Industrial Internship Report on

Weather Station using NodeMCU

Prepared by

Arnob Chakraborty

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time. My project was (Weather Station using NodeMCU) This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

1 Preface

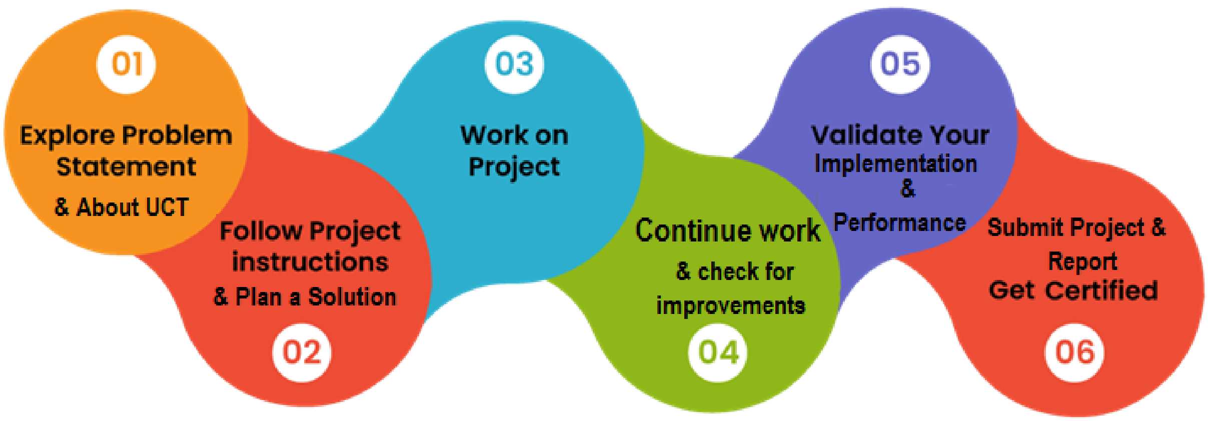
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

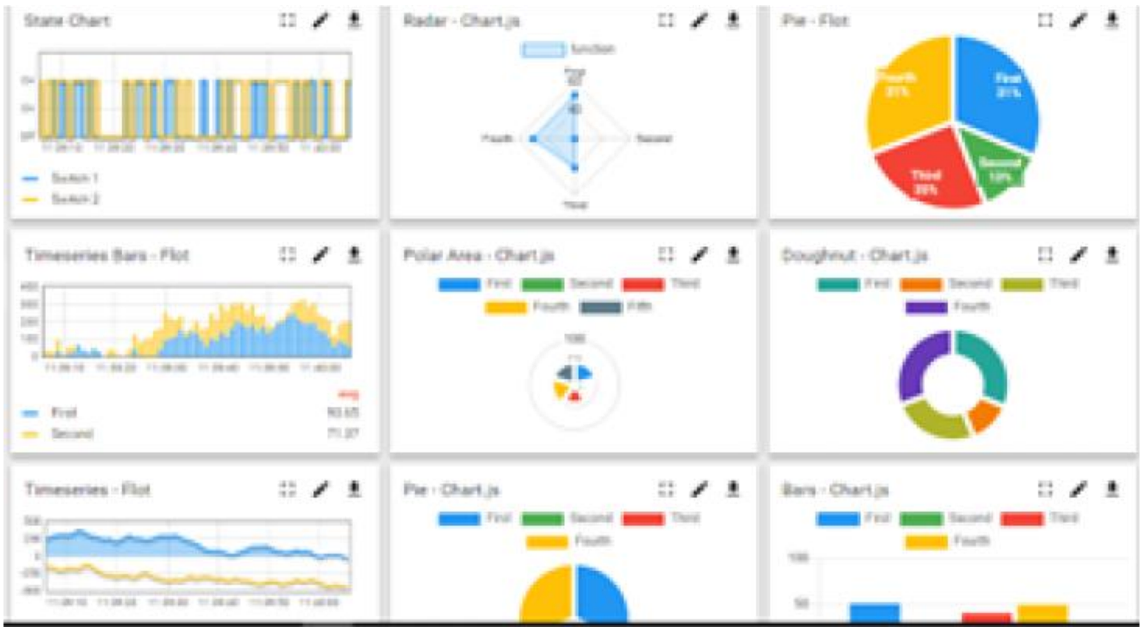
i. UCT IoT Platform (uct Insight)

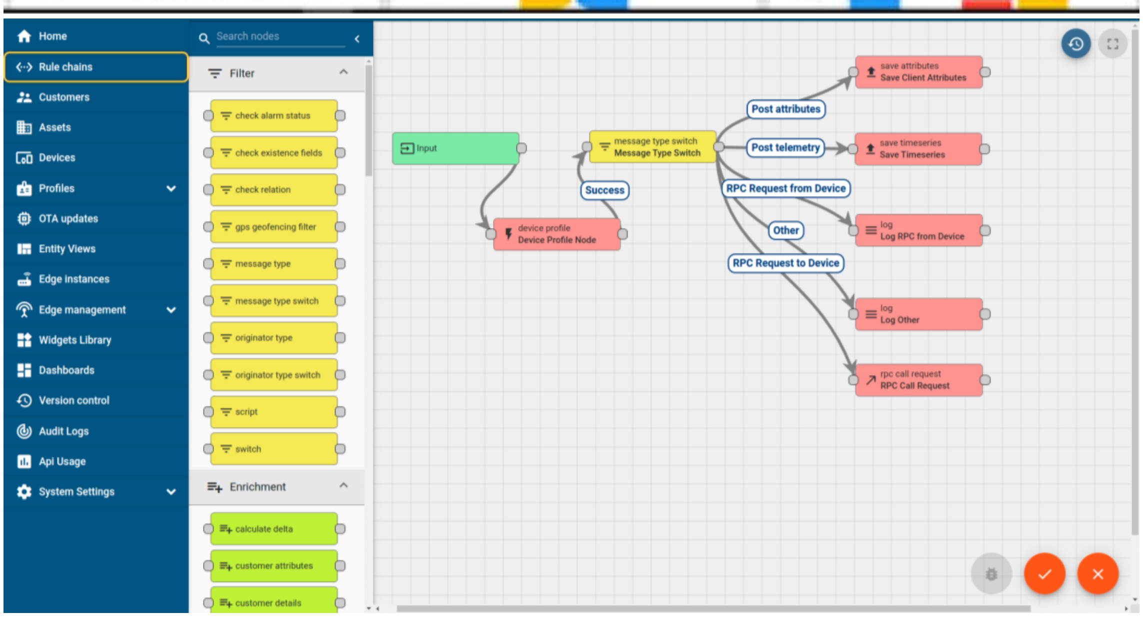
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

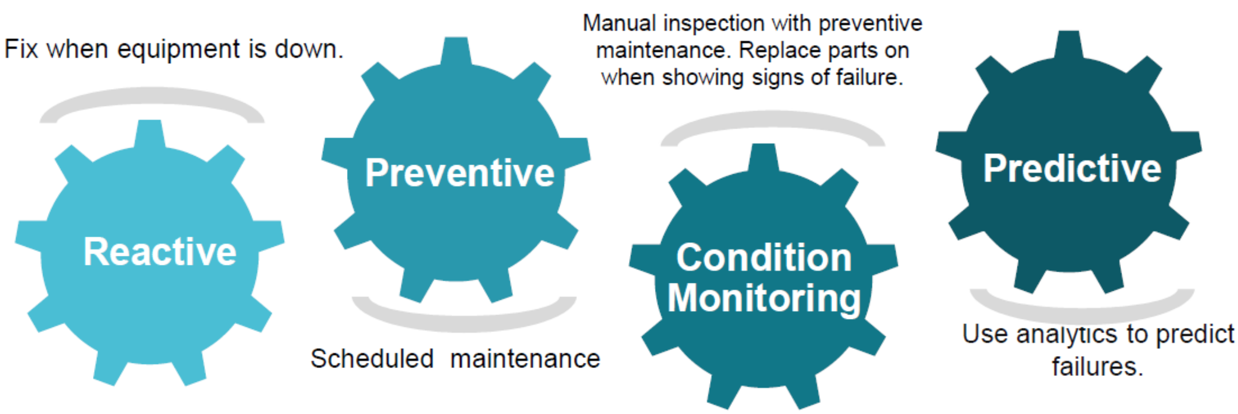
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN technology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

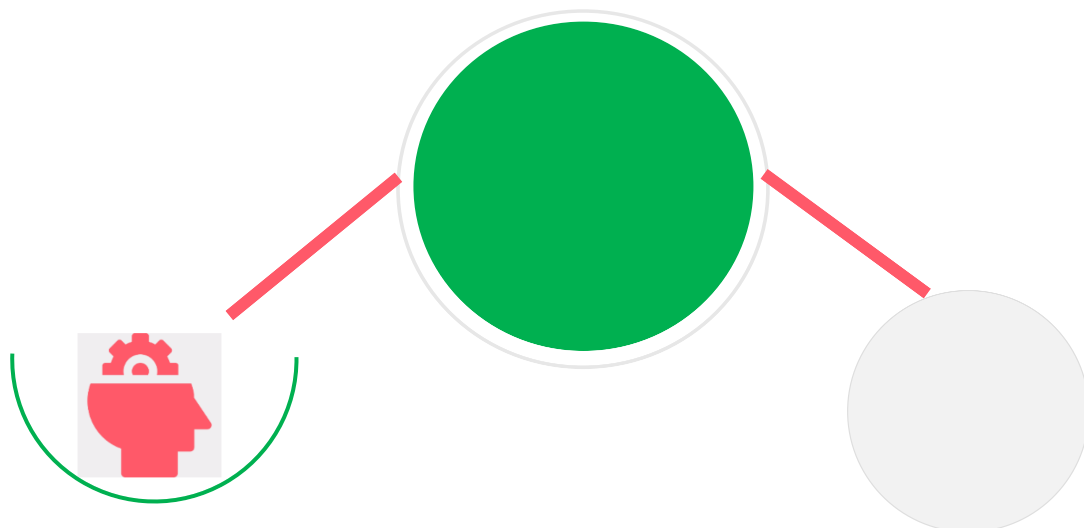
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

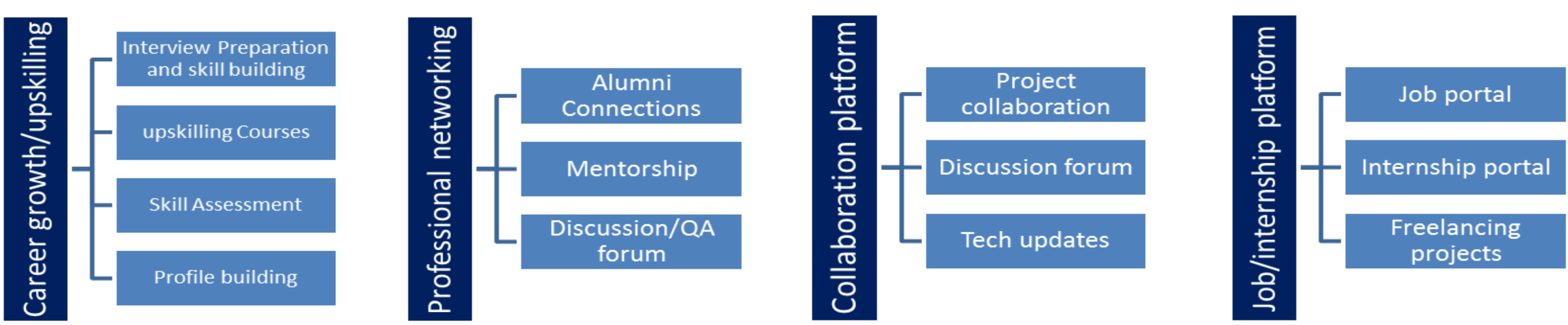
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] https://learn.upskillcampus.com/s/courses/6757d1148cbb6817f77c1d77/take

[2] https://www.youtube.com/watch?v=ytQpnt4b9CE

[3] https://www.youtube.com/watch?v=R_6twDv8w9A

2.6 Glossary

Terms	Acronym
Internet of Things	IoT
Test Plan	N/A
Test Cases	N/A
Test Procedure	N/A
Remote Access	N/A

3 Problem Statement

In today's rapidly changing climate, a compact, cost-effective weather station using the NodeMCU ESP8266 can address the need for accurate, real-time weather data by measuring temperature, humidity, and air quality with sensors like DHT11, BMP280 and MQ-135 leveraging IoT hardware, it will transmit data wirelessly to cloud platforms (e.g., ThingSpeak) for visualization, provide user-friendly access via web or mobile dashboards, ensure reliability and accuracy in diverse conditions, optimize power consumption, and enable scalability and modularity for easy upgrades, filling the gap left by bulky, costly traditional systems.

4 Existing and Proposed solution

Traditional weather monitoring systems include large-scale government-operated stations, commercial weather kits, and standalone digital devices. These solutions, while effective, are often expensive, bulky, or rely on centralized infrastructure for data collection and distribution. Portable weather stations offered by various manufacturers provide basic environmental readings but are limited in terms of customizability and scalability. Furthermore, many existing systems lack real-time cloud integration, making remote monitoring and data analysis challenging.

Limitations of Existing Solutions

1. **Cost**: High acquisition and maintenance costs for advanced systems.
2. **Accessibility**: Limited availability for individuals or small-scale users.
3. **Customizability**: Restricted ability to add or replace sensors for specific needs.
4. **Cloud Integration**: Minimal or no integration with modern IoT platforms for real-time monitoring and visualization.
5. **Scalability**: Existing systems are not designed for modular upgrades.
6. **Portability**: Bulky setups reduce ease of deployment and mobility.

4.1 Code submission (Github link):

https://github.com/arnobchak/Industrial-Internship-Report-upSkill-Campus-/blob/main/weatherstation_new_updated.ino

4.2 Report submission (Github link):

5 Proposed Design/ Model

The proposed weather station will use the NodeMCU ESP8266, a cost-effective and compact IoT module, to create a modular and scalable weather monitoring solution. By integrating sensors such as DHT11 (temperature, humidity), BMP280 (pressure sensor), and MQ-135 (air quality), the system will collect environmental data. The station will leverage Wi-Fi for real-time data transmission to cloud platforms like Node-RED, enabling remote access and visualization through a user-friendly web or mobile dashboard.

5.1 Circuit Diagram

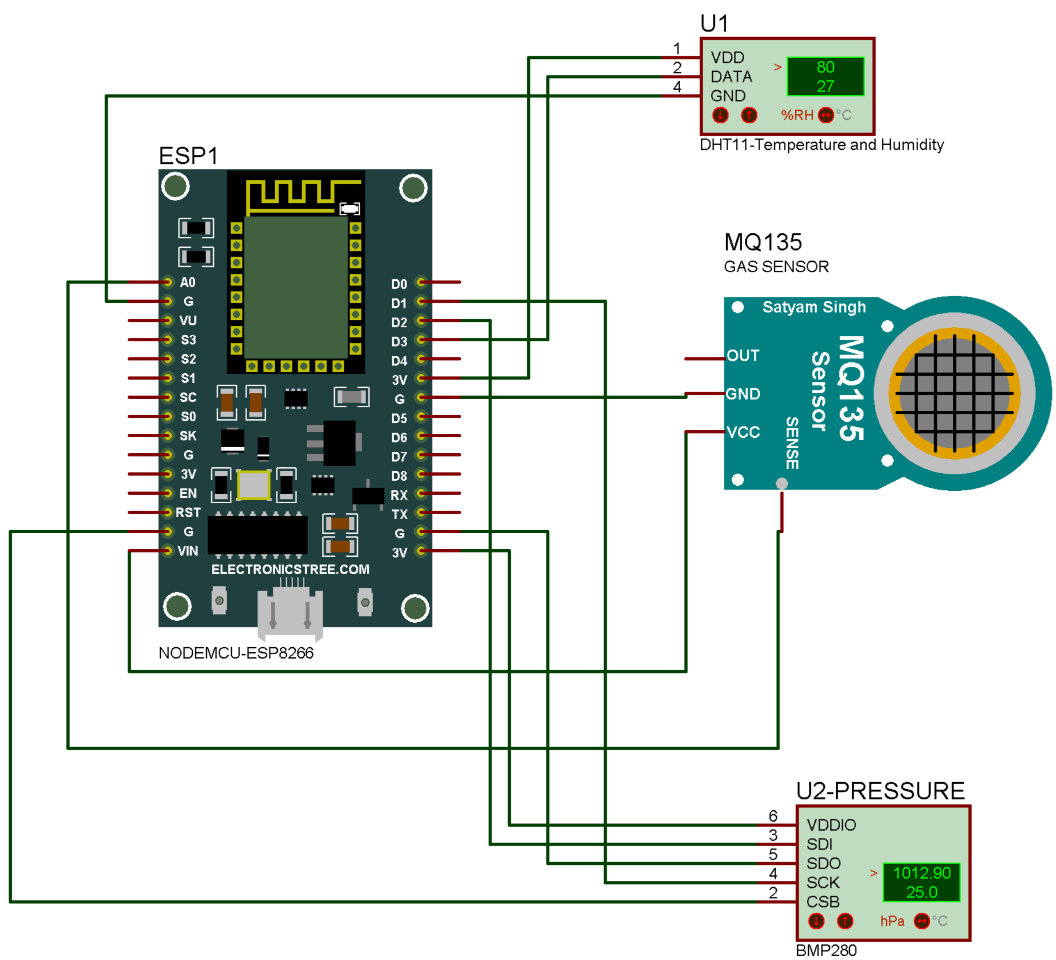


Figure 1: Weather Station Circuit Simulation Diagram

5.2 Interfaces (if applicable)

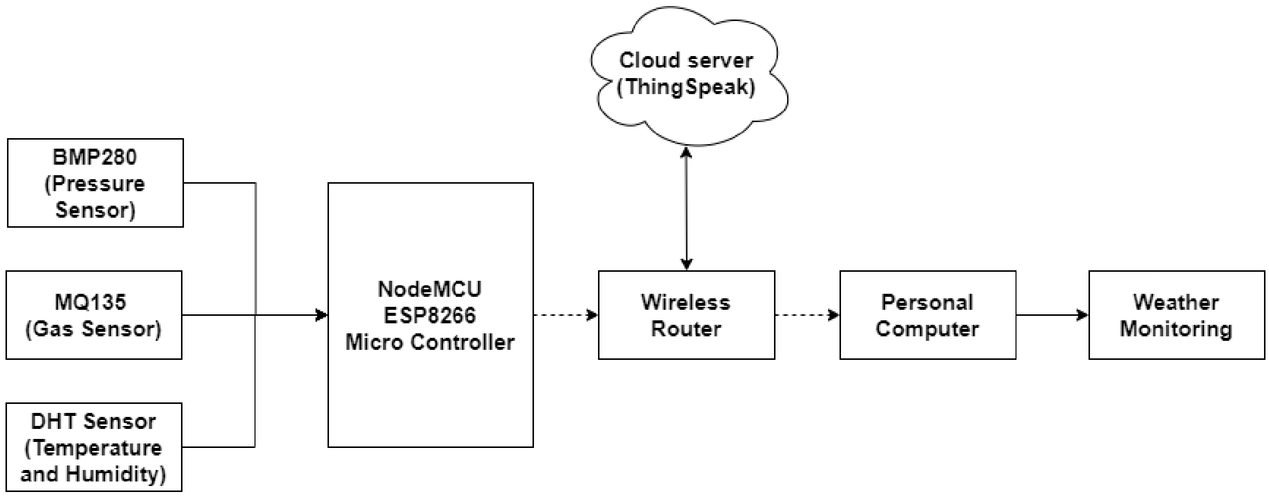


Figure 2: Block Diagram

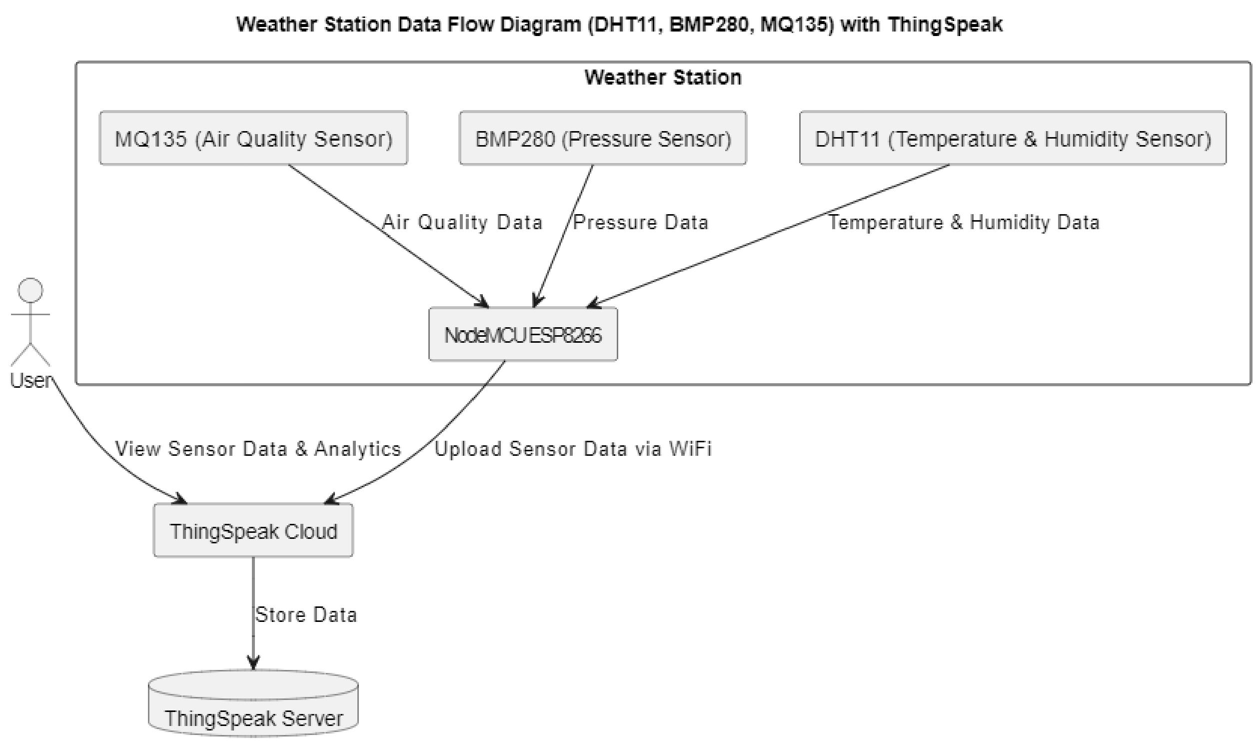


Figure 3: Data Flow Diagram

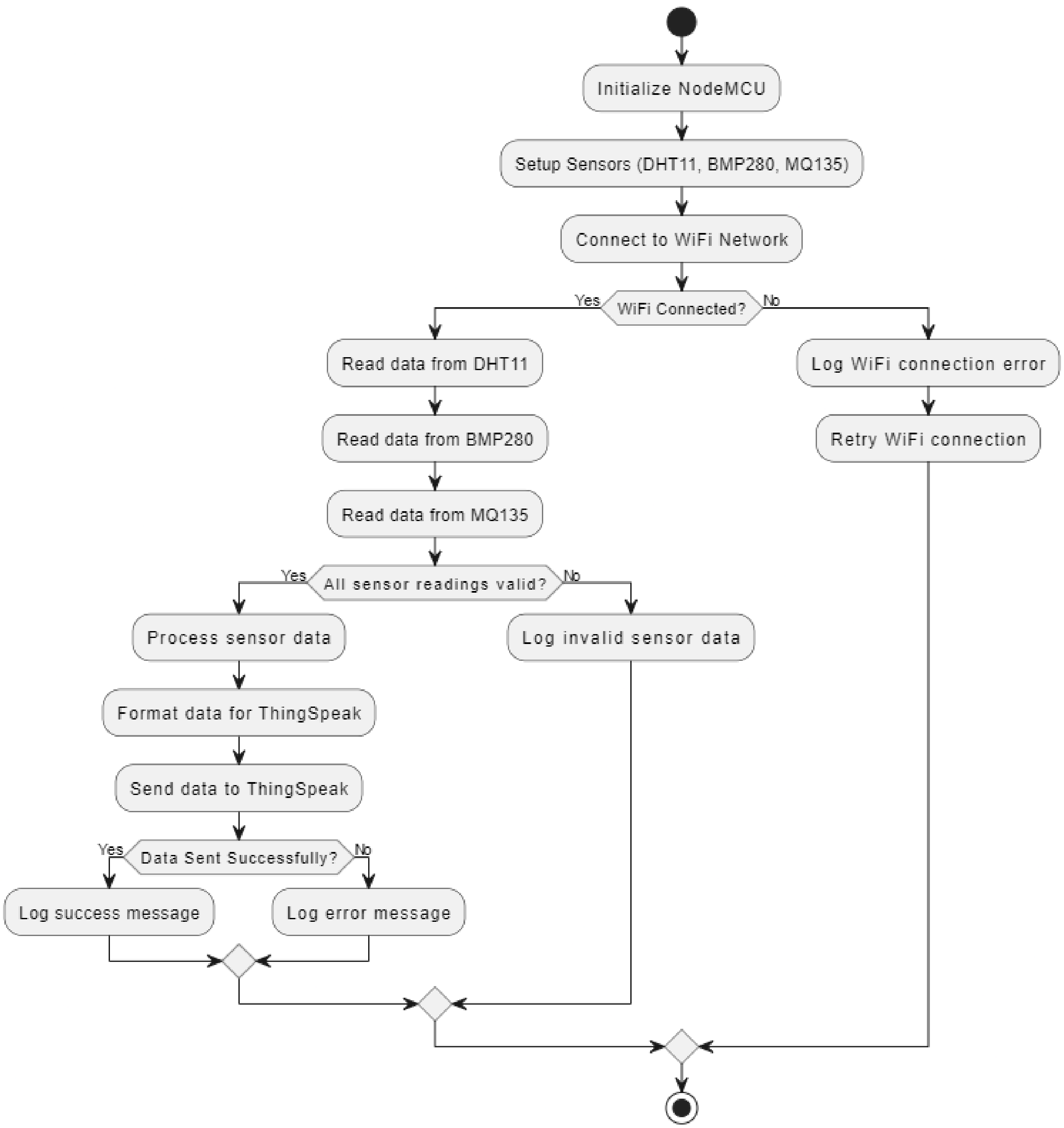


Figure 4: Flow Chart

6 Performance Test

Constraints and Design Considerations:

1. **Memory**:
   - The memory constraints of the ESP8266 remain the same. Since you're using ThingSpeak for cloud services, the reduced complexity compared to MQTT and Node-RED might alleviate some memory pressure. Using static allocation for the JSON buffer (e.g., 300 bytes) remains a good practice.
   - **Optimization**: You may still consider reducing the JSON size or leveraging more efficient data formats if memory usage becomes a bottleneck.

2. **MIPS (Speed and Processing Power)**:
   - The 80 MHz (or 160 MHz) processing speed is generally sufficient for sensor readings and data uploading to ThingSpeak. However, the BMP280 sensor may provide more data than the LDR, so the processing load could increase slightly.
   - **Optimization**: The 2-second publishing interval should still work well, but you could reduce the frequency if the number of sensors or the complexity of the computations increases.

3. **Accuracy**:
   - Replacing the DHT11 with the BMP280 sensor improves the accuracy of temperature and humidity readings, but it still requires regular calibration, especially if your use case demands high precision.
   - **Optimization**: For even better results, consider calibrating the BMP280 in different environments to improve accuracy.

4. **Durability**:
   - The durability concerns about sensors in harsh environments still apply, especially if your sensor array is deployed outdoors or in challenging conditions. The BMP280 has a similar level of environmental resilience but is more durable than the DHT11.
   - **Optimization**: Use protective casings to shield the sensors from physical damage and ensure the microcontroller operates within its environmental limits.

5. **Power Consumption**:

    - Since you're using ThingSpeak and avoiding continuous MQTT communication, the power consumption should be slightly lower than if MQTT were used continuously. Still, continuous WiFi communication remains a challenge for battery-powered setups.
    - **Optimization**: Use deep sleep modes during idle periods to save power. You can schedule the microcontroller to wake up and publish data at specific intervals.

6. **Network Stability**:

    - While ThingSpeak is cloud-based, network instability can still disrupt the system's ability to push data to the cloud. However, ThingSpeak supports buffering, and you can implement local caching during outages.

    - **Optimization**: Cache data locally on the ESP8266 when there's no network connectivity and upload it when the connection is restored.

Testing and Results (Updated):

- **Memory Usage**: Given that you're using ThingSpeak and potentially smaller data packets, memory usage should be stable within the expected limits.

- **MIPS**: The ESP8266 should handle the sensor readings and data transmission with a 2-second interval effectively.

- **Accuracy**: With the BMP280 sensor, expect more accurate readings, especially for temperature and pressure.

- **Power Consumption**: The system should still consume about 70mA during operation, but reducing WiFi usage through deep sleep modes will help lower the overall power consumption.

Future Improvements:

- You could explore different cloud services for data visualization if ThingSpeak's features become limiting.

- If battery efficiency is critical, consider exploring low-power options like the ESP32, which offers more advanced power-saving modes.

- Finally, testing your system under various environmental conditions will give you more insights into the sensor's real-world performance and durability.

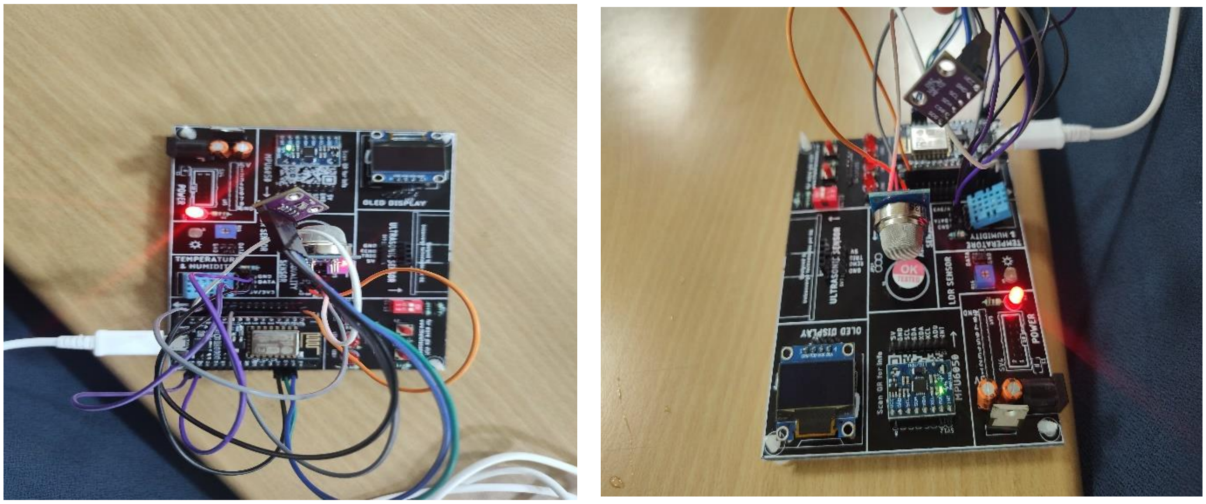


Figure 5 and 6: Practical implementation of this project

6.1 Test Plan/ Test Cases

1. **Objective**:
   - Ensure the system's **accuracy**, **reliability**, **real-time data transmission**, **power efficiency**, **scalability**, and **effective data visualization**.
2. **Data Accuracy Test**:
   - Compare sensor readings (temperature, humidity, pressure, and air quality) with calibrated external devices.
   - Confirm that the sensors are within an acceptable error margin of ±5%.
3. **Real-Time Data Transmission Test**:
   - Check if sensor data is transmitted to ThingSpeak with minimal latency.
   - Ensure that data appears on the cloud platform within 10 seconds of being collected.
4. **Power Efficiency Test**:
   - Evaluate system's battery consumption when in operation.
   - Ensure that the system enters deep sleep mode effectively to conserve power.
   - Aim for at least one week of battery life.

5. **Scalability Test**:
    - Add additional sensors and locations to the system.
    - Ensure that the system can handle increased data aggregation and storage on ThingSpeak without delays.
6. **Reliability Test**:
    - Run the system continuously for 24 hours to assess stability.
    - Monitor for any disconnections or sensor malfunctions.
    - Aim for stable, uninterrupted operation.
7. **Data Visualization Test**:
    - Ensure that data displayed on ThingSpeak's dashboard is accurate and clearly presented.
    - Use graphs and gauges to visualize data.
    - Confirm that data updates in real-time and provides a clear view of current and historical weather conditions.
8. **Expected Outcome**:
    - The system should provide reliable, accurate, and real-time weather data with low power consumption.
    - The data visualizations should be easy to understand, meeting the requirements for a functional, scalable, and reliable weather station solution.

6.2 Test Procedure

1. **Preparation**:

   - Ensure all hardware components (NodeMCU, sensors, display, power source) are connected and operational.
   - Set up cloud services (e.g., ThingSpeak).
   - Verify Wi-Fi connection for internet access.

2. **Execution**:

   - Follow the test cases outlined above in sequence, ensuring that each test scenario is performed under controlled conditions.
   - Document results for comparison with expected outcomes.

3. **Verification**:

   - Cross-check sensor readings with external devices where applicable.
   - Monitor cloud platform updates for real-time data verification.

4. **Review**:

   - Evaluate the results of each test case, ensuring that the system functions as expected under all conditions.
   - Any failures should be addressed and retested until the desired performance is achieved.

6.3 Performance Outcome

1. **Data Accuracy**:

   - Sensors should consistently produce data within an acceptable range when compared to a calibrated external device, ensuring reliable readings.

2. **Real-Time Data Transmission**:

   - The weather station should transmit data in real-time with minimal latency, ensuring that the data is available almost immediately on cloud platforms like ThingSpeak.

3. **Power Efficiency**:
    - The system should effectively utilize low power modes (deep sleep) for extended operation on battery power, ensuring sustainability for off-grid deployment.
4. **Scalability**:
    - The system should perform well when scaled to multiple sensors or locations, with data aggregation and storage in the cloud platform functioning without delays.
5. **Reliability**:
    - The system should operate reliably over long periods, without frequent disconnects or sensor malfunctions, even in extreme weather conditions.
6. **Data Visualization**:
    - The data displayed on dashboards should be accurate, visually clear, and updated in real-time, providing insights into current and historical weather conditions.

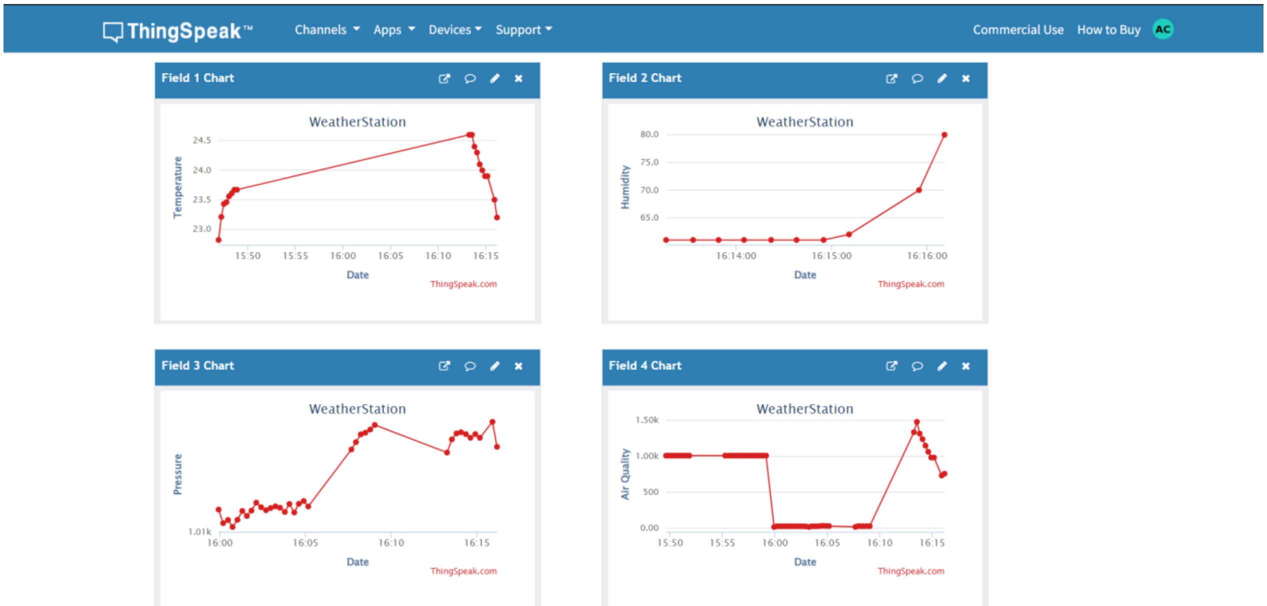


Figure 7: Data Visualization (Chart) using ThingSpeak

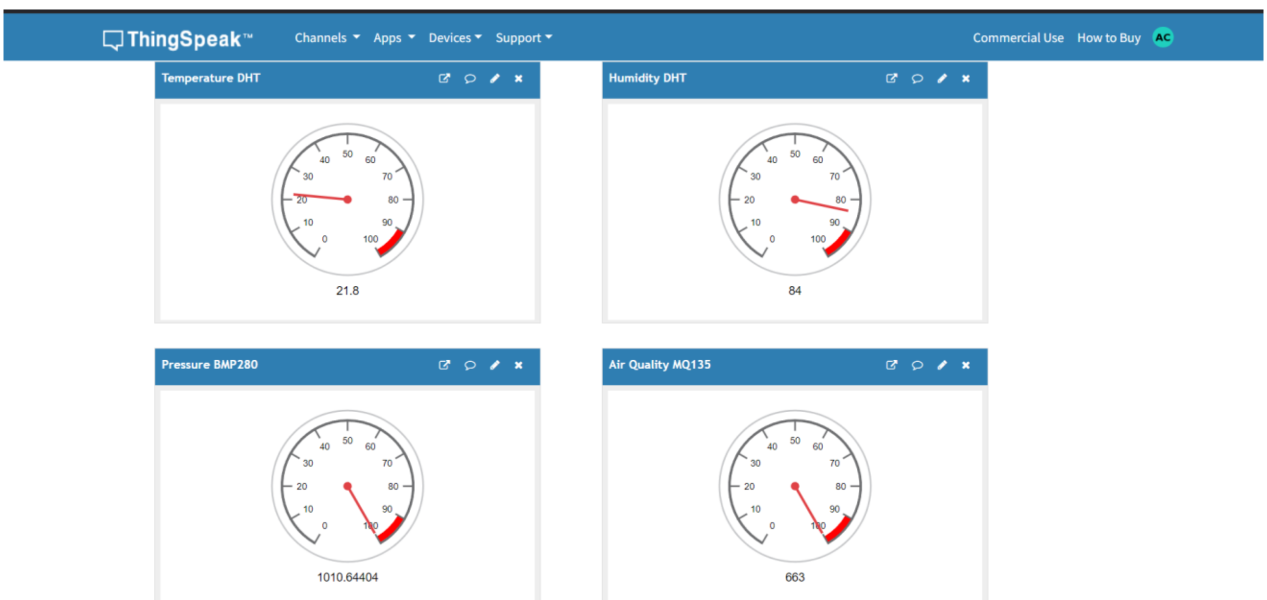


Figure 8: Data Visualization (Gauge) using ThingSpeak

7 My learnings

The weather station project using NodeMCU ESP8266 provided valuable hands-on experience in configuring the device as the central controller of an IoT system. I successfully interfaced multiple sensors, including the BMP280 for temperature and pressure, a humidity sensor, and an air quality sensor, to collect real-time environmental data. This project deepened my understanding of NodeMCU's analog pin limitations and enabled me to implement efficient techniques for managing multiple sensors. I honed my skills in sensor calibration, data acquisition, and preprocessing to ensure accurate measurements. Using ThingSpeak, I designed a cloud-based data monitoring system to visualize real-time weather data through charts and dashboards, showcasing temperature, pressure, humidity, and air quality. I explored ThingSpeak’s integration for reliable data storage and visualization, utilizing APIs for seamless communication between the NodeMCU and the cloud. The project also involved deploying automation workflows to trigger alerts based on specific conditions like high temperatures or poor air quality, enhancing local decision-making capabilities. Troubleshooting sensor connectivity issues and optimizing code for resource-constrained IoT devices improved my problem-solving skills. Overall, this project enriched my knowledge of IoT ecosystem integration and highlighted the benefits of cloud-based platforms like ThingSpeak for effective data management and analysis.

8 Future work scope

The weather station project using NodeMCU ESP8266 provided hands-on experience in configuring the device as a central controller of an IoT system. By interfacing multiple sensors, including the BMP280 for temperature and pressure, a humidity sensor, and an air quality sensor, I collected real-time environmental data. This project enhanced my understanding of NodeMCU's analog pin limitations and improved my ability to manage multiple sensors efficiently. I gained valuable skills in sensor calibration, data acquisition, and preprocessing for accurate measurements. Using ThingSpeak, I developed a cloud-based data monitoring system to visualize weather data through charts and dashboards, showcasing temperature, pressure, humidity, and air quality. I explored ThingSpeak's integration for reliable data storage and visualization while utilizing APIs for seamless communication between the NodeMCU and the cloud. The project also involved deploying automation workflows to trigger alerts based on specific conditions, enhancing local decision-making capabilities. Troubleshooting sensor connectivity issues and optimizing code for IoT devices improved my problem-solving skills. Overall, this project deepened my understanding of IoT ecosystem integration and highlighted the benefits of cloud-based platforms like ThingSpeak for efficient data management and analysis.

The project's future scope includes integrating additional sensors like rainfall gauges, wind speed sensors, and UV index sensors for more comprehensive data. Advanced data analytics and machine learning models can be used to predict weather patterns and generate actionable insights. Developing a mobile app will provide real-time updates and notifications, improving user accessibility. Incorporating renewable energy sources like solar panels will make the weather station sustainable for remote use. Deploying sophisticated Edge AI models will enable advanced local decision-making, such as predicting anomalies and triggering actions. Interoperability with other IoT systems and smart devices can support smarter environmental control. Finally, scalability can be achieved for applications in agriculture, disaster management, and smart cities.