Aim

To design and simulate a **real-time clock (RTC) and temperature monitoring system** that continuously checks the temperature and displays a **warning message** if the temperature exceeds a predefined limit.

Problem Statement

Industries, server rooms, and labs require continuous temperature monitoring to prevent equipment damage due to overheating. Manual monitoring is inefficient and prone to errors. This project aims to develop a **microcontroller-based system** that:

- Monitors temperature in real-time using a temperature sensor.
- Displays the current time and temperature.
- Generates a **visual warning message** when the temperature exceeds a set threshold.

Scope of the Solution

- Continuously monitor ambient temperature and display it on an LCD.
- Integrate an RTC module to display current time.
- Compare temperature readings with a predefined threshold.
- Display a **warning message on the LCD** if the temperature exceeds the threshold.
- Simulate using **Wokwi for validation** before hardware implementation.
- Can be expanded to add buzzer alerts, IoT logging, or relay control for cooling fans.

Required Components

Hardware:

- Arduino Uno
- DS3231 RTC Module
- LM35 or DHT22 temperature sensor

- 16x2 LCD display (I2C preferred for simplicity)
- Buzzer (optional for alert)
- Jumper wires

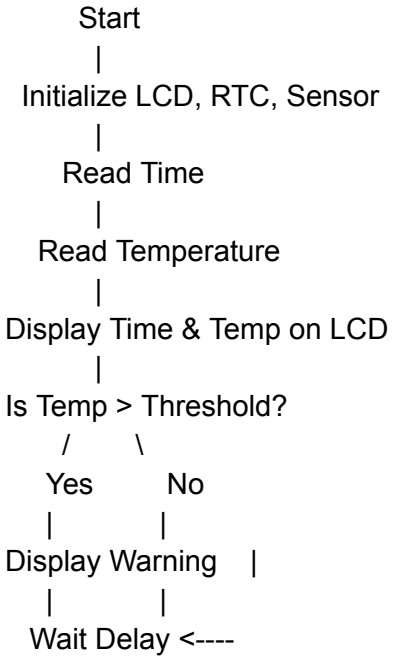
Software:

- **Arduino IDE** (code development and uploading)
- **Wokwi** (for simulation)

Cloud:

- GitHub (project documentation and code hosting)

Flowchart of the Code



Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
#include "RTClib.h"
#include "DHT.h"

// LCD setup
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27

// RTC setup
RTC_DS3231 rtc;

// DHT setup
#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

// Threshold temperature (°C)
#define TEMP_THRESHOLD 30.0

void setup() {
  lcd.init();
  lcd.backlight();

  dht.begin();

  if (!rtc.begin()) {
    lcd.print("RTC not found!");
    while (1);
  }
}

void loop() {
  // Read time
  DateTime now = rtc.now();

  // Read temperature
  float temp = dht.readTemperature();

  lcd.clear();

  // Display time
  lcd.setCursor(0, 0);
  lcd.print(now.hour());
  lcd.print(":");
  if (now.minute() < 10) lcd.print("0");
  lcd.print(now.minute());
  lcd.print(":");
  if (now.second() < 10) lcd.print("0");
  lcd.print(now.second());
```

```
  // Display temperature
  lcd.setCursor(0, 1);
  lcd.print("Temp: ");
  lcd.print(temp);
  lcd.print(" C");

  // Check threshold
  if (temp > TEMP_THRESHOLD) {
    delay(2000); // Show reading before warning
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WARNING:");
    lcd.setCursor(0, 1);
    lcd.print("HIGH TEMP!");
    delay(2000);
  } else {
    delay(1000);
  }
}
```