NEHRU ARTS AND SCIENCE COLLEGE

BATCH:6

KANAGA DURGA G (23UGIT024)

KANNAN N (23UGIT025)

LAYA SANKAR (23UGIT026)

MALAISELVAM M (23UGIT027)

EXECUTION VIDEO OF THE DEMO :

GITHUB LINK: https://github.com/Malaiselvam6300/L-T-project-batch-6

MONITOR THE TEMPERATURE DISPLAY A WARNING MESSAGE WHEN THE TEMPERATURE EXCEEDS A PREDEFINED LIMIT

AIM:

The aim of this project is to design and simulate a real-time clock and temperature monitoring system that continuously monitors environmental temperature and displays a warning message when the temperature exceeds a predefined limit. This will allow users to have real-time temperature feedback and receive alerts for safety or operational purposes.

PROBLEM STATEMENT:

With growing environmental concerns and the need for temperature regulation in various systems (such as industrial machines, servers, or smart homes), there is a need to design a real-time monitoring system that can effectively track temperature and provide alerts when it exceeds a specific threshold. This solution will ensure timely actions can be taken to avoid overheating or other temperature-related issues.

SCOPE OF THE SOLUTION

This project will provide:

(a) **Real-time temperature monitoring:** The system will read temperature data using a temperature sensor (like a DHT11 or LM35).

(b) **Clock synchronization:** The system will use a real-time clock module (like DS3231) to log the temperature with accurate timestamps**.**

(c) **Warning mechanism:** When the temperature exceeds the predefined limit, the system will trigger a warning message (can be visual or text-based).

(d) Alert system: The warning will be displayed on an LCD screen or sent as a notification (in advanced versions).

REQUIRED COMPONENTS TO DEVELOP THE SOLUTION

(a) Hardware:

(b) Arduino Uno (or similar microcontroller board)

(c) DHT22 or LM35 Temperature Sensor

(d) DS3231 Real-Time Clock (RTC) Module

(e) **LCD Display** (16x2 or similar for showing real-time data)

(f) Buzzer (for auditory alert)

(g) **LED** (for visual indication)

(h) Jumper Wires and Breadboard

(i) Software:

(j) **Arduino IDE** for programming the microcontroller

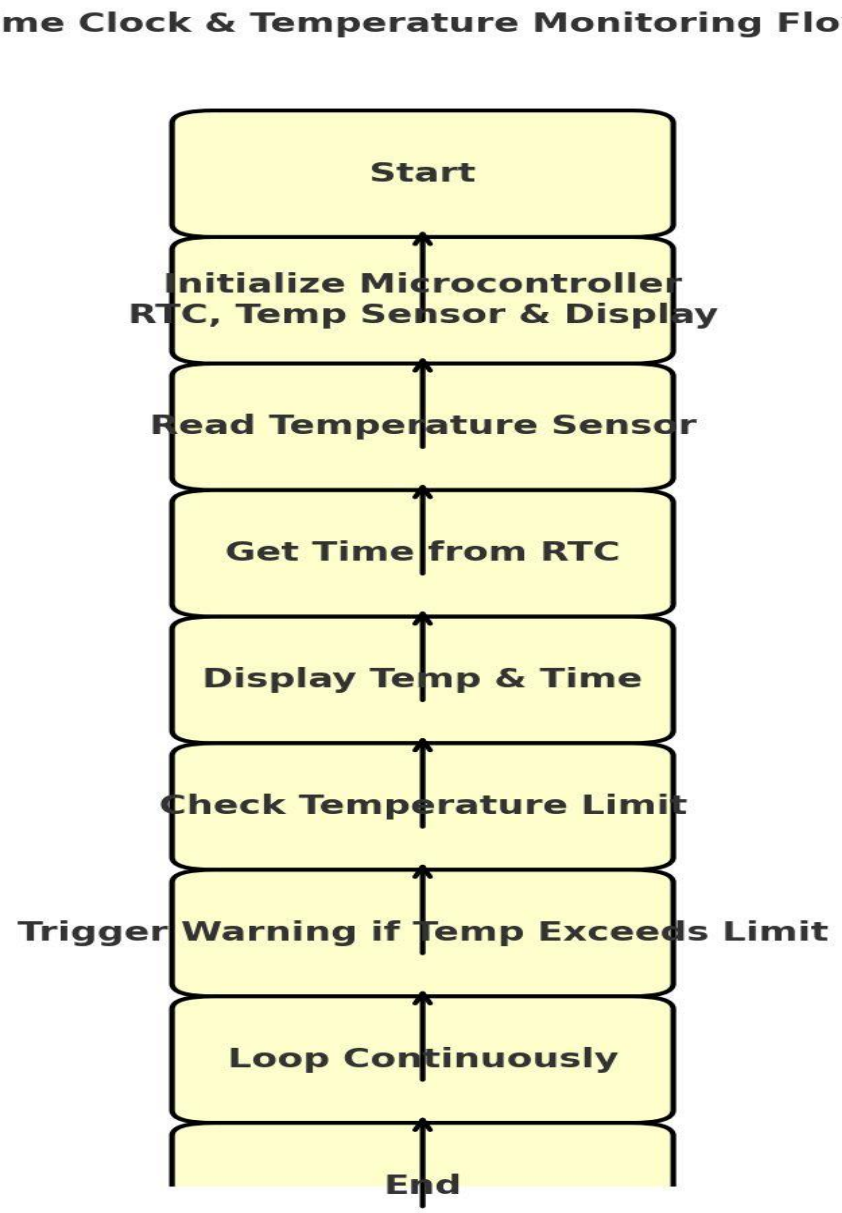
(k) Wokwi Simulator for simulating the circuit (if you want a cloud-based platform for simulation)

(l) Cloud Environment (Optional):

(m) ThingSpeak (for storing and displaying temperature data online)

(n) IFTTT (for sending notifications or emails if temperature exceeds the limit)

FLOW CHART:



The flowchart represents the step-by-step working of the system:

Start – The system begins its operation.

Initialize Microcontroller, RTC, Temperature Sensor & Display – The necessary components are set up, including the real-time clock (RTC) module, temperature sensor, and display module.

Read Temperature Sensor – The system reads the current temperature from the sensor (e.g., DHT11/DHT22).

Get Time from RTC – The system fetches the current time from the RTC module (e.g., DS3231).

Display Temp & Time – The retrieved temperature and time are displayed on the screen (LCD/OLED).

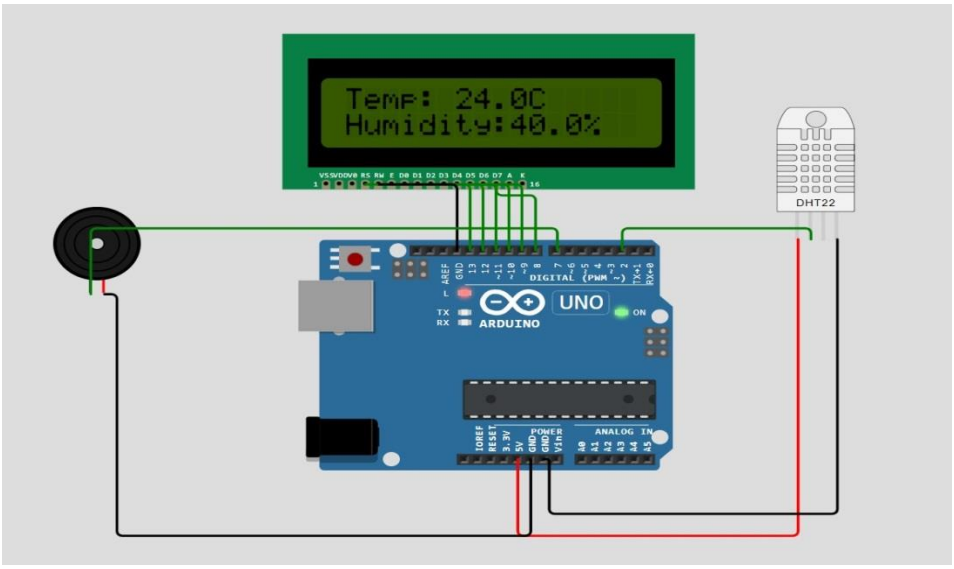
Check Temperature Limit – The system compares the temperature with a predefined threshold.

Trigger Warning if Temp Exceeds Limit – If the temperature exceeds the set limit, a warning message is displayed, and an alarm (buzzer/LED) is triggered.

Loop Continuously – The system continuously monitors temperature and time in real time.

End – The process continues indefinitely unless manually stopped.

SIMULATED CIRCUIT (WOKWI):



VIDEO OF THE DEMO:

https://github.com/Malaiselvam6300/L-T-project-batch-6