, 1);
```

```
    lcd.print("The Fan is ON");

  }
  //LOW TEMP
  else if (celcius < 20) {
    HeaterON();
    digitalWrite(3, HIGH);
    digitalWrite(2, LOW);
    digitalWrite(1, LOW);
    digitalWrite(7, LOW);
    digitalWrite(6, LOW);
    delay(800);
    lcd.setCursor(0, 1);
    lcd.print("                ");
    lcd.setCursor(0, 1);
    lcd.print("The Heater is ON");

  }

}
```

```
void Readtemp() {
  delay(500);
  lcd.setCursor(0, 1);
  lcd.print("                ");
  val = analogRead(tempPin);
  celcius = val*4.88/10;
  sprintf(temp, "%0.2d", celcius);
    lcd.setCursor(1, 1);
    lcd.print("Temp.Read:");
    lcd.print(temp);
    lcd.write(B11011111);
    lcd.print("C");
}

void FanON() {
  digitalWrite(7, HIGH);
  digitalWrite(6, LOW);
  delay(100);
}

void HeaterON() {
```

```
  digitalWrite(5, HIGH);
  digitalWrite(4, LOW);
  delay(100);
}
```

Conclusion

The chemical makeup or integrity of volatile chemicals and biological samples, respectively, are greatly influenced by temperature. If not maintained within a strict temperature range, they can become ineffective, impact experimental results, or even worse, harm the patient's health. To successfully test these samples, it is crucial for laboratories and testing facilities to maintain a regulated atmosphere.

A laboratory can perform tests and store samples in a controlled environment with the help of an IoT temperature monitoring solution, leading to accurate results.