Smart Home Temperature and Humidity Monitoring System

Abstract

This report presents the design and development of a Smart Home Temperature and Humidity Monitoring System. The system is built to monitor and control the indoor environment by collecting real-time data from temperature and humidity sensors. The system displays real-time values on an LCD screen and provides remote monitoring through a smartphone application via Wi-Fi. Moreover, it automatically controls heating or cooling appliances based on pre-set thresholds, ensuring optimal indoor climate conditions. Key challenges include integrating wireless communication and ensuring energy efficiency. The report outlines the system design, hardware-software integration, and future enhancements.

1 Objectives

The primary objectives of the Smart Home Temperature and Humidity Monitoring System are:

- Measure and monitor temperature and humidity levels inside the home using sensors.
- Display real-time data on an LCD screen for immediate feedback.
- Enable wireless communication via Wi-Fi to allow remote monitoring and control through a smartphone application.
- Automatically control appliances such as air conditioners (AC) and heaters based on pre-configured temperature and humidity thresholds.

2 Scope of Work

The scope of the system includes the following:

- The system is designed to monitor the temperature and humidity of a single indoor space.
- Wireless communication is limited to monitoring and controlling the system via a smartphone app.
- Automatic control of one air conditioning unit or heater is provided.
- Temperature and humidity thresholds must be configured manually by the user.
- This system does not feature machine learning or AI for predictive temperature control.

3 Block Diagram

The block diagram in Figure 1 illustrates the key components of the Smart Home Temperature and Humidity Monitoring System and their interactions.

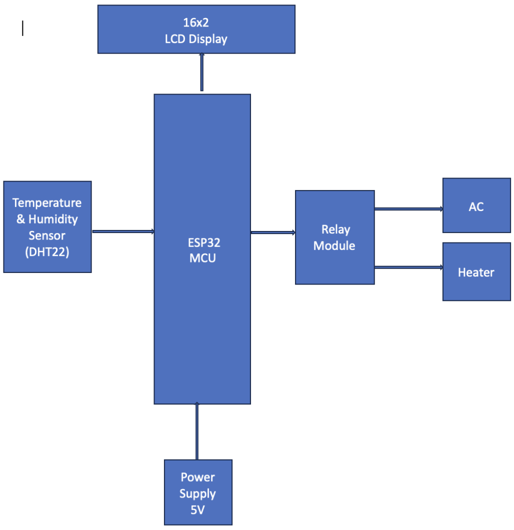


Figure 1: Block Diagram of the Smart Home Temperature and Humidity Monitoring System

4 List of Hardware Components

The system utilizes the following hardware components:

- **Microcontroller**: ESP32 (with integrated Wi-Fi for wireless communication).
- **Temperature and Humidity Sensor**: DHT22 sensor.
- **LCD Display**: 16x2 LCD to display real-time sensor data.
- **Relay Module**: Used to control AC or heater.
- **Power Supply**: 5V regulated power supply.

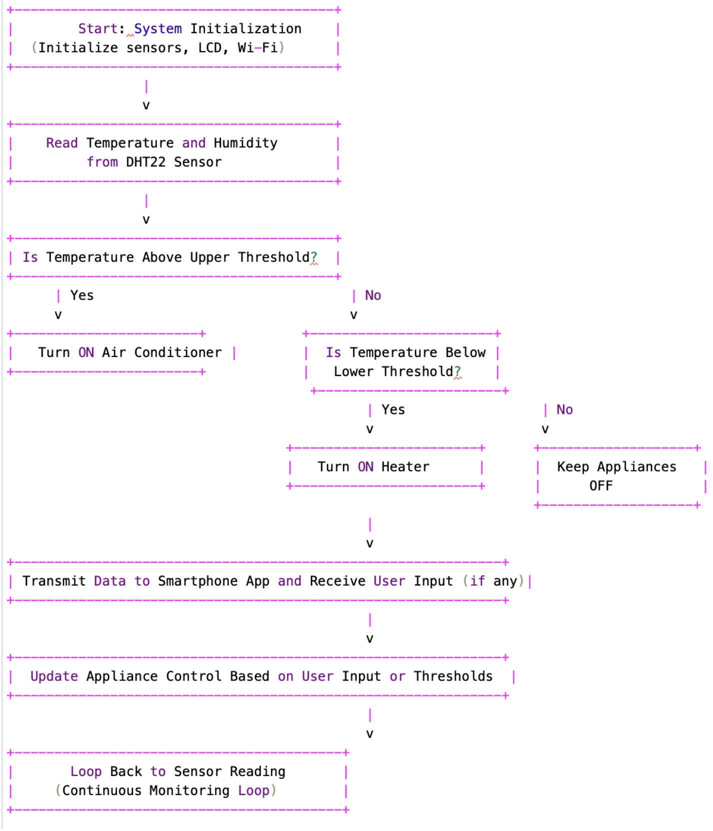
5 List of Software Components

The software components required to run the system include:

- **Embedded Code**: Written in C/C++ for controlling the ESP32 microcontroller and sensors.
- **Wi-Fi Communication Protocol**: HTTP or MQTT for communicating with the smartphone app.
- **User Interface**: Mobile app for displaying temperature and humidity data and allowing remote control of the system.
- **Sensor and Display Driver Code**: For reading sensor values and driving the LCD display.

6 Flowchart/Algorithm

The control logic for the system is explained in the flowchart (Figure 2).



```
+------------------------------------------+
|        Start: System Initialization      |
|  (Initialize sensors, LCD, Wi-Fi)        |
+------------------------------------------+
                  |
                  v
+------------------------------------------+
|    Read Temperature and Humidity         |
|         from DHT22 Sensor                |
+------------------------------------------+
                  |
                  v
+------------------------------------------+
| Is Temperature Above Upper Threshold?    |
+------------------------------------------+
      | Yes                                 | No
      v                                     v
+-------------------------+           +-------------------------+
|   Turn ON Air Conditioner |         |  Is Temperature Below |
+-------------------------+           |   Lower Threshold?    |
                                       +-------------------------+
                                               | Yes                   | No
                                               v                       v
                                   +-------------------------+  +---------------------+
                                   |   Turn ON Heater        |  |  Keep Appliances    |
                                   +-------------------------+  |       OFF           |
                                                                +---------------------+
                                               |
                                               v
+----------------------------------------------------------------+
| Transmit Data to Smartphone App and Receive User Input (if any)|
+----------------------------------------------------------------+
                                               |
                                               v
+----------------------------------------------------------------+
|  Update Appliance Control Based on User Input or Thresholds   |
+----------------------------------------------------------------+
                                               |
                                               v
+-------------------------------------------+
|        Loop Back to Sensor Reading        |
|     (Continuous Monitoring Loop)          |
+-------------------------------------------+
```

Figure 2: Flowchart of System Control Logic

Steps in Flowchart

System Initialization

The system first initializes all hardware components including sensors (D-HT22), the LCD screen, and the Wi-Fi module for communication.

Read Sensor Data

The ESP32 microcontroller reads the temperature and humidity data from the DHT22 sensor.

Check Temperature

- If the temperature exceeds the preset upper threshold, the air conditioner (AC) is turned on.
- If the temperature is below the lower threshold, the heater is turned on.
- If the temperature is within the threshold range, no action is taken and appliances remain off.

Transmit Data

The system transmits the collected data to a smartphone app via Wi-Fi. The system also listens for control commands from the app.

Update Appliance Control

Based on either the user input from the smartphone app or the temperature thresholds, the system updates the control of the air conditioner or heater.

Loop Back

The system continuously loops back to read sensor data, allowing for real-time monitoring and control.

This flowchart shows the sequence of operations for the control logic, ensuring that the system monitors environmental conditions and takes appropriate action based on temperature thresholds and user input.

The pseudocode for the control algorithm is provided below:

```
BEGIN
    Initialize sensors, LCD, and Wi-Fi module
    Connect to Wi-Fi network
    WHILE system is active:
        Read temperature and humidity data from DHT22
        Display data on LCD
        IF temperature > upper threshold:
            Activate air conditioner
        ELSE IF temperature < lower threshold:
            Activate heater
        Transmit sensor data to the smartphone app
        Receive user input (if any) from smartphone app
        Adjust AC/heater control based on user input
    END WHILE
END
```

7 Communication Protocols

The system uses several communication protocols:

- **I2C**: For communication between the ESP32 microcontroller and the LCD display.
- **Wi-Fi (HTTP/MQTT)**: Used to send temperature and humidity data to a smartphone app and receive control inputs remotely.
- **GPIO**: Used to control the relay module that activates the air conditioner or heater.

8 Integration of Hardware and Software

The hardware and software integration is achieved through the ESP32 microcontroller, which manages communication between the sensors, LCD, and the Wi-Fi module. The sensor data is collected and processed by the microcontroller, which then displays the information on the LCD and sends it via Wi-Fi to a smartphone app. Control signals from the app are processed by the microcontroller to switch appliances on or off through the relay module.

9 Data Flow

The system's data flow involves:

1. The DHT22 sensor continuously measures temperature and humidity.
2. The ESP32 reads and processes this sensor data.
3. Data is displayed on the LCD in real-time for local monitoring.
4. The ESP32 sends the data via Wi-Fi to a smartphone app for remote monitoring.
5. If the temperature exceeds or falls below predefined thresholds, the ESP32 activates the relay module to control the AC or heater.
6. The smartphone app can override the automatic controls by sending commands to the ESP32.

10 Future Improvements

Potential future improvements to the system include:

- Integration with other smart home devices such as lights and window blinds for holistic home automation.
- Adding machine learning algorithms to predict and optimize temperature and humidity control based on user preferences and environmental conditions.
- Extending the system to control multiple rooms or zones within the house.
- Implementing energy-saving features such as solar power integration or efficient power management systems.

11 Conclusion

The Smart Home Temperature and Humidity Monitoring System provides an effective way to monitor and control indoor climate, ensuring user comfort and energy efficiency. By integrating sensors, wireless communication, and automation, the system can improve quality of life in smart homes. Future expansions can make the system more versatile and energy-efficient, enhancing its usefulness in modern homes.