Smart Home Temperature Alert System

Emir Aksakal 19COMP1030 **Rana Selim 19COMP1009**
Computer Engineering Department, ISIK University
19COMP1030@isik.edu.tr 19COMP1009@isik.edu.tr

1 Introduction to the Problem

Temperature sensors are an important IoT device as they allow for real-time monitoring of temperature in various settings. In a home, for example, a temperature sensor placed in a child's room can help parents ensure that the room remains at a comfortable and safe temperature. This can be especially important for children, babies or animals who are more susceptible to temperature-related health issues. Additionally, temperature sensors can be used to monitor the temperature of refrigerators, which can help detect if the refrigerator is malfunctioning and needs to be repaired. This can prevent food spoilage, save money and prevent food-borne illness. With IoT technology, the temperature sensor can be connected to a smartphone or other device and allow for instant notifications of any temperature changes, providing peace of mind and allowing for quick action to be taken if necessary.

1.1 Use Cases

Example Use Case 1: If the user has a baby, they want the most suitable temperature in the room where their baby is. If the temperature is not within a certain value range, this situation is notified to the user via e-mail.

Example Use Case 2: When the user is not at home, he is curious about the state of the house. There may be a fire in the house. In such cases, if the temperature is above a threshold, the user is notified via e-mail.

Example Use Case 3: If the user owns a refrigerator and is concerned about the condition of the refrigerator, the user will be notified if the device attached to the refrigerator is not within a certain value range.

2 Your Method to Solve the Problem

As a solution for our project, we designed a hardware that measures temperature using a temperature sensor, NodeMCU and a mail address, we warn the user by sending e-mails when the temperature exceeds the threshold value that we evaluate.

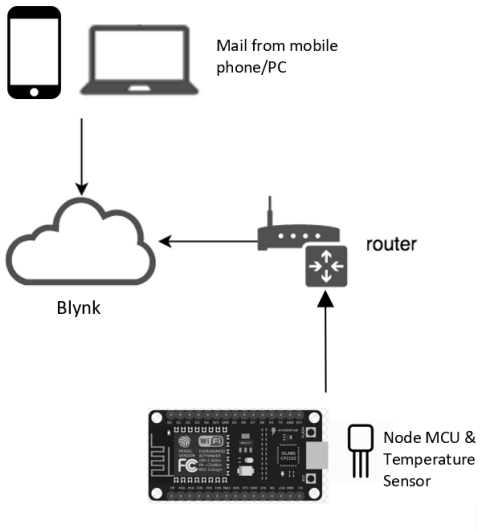

Figure 1: The proposed system overview

2.1 Hardware Design

Hardware Specifications:
- Node MCU ESP 8266 with CP 2102
- DHT11 Temperature Sensor
- Wi-Fi Connection
- USB cable
- Laptop

	Instant monitoring	E-mail
5°	Yes	None
10°	Yes	None
20°	Yes	None
27°	Yes	None
29°	Yes	None
30°	Yes	Sending
35°	Yes	Sending

Table 1: In case threshold value is 30°

2.2 Software Design

- Ardunio IDE
- ESP8266WiFi Library
- BlynkSimpleEsp8266 Library
- DHT Library
- DHT Library

2.2.1 Functional Requirements

1) Basic understanding of app UI (User Interface) for any kind of user and age.

2) An email format that is basic but designed for emergency situations to alert user.

3) The system should get values from temperature sensor and send them to the app of the user.

4) The system must warn user if the values getting from temperature sensor has exceeded the threshold value that user selected before.

2.2.2 Non-Functional Requirements

Usability: All users should be able to access room temperature by app with any smart phones. Also, all users should be able to get emergency mail with an e-mail.

Reliability: It is accurate all the time unless sensor is in a safe place and not interrupt by any item to get a reliable temperature.

Availability: The system should be available from any network and be available by 99Performance: The response time to the temperature of the environment should not exceed 1 minute.

3 Results and Analysis

First, we ensured that we have the required hardware/system. We configured the system and tested the functionality of our IoT device. We have experienced that it fulfills all functional requirements. Then we tested the device at varying temperature levels and intervals in places with an internet connection. Our sensor has successfully measured temperatures between 0 50°C.

Response Time: If the temperature is outside the threshold values, the notification will arrive within 10 seconds. As a result of our tests, we saw that an e-mail was sent within 5 seconds on average.

Error Rate: We aimed to keep the error rate as low as possible, below 15 percent if possible. As a result, we got an error rate of 1

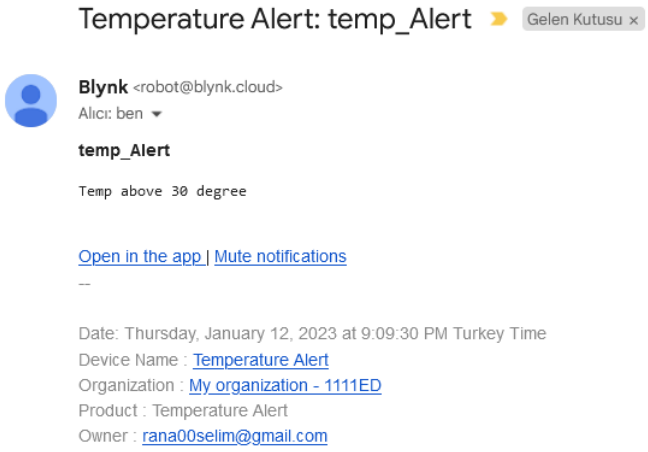

Figure 2: The proposed system overview

4 Conclusions and Future Work

The main goal of this IoT project was to design a system that could monitor the temperature of a specific room and send notifications to the user if the temperature falls outside a certain range. The system was designed using temperature sensors, a microcontroller, and IoT connectivity to send data to a mobile app for visualization and to an email server for notifications.

The system was tested in a controlled environment and the results showed that the temperature sensor was able to accurately detect temperature changes within ±2°C of the actual temperature. The temperature sensor used in the project was calibrated and cross-checked with a reference thermometer to ensure the accuracy of the readings.

The temperature data was collected at regular intervals and sent to the cloud server for storage and analysis.

The notifications were sent promptly to the user's email and mobile app when the temperature exceeded the pre-set threshold. The notifications were customizable and allowed the user to set the threshold temperature and the time interval for notifications. Users were also able to turn off or on the notifications as per their preference.

The mobile app provided a useful visualization of the temperature data over time, allowing the user to easily monitor and track the temperature. . The app was user-friendly and easy to navigate. Users could also change the code to set the threshold temperature and the time interval for notifications.

The system was designed to be scalable, which means that it could accommodate more temperature sensors in case of monitoring temperature in multiple rooms.

The main goal of this IoT project was to design a system that could monitor the temperature of a specific room and send notifications to the user if the temperature falls outside a certain range. The system was designed using temperature sensors, a microcontroller, and IoT connectivity to send data to a mobile app for visualization and to an email server for notifications.

The system was tested in a controlled environment and the results showed that the temperature sensor was able to accurately detect temperature changes within ±2°C of the actual temperature. The temperature sensor used in the project was calibrated and cross-checked with a reference thermometer to ensure the accuracy of the readings. The temperature data was collected at regular intervals and sent to the cloud server for storage and analysis.

The notifications were sent promptly to the user's email and mobile app when the temperature exceeded the pre-set threshold. The notifications were customizable and allowed the user to set the threshold temperature and the time interval for notifications. Users were also able to turn off or on the notifications as per their preference.

The mobile app provided a useful visualization of the temperature data over time, allowing the user to easily monitor and track the temperature. . The app was user-friendly and easy to navigate. Users could also change the code to set the threshold temperature and the time interval for notifications.

The system was designed to be scalable, which means that it could accommodate more temperature sensors in case of monitoring temperature in multiple rooms.

Overall, the system was successful in achieving its goal of providing real-time temperature monitoring and notifications. The system performed well and met the specifications outlined in the design phase. Based on these results, it can be concluded that the system is reliable and can be used for temperature monitoring in a home or small office setting. The system can be easily integrated with other IoT devices and the data collected can be used for further analysis.

In future, we could consider testing the system in a real-world environment to ensure its performance in different conditions. Additionally, we could add additional sensors to monitor humidity, motion, or other factors that could affect temperature. The system could also be enhanced with more advanced machine learning algorithms to analyze the data and provide more accurate predictions or recommendations.

Overall, the results of the project demonstrate that the system is a reliable and effective solution for temperature monitoring and notifications. The system is easy to use, and the user interface is intuitive. The system also provides a cost-effective solution for temperature monitoring and can be easily integrated with other IoT devices. It's also eco-friendly, as it conserves energy by being low-power.

5 References

- Performance Testing Metrics – A Detailed Guide for Businesses https://ieeexplore.ieee.org/document/9487691
- IOT based Temperature Monitoring System https://www.testingxperts.com/blog/performance-testing-metrics
- Non-Functional Requirements https://www.scaledagileframework.com/nonfunctional-requirements/
- Functional Requirements https://slideplayer.com/slide/6004175/
- Functional and Non-Functional Requirements

https://www.nuclino.com/articles/functional-requirements

- How to use Blynk IoT platform https://www.youtube.com/watch?v=X5zVaGk8QV0