FPT UNIVERSITY

FALL24 IoT102t Weather Station

1 Nguyen Huynh Nhu An,
2 Hoang Van Huy,
3 Nguyen Dinh Luc,
4 Ta Giang Nam,
5 Ho Dinh Anh

FPT University, Ho Chi Minh Campus, Viet Nam

{annhnse180768, huyhvse180722, lucndse180732,
namtgse182707, anhddse180670}@fpt.edu.vn

December 2024

1 Introduction

The **"Weather Station"** project is designed to monitor and alert users about indoor and outdoor environmental conditions. The system uses temperature and humidity sensors inside the rooms to track changes in the environment and send alerts when there are significant temperature fluctuations that may affect household items. Additionally, the system integrates with weather APIs to provide real-time information about outdoor weather, delivering alerts about possible weather conditions such as rain, heat, or strong winds that could impact the surroundings. To enhance user convenience, the system is also integrated with a Telegram bot, allowing users to receive notifications and interact with the system via a mobile app. This integration provides an easy and efficient way for users to monitor and manage environmental conditions remotely. Although the project has not yet automatically adjusted the temperature, it provides users with timely information so they can take action. The goal of this project is to help protect household items, save energy, and improve safety and comfort for users in their home environment.

2 Methods and Materials

2.1 Components and Peripheral Devices

The developed Weather Station system incorporates various electronic devices and components, e.g..., an ESP32 board as the brains of the system, sensors to measure temperature and humidity, a controller to manage equipment like alarms, displays to present weather information, etc., to perform its intended functions. The complete list of required components and tools is provided in Table.

Table 2.1: System's components and peripheral devices

Components/devices	**ID/remarks**
ESP32-2432S028 board	Integrated Wi-Fi, I2C", TFT2.8"
DS3231 real-time clock module	Accurate timekeeping
AHT20 temperature and humidity sensor	I2C interface
Speaker	Audio output device
3D-printed stand	Customizable housing

Explain System Components:
-Microcontroller (ESP32): This microcontroller is the control center of the system. It performs all the calculations and communication between the sensors, the user interface, the Telegram bot, and the weather API. The ESP32 is also responsible for the Wi-Fi connection to send weather data and receive requests from the user.
-ATH20 Sensor (Temperature and Humidity): The AHT20 sensor is used to measure the temperature and humidity in the environment. This data will be read by the ESP32 and displayed on the user interface and sent to the Telegram Bot when requested.
-Weather API (e.g. open-meteo): The system connects to a weather API to get weather data based on the user's location. Data from this API will provide information such as current temperature, humidity, weather conditions, and other meteorological information.
-RTC (Real-Time Clock): The RTC module provides real-time information to the system.
-User Interface (UI) with LVGL: The user interface uses the LVGL library to display weather data (temperature, humidity) and other settings such as alarms, sensor status, and notifications. This interface can interact with the user via the touch screen.
-Telegram Bot: The Telegram bot allows users to request weather information remotely. It receives commands from the user and responds with current weather data from the AHT20 sensor and from the weather API.
-Audio: Used for alarms

2.2 System Model and Block Diagram

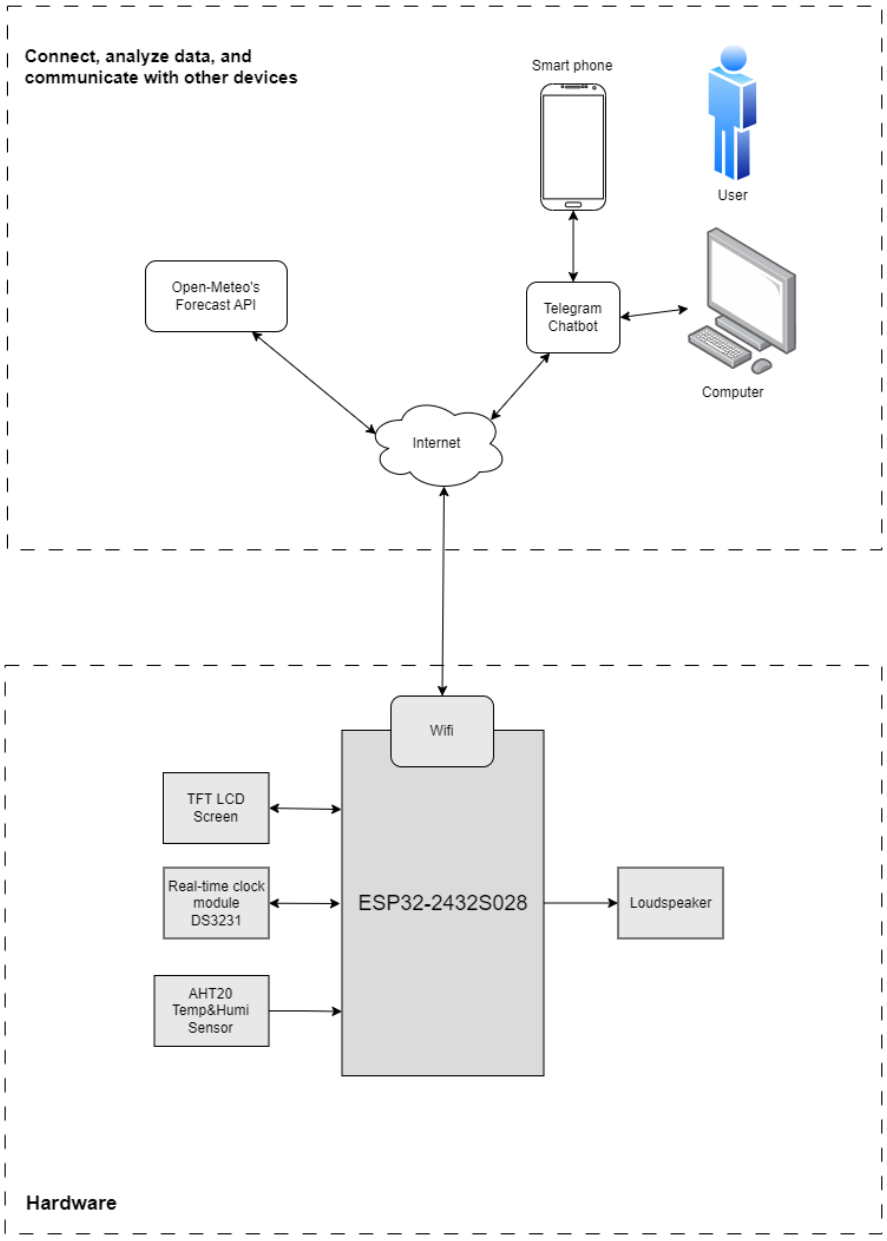

Figure 2.2: System Model and Block Diagram

The Weather Station system is a real-time environmental monitoring device that uses sensors to measure indicators such as temperature and humidity. The collected data will be displayed on the user interface (GUI) and can be sent via Telegram Bot to notify the user. Here are the components and how the system works:

Weather Data Measurement: The ESP32 microcontroller uses the AHT20 sensor to measure the temperature and humidity in the environment. In addition, the ESP32 controller will send an HTTP request (GET request) to the weather API to get weather data for a specific location (usually by GPS coordinates or city name). The data received from the API will include information such as temperature, humidity, weather conditions, and other information. The system will use this data to update the user interface (UI) and can also send information via Telegram Bot when requested by the user. This data will be displayed on the user interface (UI) and can be sent via Telegram bot.

Update data periodically: Sensor data is updated periodically using timers in the system. Every certain period of time (e.g. 2 seconds), the system will go to the weather API to get weather data, read data from the sensor, and update the interface.

User interaction: Users can request weather information via Telegram Bot. When the bot receives the command, it will send temperature and humidity data to the user. Users can also view the information directly on the built-in screen of ESP32.

Notification when there is a special event: If the temperature exceeds a certain level (e.g. 35°C), the system will send a notification via Telegram bot to warn the user.

Alarm setting: Users can set the alarm time via the user interface. When the alarm time comes, the system will trigger the alarm and can play music through the speaker.

3 Electronic Circuit/Hardware Interfacing

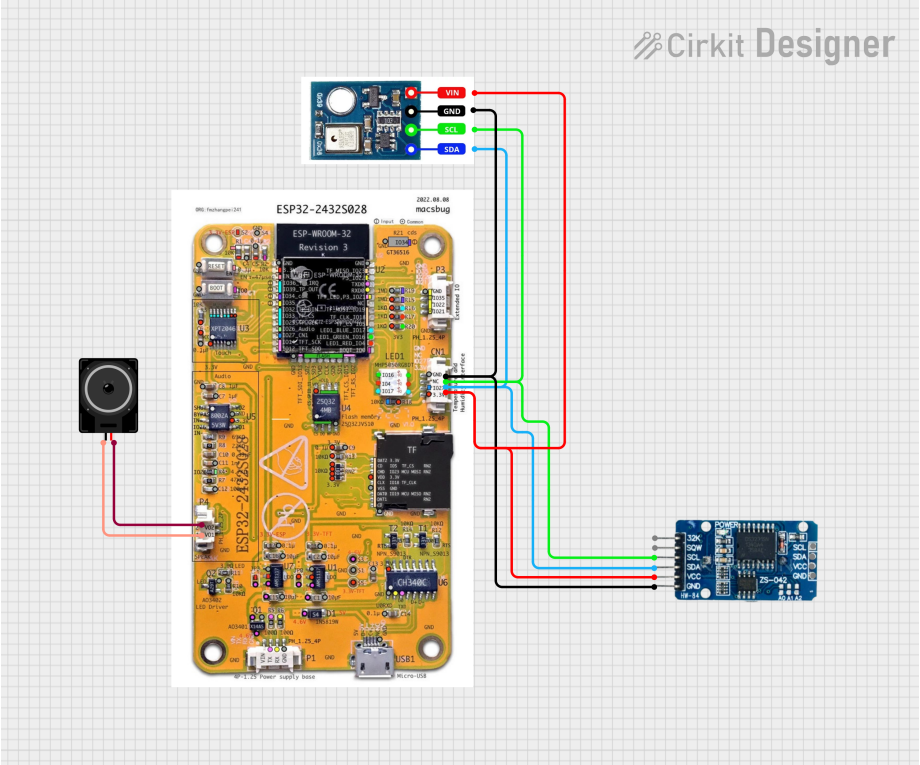

Figure 3.0: Circuit schematic/hardware interfacing

Table 3.1: Interfacing between ESP32-2432S028 and its components

ESP32-2432S028	DS3231	ATH20
VCC(3.3V)	VCC	VIN
GPIO27	SDA	SDA
GPIO22	SCL	SCL
GND	GND	GND

4 Programming Flowchart

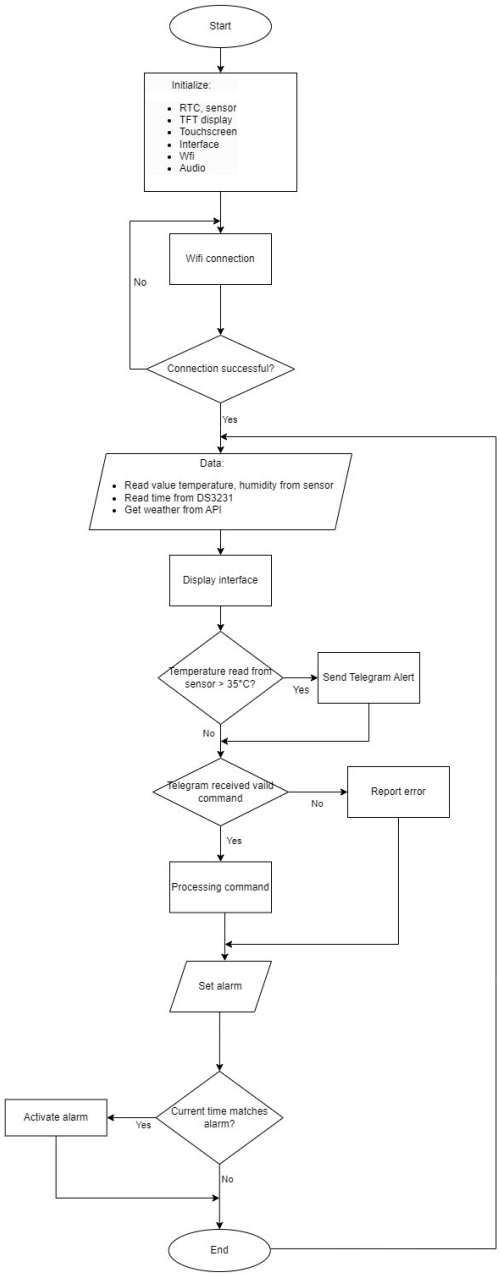

Figure 4.0: Programming flowchart of the developed system

Steps Explained:
1. Start The process begins with system initialization.
2. Initialize Components The system initializes key hardware and software components: RTC (Real-Time Clock), Sensors, TFT Display, Touchscreen, WiFi, Audio module
3. WiFi Connection The system attempts to establish a WiFi connection. If the connection fails, the process loops back until a successful connection is established.
4. Data Retrieval Once connected, the system collects data: Reads temperature and humidity values from the sensor. Reads time from the DS3231 RTC module. Retrieves weather information from an API.
5. Display Interface The gathered information is displayed on the interface (e.g., touchscreen or TFT display).
6. Temperature Monitoring If the temperature exceeds 35°C, the system sends an alert via Telegram.
7. Telegram Command Processing If a valid command is received via Telegram, it processes the command. If the command is invalid, it reports an error.
8. Set Alarm Based on the command received, the system sets an alarm.
9. Alarm Activation If the current time matches the set alarm time, the system activates the alarm. If not, it waits until the condition is met.
10. End The process ends or restarts for continuous monitoring.

5 Applications of the project

5.1 Home Climate Control

The main application of the smart weather station is monitoring and controlling indoor temperature. By integrating temperature sensors with cloud-based systems, users can receive real-time alerts about the temperature in their homes. This is useful for preventing extreme temperature changes that could lead to discomfort or property damage (e.g., frozen pipes in cold weather or excessive heat during the summer).

5.2 Emergency Alerts and Notifications

The system can send alerts to users when indoor temperatures exceed predefined thresholds, indicating potential issues like heating system failures or air conditioning malfunctions. These alerts can be received through mobile apps, enabling users to take action promptly.

5.3 Energy Efficiency

Smart weather stations can contribute to energy savings by helping users optimize their heating and cooling systems. By providing accurate data on temperature fluctuations and offering suggestions based on real-time data, users can improve energy efficiency and reduce utility costs.

5.4 Health and Safety

By monitoring temperature and humidity, the smart weather station ensures that the living environment remains comfortable and healthy, preventing issues like mold growth or other health-related concerns. Families with elderly members or children can benefit from continuous monitoring to maintain a safe living environment.

5.5 Smart Home Integration

The weather station can be integrated with other smart home devices, such as thermostats, air purifiers, and security systems. This integration allows for automated actions based on weather data, such as turning on the heating system when the temperature drops below a certain level or activating fans when the temperature rises too high.

6 Conclusion

This smart weather station project is not only a way to track and manage the indoor environment but also serves as a preventive tool for health, comfort, and safety. It integrates modern technology (sensors, APIs, and mobile apps) to offer a seamless experience for users in managing their household environment effectively.

References

1. Random Nerd Tutorials. *Programming the ESP32-CYD Cheap Yellow Display with VS Code*. Available online at https://randomnerdtutorials.com/programming-esp32-cyd-cheap-yellow-display-vs-code/.

2. "Cheap Yellow Display" GitHub Repository.Available online at https://github.com/witnessmenow/ESP32-Cheap-Yellow-Display/tree/main.

3. Universal Arduino Telegram Bot GitHub Repository. Available online at https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot/tree/master.

4. *Run multiple tasks on the ESP32 with FreeRTOS*. Available online at https://www.youtube.com/watch?v=WQGAs9MwXno.