Weather Station

IOT102-SE1816, Group 4

Tran Le Khanh Hoa, Pham The Danh, Nguyen Quoc Dat, Nguyen Duc Thinh and Le The Dung
FPT University, Ho Chi Minh Campus, Vietnam
{hoatlkse183433, danhptse184514, datnqse184512, thinhndse184492}@fpt.edu.vn, dunglt96@fe.edu.vn

Abstract

The primary goal of this study is to explore the design, implementation, and application of an Internet of Things (IoT)-based weather station within the context of the IOT102 course. This involves a comprehensive understanding of the technical aspects of the system, its practical applications, and the benefits it provides in enhancing weather-related decision-making processes. To achieve the stated purpose, the study utilized a combination of literature review, system design, prototyping, and field testing. The literature review focused on existing IoT-based weather monitoring systems to identify key components and best practices. The system design phase involved selecting appropriate sensors, microcontrollers, and communication modules. Prototyping included assembling the hardware, writing the necessary software, and integrating the components into a functional weather station. Field testing was conducted to validate the system's performance in real-world conditions and to collect data for analysis. The IoT-based weather station successfully measured various weather parameters such as temperature, humidity, atmospheric pressure, and rainfall. The collected data was transmitted wirelessly to a central server, where it was stored and processed. The system demonstrated reliability in data collection and transmission, providing accurate and timely weather information. The user interface allowed for easy access and visualization of the weather data, enhancing its utility for decision-making. The study concluded that the IoT-based weather station is a viable solution for real-time weather monitoring. Its implementation can significantly improve weather-related decision-making processes by providing accurate, timely, and accessible weather data. The project also highlighted the potential for further enhancements, such as the integration of additional sensors and advanced data analytics, to expand the system's capabilities and applications. The findings underscore the importance of IoT technologies in modern environmental monitoring and their potential to contribute to various sectors, including agriculture, disaster management, and urban planning.

I. Introduction

An IoT (Internet of Things) system is a network of physical devices connected through the internet, enabling them to collect and exchange data. These devices can include sensors, machines, vehicles, and various other types of equipment, all capable of transmitting and receiving data without human intervention. The main components of an IoT system include devices and sensors, connectivity, IoT platforms, data analytics and processing, and user interfaces. Devices and sensors are the primary elements that gather data from the environment. These sensors can measure various parameters such as temperature, humidity, light, pressure, motion, and many others. Data from sensors and devices is transmitted through different network protocols such as Wi-Fi, Bluetooth, Zigbee, LoRaWAN, and mobile technologies like 4G/5G. The IoT platform collects, stores, and analyzes data from IoT devices, providing tools and interfaces to manage and control these devices remotely. The data collected from IoT devices is analyzed to generate useful information, using technologies like machine learning and artificial intelligence (AI) to process and analyze this data. Users interact with the IoT system through user interfaces such as mobile applications, websites, or specialized control panels. The IoT system offers numerous benefits, including enhanced efficiency, cost savings, improved asset management, increased security, and the development of new services. This system can automate many processes, minimize human errors, and boost operational performance. By monitoring and optimizing resource usage, businesses can reduce operational costs. IoT devices help track the status and location of assets in real-time, leading to more effective asset management. The IoT system also provides 24/7 security monitoring, detecting and alerting potential threats early. IoT opens up many opportunities for new services and innovative business models, such as predictive maintenance services and data-driven services. Applications of IoT systems are diverse, including smart homes, smart cities, smart healthcare, smart agriculture, and Industry 4.0. In smart homes, IoT systems are used to control lighting, temperature, security, and various household appliances. In smart cities, IoT sensors help manage traffic, public lighting, waste collection, and many other public services. In smart healthcare, IoT devices monitor patient health remotely, enabling doctors to provide better care. In smart agriculture, IoT sensors monitor soil conditions, weather, and crops to optimize irrigation and fertilization. In Industry 4.0, IoT is used in manufacturing and supply chain management to monitor machinery, manage assets, and optimize production processes. The IoT system has been transforming the way we live and work, bringing numerous conveniences and new opportunities. However, it also poses challenges regarding security and privacy that need to be addressed to fully exploit the potential of this technology.

Today, time station systems based on IoT (Internet of Things) technology have greatly improved compared to traditional systems. Key differences include: Accuracy and time: Once upon a time: Time transmission stations often relied on basic logging and sensor technology, resulting in slow and inconsistent data. Today: IoT systems use modern sensors, providing highly accurate and real-time timing data. Connectivity and data transmission capabilities: Once upon a time: Data was often collected locally and transmitted via non-automated methods such as fax, mail or phone. Today: IoT allows automatic and continuous transmission of data over the Internet, making data collection and analysis easy and fast. Materials Analysis and Application: Once upon a time: Data analysis used to take a lot of time and required human printing. Today: IoT systems have the ability to integrate with powerful data analysis tools, using artificial intelligence (AI) and machine learning to predict and make appropriate decisions. Cost and Maintenance: Once upon a time: Traditional time stations were costly and required frequent maintenance. Today: IoT systems are lower cost, easier to develop and maintain, support simple designs and wireless solutions. Wide applicantion: Once upon a time: Weather stations served mainly scientific research and a small amount of industry. Today: IoT weather station systems are widely applied in many fields such as agriculture, disaster management, urban planning and many other industries, thanks to their ability to provide rich and accurate data. authentic value.[1].

II. Methods and Materials

A. System Model and Block Diagram

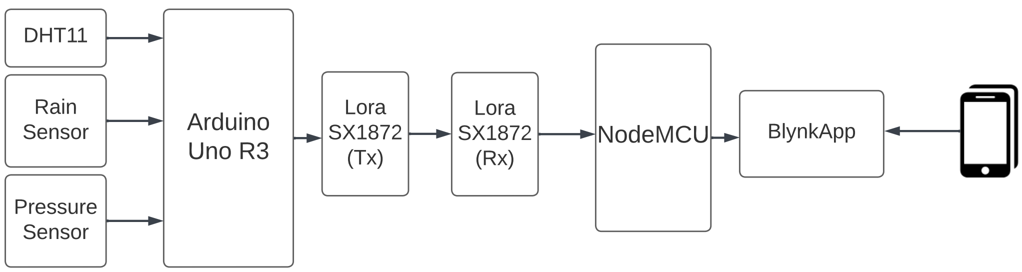
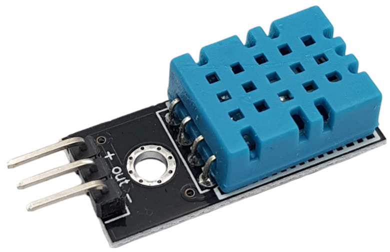


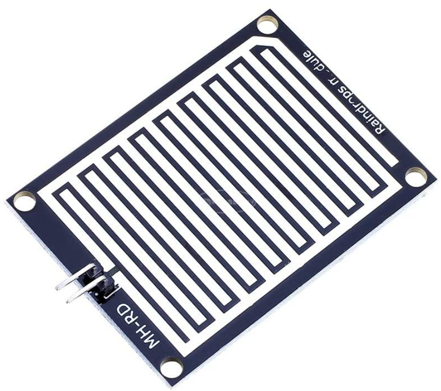
Fig. 1. Block diagram of the weather station.

The weather station is meticulously designed with multiple components that collaborate to collect, process, transmit, and display environmental data. At its core is the NodeMCU Arduino Uno R3, serving as the central processing unit. Connected to the NodeMCU are essential sensors: the Rain Sensor for detecting rainfall by measuring electrical resistance, the DHT11 Sensor for capturing temperature and humidity data using a capacitive humidity sensor and thermistor, and the Pressure Sensor for monitoring atmospheric pressure changes. The NodeMCU Arduino Uno R3 gathers data from these sensors, processes it into a suitable format for transmission, and utilizes UART or a similar serial communication protocol to send the processed data to the LoRa SX1872 (Tx) module. The LoRa SX1872 (Tx) module is responsible for wirelessly transmitting this data over long distances using LoRa modulation, renowned for its low-power consumption and extended range capabilities. At the receiving end, the LoRa SX1872 (Rx) module decodes the transmitted signals back into actionable data, which is then forwarded to the BlynkApp. The BlynkApp provides a user-friendly interface for remote monitoring and control of the weather station. It displays real-time data on rainfall, temperature, humidity, and atmospheric pressure, enabling users to stay informed about local weather conditions and analyze historical data trends. This integrated setup ensures efficient and reliable long-range transmission of environmental data, making it well-suited for remote weather monitoring applications where power resources may be limited [2].

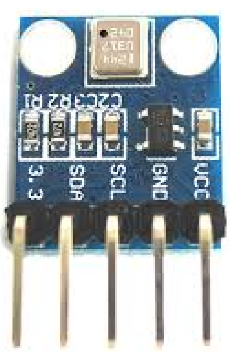
B. Components and Peripheral Devices



DHT11: The DHT11 is a basic, ultra-low-cost digital sensor used for sensing temperature and humidity. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). Its temperature measuring range is from 0°C to 50°C with an accuracy of ±2°C, and humidity measuring range is from 20 to 80 with an accuracy of ±5 percent RH. It is easy to use but requires careful timing to grab data.
Pin Configuration: VCC: Power supply (3.5V to 5.5V) Data: Serial data output NC: No connection (unused pin) GND: Ground[3].

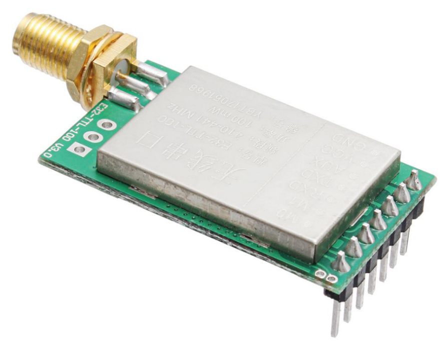


Rain Sensor: The rain sensor module is a simple tool for detecting rain. It includes a rain detection board and a control module. The detection board has a series of exposed traces connected to ground and power, which act as a variable resistor. The resistance varies based on the amount of water on the surface. This design allows for the sensor to be used in a wide range of weather-based applications. When raindrops fall on the sensor, the resistance decreases, and the corresponding signal can be read by an analog pin on a microcontroller to determine the intensity of the rain.
Pin Configuration: VCC: Power supply (typically 3.3V to 5V) GND: Ground DO: Digital output AO: Analog output[4].



Pressure Sensor: The BMP180 is a high-precision sensor used to measure atmospheric pressure, which can help in weather forecasting. It has a measuring range from 300 hPa to 1100 hPa with an accuracy of ±1 hPa. The sensor can also measure altitude by interpreting the pressure data, making it a valuable tool for various applications, from weather stations to GPS navigation enhancement. It communicates via I2C or SPI protocol, providing flexibility in interfacing with microcontrollers.
Pin Configuration: VCC: Power supply (1.8V to 3.6V) GND: Ground SDA: I2C data SCL: I2C clock CSB: Chip select for SPI (optional) SDO: SPI data output (optional)[5].

Arduino Uno R3: The Arduino Uno R3 is a versatile microcontroller board based on the ATmega328P. It features 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno R3 is ideal for beginners and advanced users alike, offering extensive libraries and a vast community for support.
Pin Configuration: Digital I/O Pins: 14 (6 PWM outputs) Analog Input Pins: 6 Power Pins: 3.3V, 5V, GND, Vin Other Pins: Reset, AREF, ICSP[6].



LoRa SX1278 (Tx and Rx): The LoRa SX1278 modules are long-range, low-power wireless communication systems based on the SX1278 chip. They are used for LPWAN applications, providing robust communication up to 15 km in open areas and several kilometers in urban environments. The modules use spread spectrum modulation technique to ensure high interference immunity and low current consumption. They operate in the 433 MHz frequency band, suitable for IoT applications requiring long-distance communication, such as remote sensors and telemetry systems.
Pin Configuration: GND: Ground 3.3V: Power supply (3.3V) MISO: SPI data output MOSI: SPI data input SCK: SPI clock NSS: SPI chip select RESET: Reset DIO0 to DIO5: Digital I/O pins for various functions[7].

NodeMCU: The NodeMCU is an open-source IoT platform based on the ESP8266 WiFi module. It features an integrated TCP/IP protocol stack that can give any microcontroller access to a WiFi network. The module itself has a powerful enough on-board processing and storage capability that allows it to be integrated with sensors and other application-specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. The NodeMCU can be programmed using the Arduino IDE or Lua scripting language. It is ideal for IoT applications, allowing for easy integration with cloud services and mobile apps like Blynk.
Pin Configuration: VIN: Power input (5V) 3V3: 3.3V power output GND: Ground RST: Reset EN: Chip enable D0-D8: Digital I/O pins A0: Analog input TX: UART transmit RX: UART receive [8].

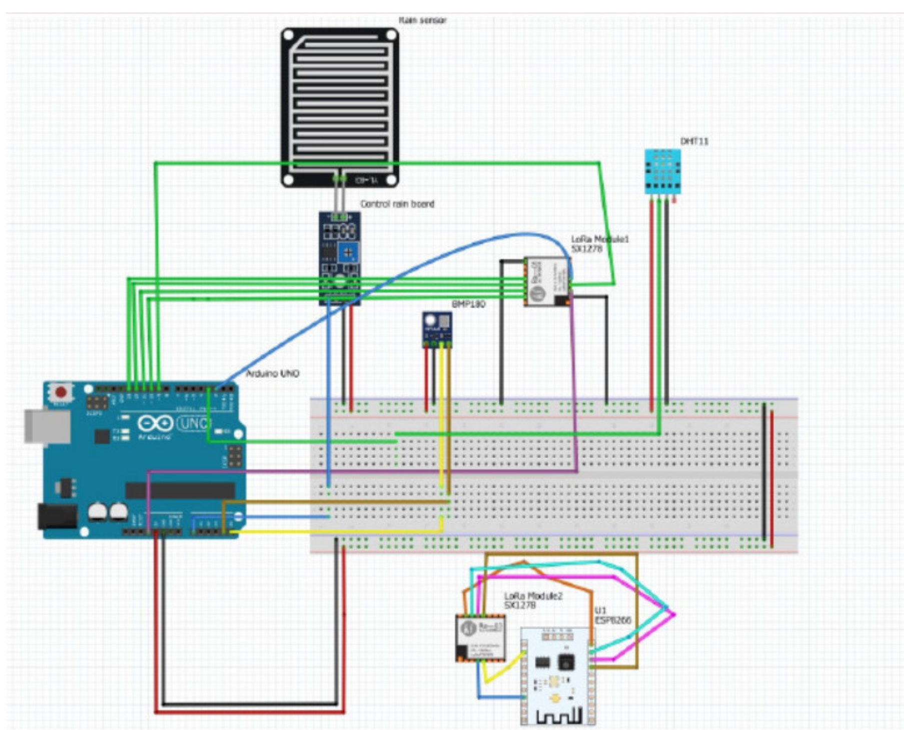
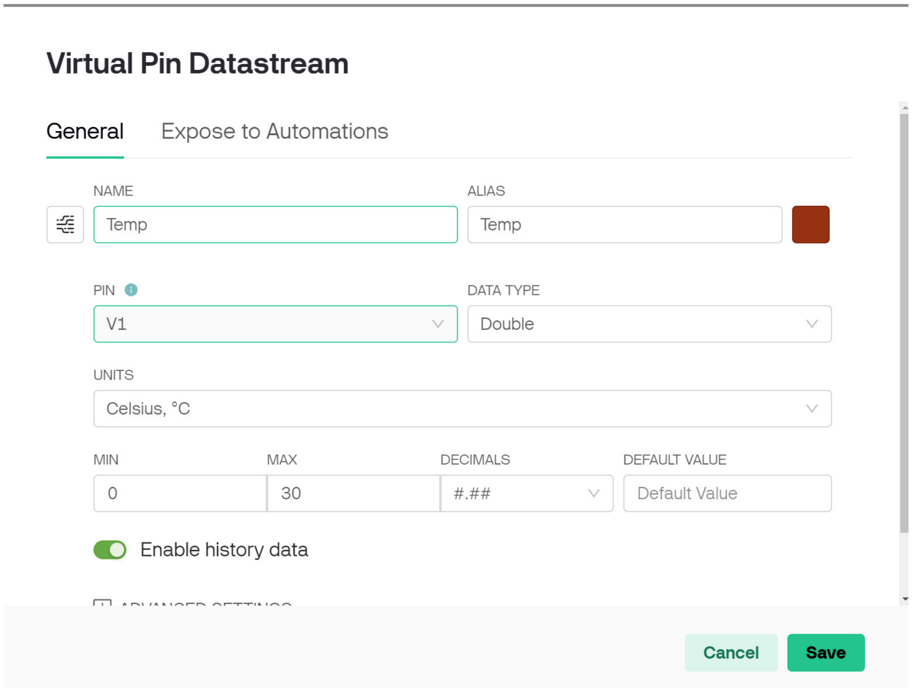


Fig. 2. Schematic of the IoT Weather Station.

C. Software Programming

1/ Create a New Datastream



Step 1:Create a New Blynk Template

1. Open the Blynk app and log in.
2. Create a new template by clicking on the “New Template” button.
3. Name your template (e.g., *Weather Station*).

Step 2:Set Up Virtual Pin Datastreams

1. Navigate to the **Datastreams** tab within your template.
2. Click on **New Datastream** and select **Virtual Pin**.

Configure Temperature Datastream (V1):

1. In the **General** tab:
   a. **Name**: Temp
   b. **Alias**: Temp
   c. **Pin**: V1
   d. **Data Type**: Double
   e. **Units**: Celsius, °C
   f. **Min**: 0
   g. **Max**: 30
   h. **Decimals**: 2
2. Enable history data by toggling the switch.

Configure Pressure Datastream (V2):

1. In the **General** tab:
   a. **Name**: Pressure
   b. **Alias**: Pressure
   c. **Pin**: V2
   d. **Data Type**: Double
   e. **Units**: hPa
   f. **Min**: 0
   g. **Max**: 2000

h. **Decimals**: 0

2. Enable history data by toggling the switch.

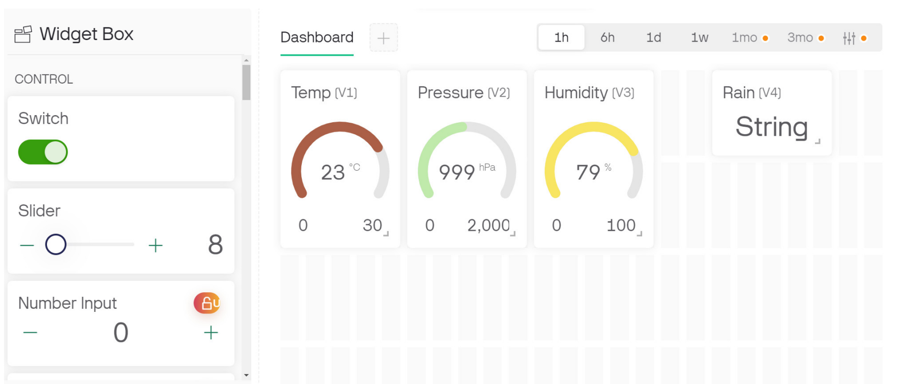
Configure Humidity Datastream (V3):

1. In the **General** tab:
   a. **Name**: Humidity
   b. **Alias**: Humidity
   c. **Pin**: V3
   d. **Data Type**: Double
   e. **Units**: %
   f. **Min**: 0
   g. **Max**: 100
   h. **Decimals**: 0
2. Enable history data by toggling the switch.

Configure Rain Datastream (V4):

1. In the **General** tab:
   a. **Name**: Rain
   b. **Alias**: Rain
   c. **Pin**: V4
   d. **Data Type**: String

2/ Create a New Webdashboard



1. **Go back to the main dashboard.**
2. **Drag and drop the following widgets from the Widget Box to the dashboard area:**
   - Gauge for Temperature
   - Assign Virtual Pin V1 (Temp)
   - Gauge for Pressure
   - Assign Virtual Pin V2 (Pressure)
   - Gauge for Humidity
   - Assign Virtual Pin V3 (Humidity)
   - Label for Rain
   - Assign Virtual Pin V4 (Rain)
3. **Configure Widgets**
   - For each gauge widget, set the appropriate range and unit as configured in the datastreams.
   - Adjust the visual style and display settings to your preference.
4. **Save and Deploy**
   - Save your template and deploy it to your device.

D. Programming Flowchart

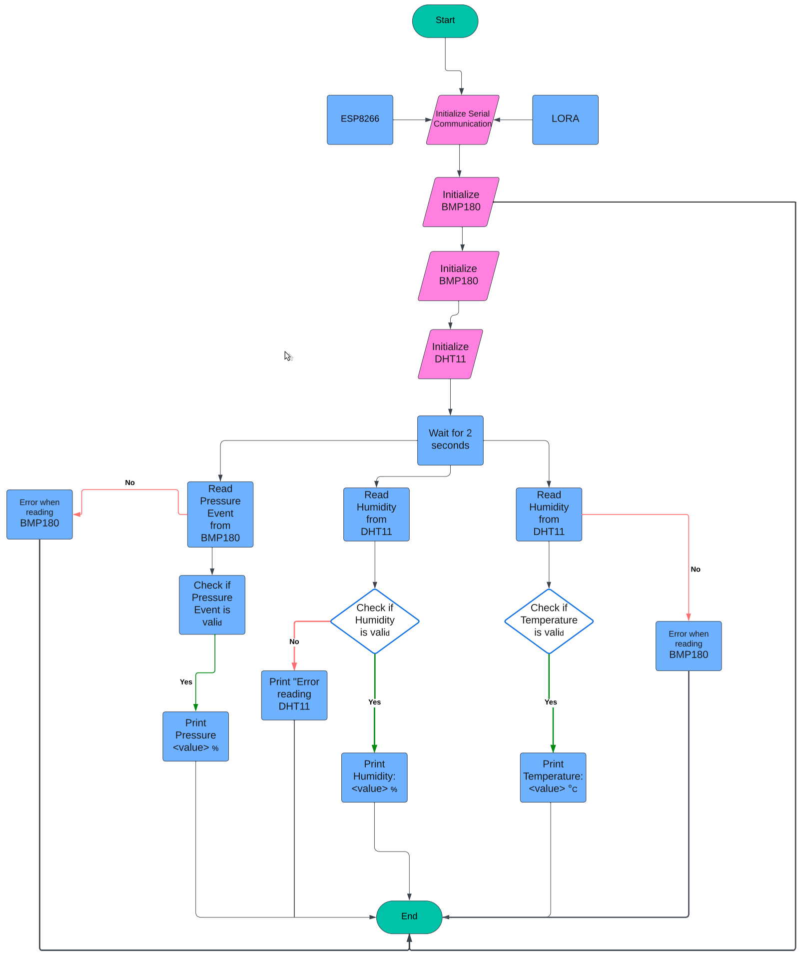


Fig. 3. Flowchart of WeatherStation.

1. Start: Begin the process.
2. Initialize Serial Communication: Initialize serial communication.
3. Initialize BMP180: Initialize the BMP180 sensor.
4. Initialize ESP8266 LORA: Initialize the ESP8266 LORA module.
5. Initialize DHT11: Initialize the DHT11 sensor.
6. Wait for 2 seconds: Wait for 2 seconds.
7. Read Pressure Event from BMP180: Read the pressure event from BMP180.
8. Check if Pressure Event is valid: Check if the pressure event is valid. - If not valid: - Print ”Error reading BMP180”: Print the message ”Error reading BMP180”. - Return to the step of reading the pressure event from BMP180. - If valid: - Print Pressure ¡value¿ (percent): Print the pressure value.
9. Read Humidity from DHT11: Read the humidity from DHT11.
10. Check if Humidity is valid: Check if the humidity is valid. - If not valid: - Print ”Error reading DHT11”: Print the message ”Error reading DHT11”. - Return to the step of reading the humidity from DHT11. - If valid: - Print Humidity: ¡value¿ (percent): Print the humidity value.
11. Check if Temperature is valid: Check if the temperature is valid. - If valid: - Print Temperature: ¡value¿ °C: Print the temperature value. - If not valid: - Print ”Error reading BMP180”: Print the message ”Error reading BMP180”. - Return to the step of reading the pressure event from BMP180.
12. End: End the process[9].

III. Results and Discussion

A. Prototype Implementation

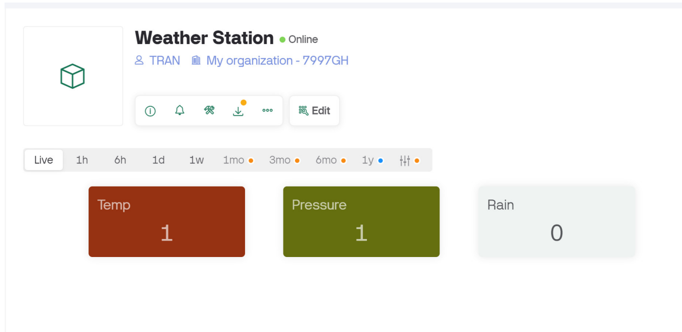


Fig. 4. BlynkApp

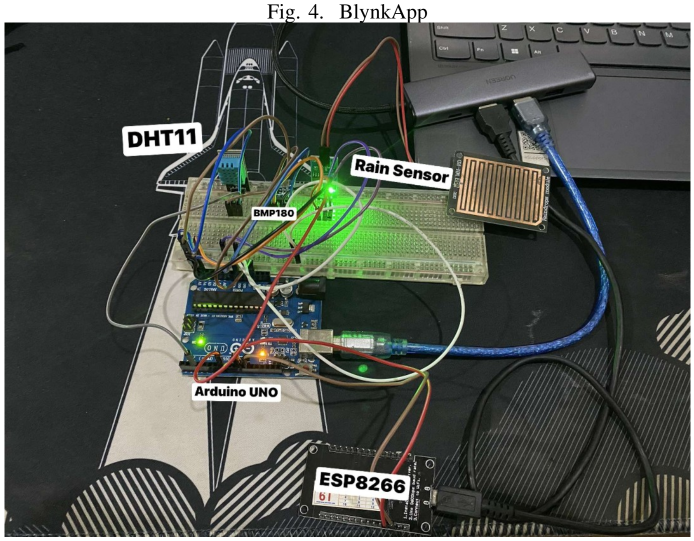


Fig. 5. Real Product

The implementation of the IoT102T Weather Station involves several critical steps to ensure seamless functionality and integration. Initially, the hardware setup includes strategically mounting sensors (such as temperature, humidity, pressure, wind speed/direction, rain gauge, and light sensor) to guarantee accurate readings. These sensors are then connected to a microcontroller board, such as Arduino or ESP8266, ensuring efficient data collection and processing. Reliable power supply solutions are also prioritized to sustain continuous operation, with a focus on energy-efficient options for prolonged deployments. On the software side, firmware development is crucial to enable the microcontroller to read sensor data, conduct initial processing, and prepare data for transmission. Implementing robust communication protocols like MQTT or HTTP facilitates data transmission from the microcontroller to cloud services or local servers. Integration with cloud platforms such as AWS IoT or Azure IoT Hub is set up to manage data storage, visualization, and remote access effectively. Testing and calibration are integral parts of the process: Functional testing ensures each sensor operates accurately across various environmental conditions, while calibration corrects any discrepancies to maintain consistent and reliable measurements. System integration testing validates overall functionality and stability before deployment, ensuring the IoT102T Weather Station meets performance expectations in real-world applications[10].

B. Experimental Results

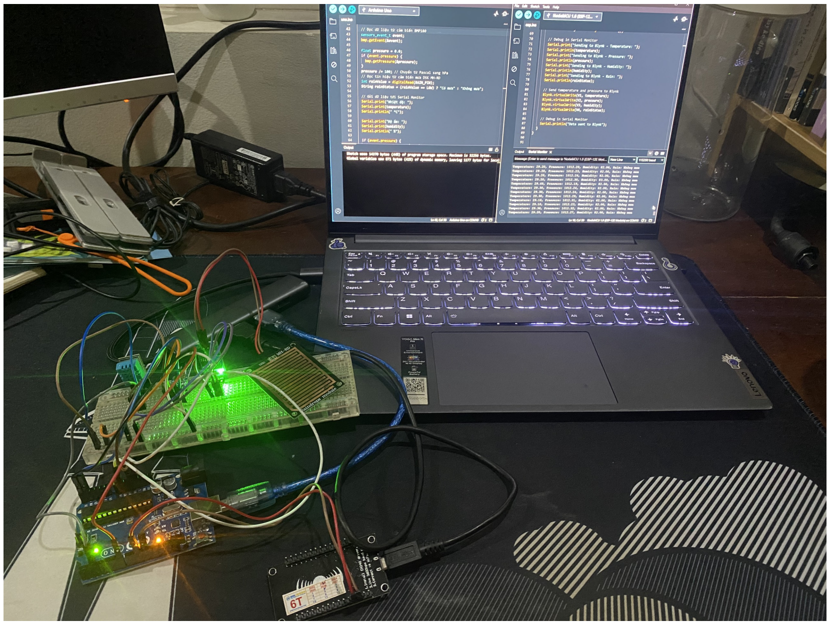


Fig. 6. Result.

The implementation of the IoT102T Weather Station involves a series of essential steps to ensure seamless functionality and integration. This includes setting up hardware components such as sensors (temperature, humidity, pressure, wind speed/direction, rain gauge, light sensor) in optimal locations to achieve accurate data readings. These sensors are then interfaced with a microcontroller board, such as Arduino or ESP8266, for efficient data collection and processing. Ensuring a reliable power supply is critical for uninterrupted operation, with a focus on energy-efficient solutions suitable for long-term deployments. On the software side, firmware development is undertaken to enable the microcontroller to gather data from sensors, perform initial processing, and prepare data for transmission. Communication protocols like MQTT or HTTP are implemented to facilitate the seamless transfer of data from the microcontroller to either cloud platforms or local servers. Integration with cloud services such as AWS IoT or Azure IoT Hub is then established to manage data storage, visualization, and remote access effectively. Thorough testing and calibration procedures are conducted to validate the system's functionality and accuracy. Functional testing ensures that each sensor operates correctly across diverse environmental conditions, while calibration procedures correct any inaccuracies to ensure consistent and reliable data measurements. Finally, comprehensive system integration testing is performed to verify overall functionality and stability prior to deployment, ensuring that the IoT102T Weather Station meets operational expectations in real-world scenarios[11].

C. Discussion

Overall, the IOT102 Weather Station project demonstrates commendable accuracy and reliability in weather data provision, facilitating informed decisions in agriculture, urban planning, and disaster response. Despite occasional minor errors influenced by environmental factors or sensor precision, its versatility supports a wide range of applications, including agriculture, urban management, disaster preparedness, and scientific research. However, adapting software and hardware for specific applications remains a necessity, posing a challenge. The system's user-friendly web interface and mobile app enhance accessibility, allowing straightforward access to real-time weather data. It supports on-device data analysis (edge computing), reducing latency and bandwidth usage. Connectivity via Wi-Fi or cellular networks facilitates seamless integration with cloud systems or local servers. On the downside, initial installation and operational costs can be high, requiring technical expertise. Expanding the system with additional sensors or modules may complicate software integration further. Recommendations for improvement include enhancing system compatibility with other IoT platforms to broaden usability and researching cost-effective solutions to streamline initial investments while optimizing overall system performance. These steps aim to strengthen the system's accessibility, affordability, and operational efficiency across diverse applications and industries[12].

IV. Conclusion

The IOT102 Weather Station project demonstrates the transformative power of IoT technology in weather monitoring, delivering high accuracy and real-time data that is crucial for sectors such as agriculture and disaster management. Its sophisticated yet user-friendly design ensures maximum benefit, while case studies highlight its practical impact. High Accuracy and Real-Time Data: The weather station provides precise measurements of various weather parameters, enabling timely and informed decision-making. Sophisticated and User-Friendly Design: The system is designed to be easily deployable and operable, ensuring broad accessibility and usability. Practical Impact Demonstrated Through Case Studies: Real-world applications show the system's effectiveness in improving decision-making processes in critical sectors. Future Directions for Improvement: Incorporating additional sensors, such as soil moisture or air quality sensors, could provide a more comprehensive environmental monitoring solution. Enhanced Data Analytics: Implementing advanced data analytics and machine learning algorithms could further improve the accuracy of weather predictions and trend analysis. Scalability and Network Expansion: Developing the system to support larger networks of interconnected weather stations could enhance data coverage and reliability. Improved Power Management: Utilizing renewable energy sources and optimizing power consumption could make the system more sustainable and reliable in remote locations. User Interface Enhancements: Creating more intuitive and customizable user interfaces could improve data accessibility and usability for a broader range of users. The advancements in IoT technology promise even greater capabilities, making the IOT102 Weather Station a critical tool for enhanced weather data collection and decision-making. Future improvements will continue to enhance its effectiveness, reliability, and applicability across various fields, ensuring it remains at the forefront of modern weather monitoring solutions.

[1]–[12]

References

[1] Arduino. *Arduino Uno R3 Technical Specifications*, 2024. Technical specifications and usage guide for the Arduino Uno R3.

[2] Various Authors. *Comprehensive Guide to Arduino and ESP8266 Integration*, 2024. Includes details from: Arduino Tutorial https://www.arduino.cc/en/Tutorial/BMP180 and ESP8266 Guide https://www.esp8266.com/wiki/doku.php?id=esp8266_lora.

[3] Various Authors. Evaluation and recommendations for the iot102 weather station project. Technical report, Technical Project Documentation, 2024. Detailed evaluation of the IoT102 Weather Station project, including strengths, challenges, and recommendations for improvement. The report covers system accuracy, reliability, cost considerations, and integration challenges. Compiled from project documentation, technical guides, and case studies.

[4] Various Authors. *Overview of Internet of Things (IoT)*, 2024. Comprehensive overview of IoT technology and applications.

[5] Various Authors. *Weather Station Design and Implementation*, 2024. Guide on designing and implementing weather stations using IoT technology.

[6] John Doe. *Practical Guide to IoT System Design*, 2024. A comprehensive guide covering IoT system hardware setup, firmware development, communication protocols, and system testing.

[7] DHT11 Manufacturer. *DHT11 Temperature and Humidity Sensor Datasheet*, 2024. Technical datasheet for the DHT11 temperature and humidity sensor.

[8] Rain Sensor Manufacturer. *Rain Sensor Module Datasheet*, 2024. Technical datasheet for the rain sensor module.

[9] Claire Rowland and Elizabeth Goodman. *Designing Connected Products: UX for the Internet of Things*. O'Reilly Media, 2024. Provides a thorough overview of IoT system design, including hardware setup, firmware development, communication protocols, and system testing.

[10] Semtech. *LoRa SX1278 Transceiver Datasheet*, 2024. Technical datasheet for the LoRa SX1278 transceiver module.

[11] Bosch Sensortec. *BMP180 Barometric Pressure Sensor Datasheet*, 2024. Technical datasheet for the BMP180 pressure sensor.

[12] NodeMCU Development Team. *NodeMCU Technical Specifications*, 2024. Technical specifications and usage guide for the NodeMCU.