Weight Scaling

Made by : Salaheldin elshafey 20100205
Ali gaber 20100431
Laila Labib 20100133
Hania Helmy 20100450

1- OverView

This project leverages the ESP8266 microcontroller, HX711 weight sensor module, and MQTT protocol to create a smart weight monitoring system. The system is designed to measure weight data, process it, and publish the readings to a remote MQTT broker for real-time monitoring. This report includes details from the provided first_try.ino code, highlighting the technical implementation, challenges encountered, and potential applications.
The system has been designed with scalability in mind, enabling future integration with more sophisticated IoT ecosystems. Its primary focus is to address the need for remote and automated weight monitoring, which is a common requirement in industries like logistics, healthcare, and agriculture.

2- Components

- Hardware:

The hardware used in this project includes the ESP8266 microcontroller, which provides Wi-Fi connectivity and acts as the central processing unit for the system. It handles data processing and communication with the MQTT broker. The HX711 load cell amplifier amplifies the electrical signals from the load cell and converts them into digital readings. The load cell is a key component that measures the applied weight and outputs an electrical signal corresponding to the load. For interfacing the HX711 module with the ESP8266, GPIO pins DT (D5) and SCK (D6) are used. These connections ensure seamless communication and accurate data transfer from the weight sensor to the microcontroller. The system is

powered using a standard USB connection, which also simplifies its deployment in various scenarios.

- Software :

The software for this project is developed in Arduino (C++), leveraging a modular approach for enhanced readability and maintainability. Key libraries include ESP8266WiFi.h for establishing and maintaining Wi-Fi connections, PubSubClient.h for implementing MQTT communication, and HX711.h for interfacing with the HX711 module. These libraries provide robust and well-documented APIs, reducing development time and ensuring reliability.

- Network :

The network configuration includes Wi-Fi credentials with SSID Ali Gaber and password Aligaber24!. The project uses the public MQTT broker broker.hivemq.com to facilitate data exchange. The MQTT topic weight_monitoring is used to organize and publish weight readings, allowing multiple subscribers to access the data concurrently. This setup leverages MQTT's lightweight protocol, making it ideal for IoT applications with limited bandwidth.

3- Features

The system supports real-time weight monitoring by reading data from the HX711 module. To ensure accuracy and stability, the system averages 10 readings before publishing the data. This reduces noise and compensates for minor fluctuations in sensor readings. Remote monitoring is facilitated through MQTT, enabling users to access weight data in real time via the weight_monitoring topic.

Wi-Fi connectivity is a cornerstone of the system, ensuring that data is transmitted seamlessly to the MQTT broker. The ESP8266 microcontroller includes robust reconnection logic, automatically reestablishing connections to both the Wi-Fi network and MQTT broker if they are disrupted. Debugging support is integrated through the Serial Monitor, providing detailed logs for testing and troubleshooting.

4- Code functionality

- Setup phase

The setup initializes the HX711 module using scale.begin(DT, SCK) to prepare it for weight measurement. The Wi-Fi connection is established through the connectToWiFi() function, which includes retry mechanisms to handle connection failures. The MQTT client is configured using connectToMQTT(), ensuring that the system is ready for data transmission.

- Main loop

The main loop continuously monitors the connection to the MQTT broker, attempting reconnection if needed. Weight data is read using scale.get_units(10), which calculates an average of 10 readings for enhanced accuracy. The processed data is then published to the MQTT topic using client.publish(). This loop ensures that the system operates in real time, providing continuous updates.

- Helper Functions

connectToWiFi(): Manages Wi-Fi connectivity, displaying connection status through the Serial Monitor. **connectToMQTT()**: Handles MQTT

broker connectivity, including retry logic and error handling.

5- Benefits

The system offers significant advantages for both industrial and personal applications. Remote accessibility enables users to monitor weight data from any location with internet access, reducing the need for physical presence. The project's cost-effectiveness stems from its use of affordable hardware and free MQTT services. Scalability is another key feature, allowing integration with additional IoT devices and systems to expand its functionality. Furthermore, the user-friendly design simplifies deployment and data access, making it suitable for a wide range of users.

6- Future Enhancements

To enhance the system's security, future versions can implement secure MQTT communication with TLS encryption and add user authentication for Wi-Fi and broker access. Data logging capabilities can be introduced to store historical weight data in a centralized database, enabling trend analysis and predictive analytics. Advanced notification systems, such as email or SMS alerts, can improve usability by providing instant updates for predefined weight thresholds.

Developing a dedicated mobile or web application will further enhance the user experience, offering intuitive dashboards and real-time data visualization. Optimizing the system's power consumption is another priority, particularly for portable or battery-powered deployments. These enhancements will increase the system's versatility and appeal to a broader audience.

7- Conclusion

The IoT Weight Monitoring System exemplifies the power of IoT technologies in delivering real-time monitoring solutions. By integrating state-of-the-art hardware, robust software, and efficient networking protocols, the project addresses key challenges in weight monitoring. Its modular design and scalability make it adaptable to various industries, including logistics, healthcare, and agriculture. With planned enhancements, the system has the potential to become a comprehensive solution for smart monitoring needs.