MQTT-Based IoT Environmental Monitoring and Remote Actuation using ESP32

Dikssha Bhonsle, and Rahul Johari

University School of Automation and Robotics (USAR),
Guru Gobind Singh Indraprastha University, New Delhi, India

Abstract

The rapid growth of the Internet of Things (IoT) has enabled seamless communication between physical devices and cloud platforms, supporting real-time monitoring, automation, and remote control [8]. This paper presents the design and implementation of an IoT-based environmental monitoring and control system using an ESP32 microcontroller [3], the lightweight MQTT messaging protocol, and the HiveMQ cloud broker. The ESP32 acquires temperature and humidity data from a DHT11 sensor and publishes it to an MQTT topic for real-time monitoring by subscribed clients.The system also demonstrates bidirectional communication, where an external LED is remotely controlled through MQTT commands, illustrating cloud-based actuation. The lightweight nature of MQTT [12] ensures efficient and low-latency data exchange suitable for resource-constrained embedded devices.

Keywords

Internet of Things (IoT), ESP32, MQTT, HiveMQ, DHT11 Sensor, Remote Monitoring, Actuator Control

Introduction

The rapid growth of the Internet of Things (IoT) has enabled seamless interaction between physical devices and cloud platforms, facilitating real-time monitoring, automation, and remote control [7], [8]. IoT systems integrate sensors, embedded controllers, communication protocols, and cloud services to create intelligent and interconnected environments.

With increasing adoption across domains such as smart homes, healthcare, agriculture, and industrial automation, IoT has become a critical component of modern digital infrastructure [9], [10].

Despite the growing adoption of IoT technologies, many real-world deployments face challenges related to efficient data transmission, system scalability, and reliable cloud integration. Resource-constrained embedded devices require communication protocols that minimize bandwidth usage while maintaining real-time responsiveness [12]. Addressing these challenges is essential for developing practical IoT systems that can operate effectively in dynamic environments.

At the core of an IoT system is a microcontroller that acquires data from sensors and communicates with cloud services. In this work, the ESP32 microcontroller [3], [4] is deployed due to its low cost, integrated Wi-Fi capability, and suitability for embedded IoT applications. Environmental parameters such as temperature and humidity are monitored using a DHT11 sensor [6], demonstrating a basic yet practical sensing mechanism for real-time data acquisition.

Efficient data transmission in IoT environments requires lightweight communication protocols that operate reliably on resource-constrained devices. To address this requirement, the proposed system utilizes MQTT, a publish–subscribe messaging protocol [1], [2], [12], optimized for low latency and minimal bandwidth consumption. The HiveMQ public broker [5] is used to facilitate cloud communication, enabling reliable data visualization and remote command exchange through web-based clients.

In addition to environmental monitoring, the system demonstrates remote actuation through an LED connected to the ESP32. The device subscribes to a control topic and responds to ON/OFF commands published by a remote client, illustrating bidirectional communication between cloud platforms and physical devices. This project presents a complete IoT workflow encompassing data acquisition, wireless communication, cloud integration, remote monitoring, and actuation, providing a practical foundation for scalable and real-world IoT applications.

Literature Survey

The Internet of Things (IoT) has emerged as a key technological domain enabling seamless interaction between physical devices and cloud platforms for real-time monitoring, automation, and intelligent decision-making [7], [8]. Prior research highlights the widespread adoption of IoT systems across smart homes, healthcare, agriculture, and industrial automation, with microcontrollers and wireless communication forming the backbone of these applications [9], [10]. A recurring theme in the literature is the requirement for low-power and lightweight communication protocols suited for resource-constrained embedded devices.

Traditional protocols such as HTTP introduce high latency and overhead, motivating the development of MQTT (Message Queuing Telemetry Transport), a lightweight publish–subscribe protocol optimized for efficient data transfer [1], [2], [12]. Studies emphasize MQTT's scalability, reliability on unstable networks, and suitability for real-time IoT deployments. Cloud-based MQTT brokers—including Mosquitto and HiveMQ—play a crucial role in routing messages and ensuring seamless device-to-device and device-to-cloud communication. HiveMQ, in particular, is widely used in experimental and educational IoT systems due to its WebSocket interface and stable public broker support [5].

The ESP32 microcontroller has been frequently adopted in IoT research for its integrated Wi-Fi, low power consumption, and dual-core processing capabilities, making it suitable for MQTT-based sensor networks [3], [4]. Temperature and humidity monitoring is commonly implemented using low-cost sensors such as the DHT11, valued for its simplicity and accessibility in prototype applications [6]. Existing studies show that such sensors, when combined with cloud platforms, can effectively support environmental monitoring and automation tasks. Literature on actuator control further demonstrates how IoT microcontrollers can process MQTT commands to operate LEDs, relays, and other devices, forming the basis of smart home and industrial control systems [12].

Overall, the literature strongly supports the architectural choices in this project—ESP32 for communication and processing, MQTT for efficient messaging, HiveMQ as the broker, DHT11 for environmental sensing, and LED-based actuation aligning well with validated IoT design methodologies in contemporary research [7], [12].

Materials and Methods

The proposed IoT-based environmental monitoring and control system consists of hardware components, software tools, and cloud communication services working together to enable real-time data exchange and remote actuation. The core processing unit of the system is the ESP32 microcontroller, [3], [4], selected for its integrated Wi-Fi capability, low power consumption, and suitability for embedded IoT applications.

For data acquisition, a DHT11 temperature and humidity sensor [6] was interfaced with the ESP32. The ESP32 is configured to connect to a wireless local area network using its built-in Wi-Fi functionality, enabling direct communication with the MQTT broker over the internet.

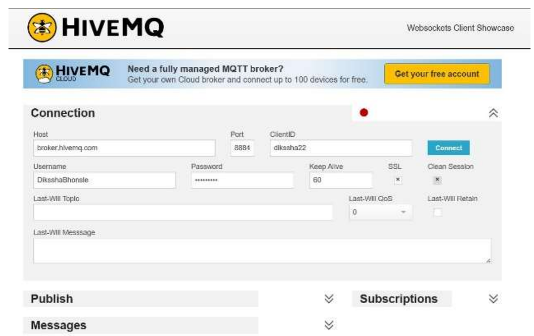

Fig. 1: HiveMQ WebSocket client interface showing the MQTT connection parameters configured during system setup.

Sensor readings obtained from the DHT11 are structured into lightweight JSON format before transmission, allowing easy parsing across different client platforms. This data representation improves interoperability and ensures compatibility with web-based and mobile MQTT clients. The sensor provides digital readings of environmental parameters, which are processed by the microcontroller and formatted into structured JSON data. These readings are transmitted wirelessly using the MQTT communication protocol [1], [2].

MQTT (Message Queuing Telemetry Transport) [1], [12] is employed as the messaging protocol due to its lightweight publish–subscribe architecture, which is well suited for resource-constrained devices. The system operates in a continuous loop where the ESP32 periodically acquires sensor data, publishes the formatted values to the designated MQTT topic, and remains subscribed to control topics for incoming commands. Messages published by remote clients are routed through the broker to the ESP32, which processes them in real time and triggers the corresponding actuation. This workflow demonstrates a closed-loop IoT system integrating sensing, communication, and response. The ESP32 acts as an MQTT client that publishes sensor data to a predefined topic and subscribes to a control topic for receiving remote commands. The HiveMQ public broker [5] was used to manage message exchange between the ESP32 and remote clients, enabling cloud-based communication without the need for a dedicated server. The system followed a cyclic operational methodology in which sensing, data transmission, and actuation are performed continuously. Sensor data was published at regular intervals, while incoming control commands are processed in real time. This ensures timely updates and responsive control without blocking system execution. Such a non-blocking and event-driven approach is essential for scalable IoT systems where multiple devices may operate concurrently [12].

For actuation, an external LED is connected to one of the ESP32's GPIO pins and configured as a digital output. Control commands are published to the subscribed MQTT topic, allowing the ESP32 to toggle the LED state accordingly. This demonstrates bidirectional communication between the cloud and physical devices. The system follows a cyclic operational methodology in which sensing, data transmission, and actuation are performed continuously. Sensor data is published at regular intervals, while incoming control commands are processed in real time. This ensures timely updates and responsive control without blocking system execution. Such a non-blocking and event-driven approach is essential for scalable IoT systems where multiple devices may operate concurrently.

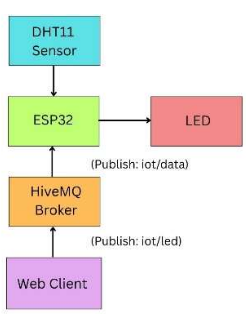

Fig 2: System architecture of the proposed IoT-based environmental monitoring and control system using MQTT.

Results and Discussions

The proposed IoT-based environmental monitoring and control system was implemented and evaluated to verify its ability to support real-time data transmission and remote actuation using a lightweight cloud-based communication framework. The system demonstrated continuous acquisition of environmental parameters, reliable message delivery through the MQTT protocol [1], and responsive control of an external actuator.

During operation, the ESP32 microcontroller successfully connected to the configured wireless network and established communication with the HiveMQ broker. Temperature and humidity values obtained from the DHT11 sensor were periodically read and formatted into structured JSON messages. These messages were published to a predefined MQTT topic, allowing subscribed clients to receive real-time updates without direct communication with the device. The use of topic-based messaging enabled efficient data dissemination while maintaining loose coupling between system components [1], [12].

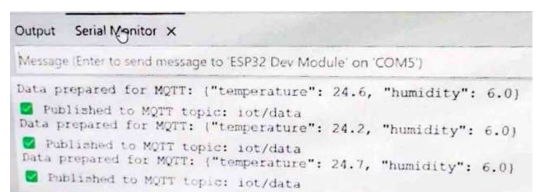

Fig. 3: Serial monitor output displaying JSON-formatted temperature and humidity data published via MQTT.

The data published by the ESP32 was consistently observed on the HiveMQ WebSocket client, confirming correct broker configuration and stable device-to-cloud connectivity.

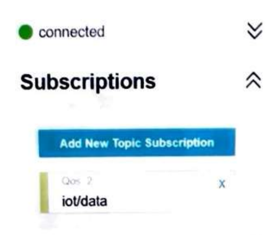

Fig. 4: The HiveMQ WebSocket client is subscribed to the topic iot/data, enabling reception of real-time temperature and humidity data published by the ESP32.

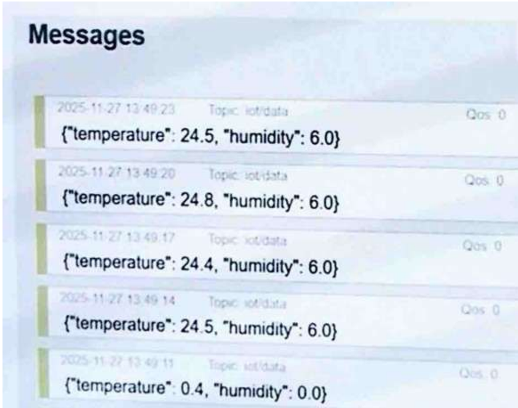

Fig. 5: Real-time sensor data received on topic iot/data

The structured JSON format ensured clarity and ease of interpretation, making the data suitable for use across multiple platforms, including web-based dashboards and mobile interfaces. This approach aligns with common IoT practices where standardized data formats improve system interoperability and scalability [8], [10].

A key outcome of the implementation was the successful demonstration of asynchronous communication. Unlike request–response models, the publish–subscribe architecture allowed sensor data to be transmitted independently of client availability [2]. Subscribers could join or leave the system dynamically without impacting the functioning of the ESP32, highlighting the flexibility and robustness of the MQTT-based design [12].

Overall, the system verified the effectiveness of combining embedded hardware, lightweight messaging protocols, and cloud-based brokers to achieve reliable real-time monitoring. The results confirm that the proposed architecture can serve as a foundational IoT framework capable of supporting distributed clients and scalable deployments [8], [9].

In addition to environmental data monitoring, the proposed system successfully demonstrated remote actuation through cloud-based command exchange. The inclusion of an external LED connected to the ESP32 served as a simple yet effective actuator for validating bidirectional communication within the IoT framework [9]. This functionality highlights a key aspect of IoT systems, where devices not only transmit data to cloud platforms but also respond to incoming control commands [10].

The ESP32 was configured to subscribe to a dedicated MQTT control topic, allowing it to listen continuously for incoming messages published by remote clients. When a control command such as "ON" or "OFF" was published to the topic, the broker forwarded the message to the subscribed ESP32 in real time [5].

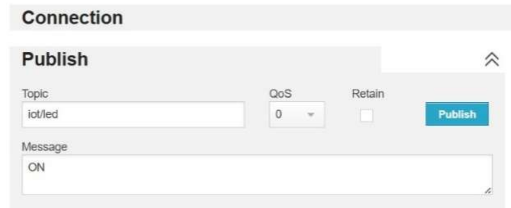

Fig. 6: Publishing ON command on topic iot/led for actuation

Upon receiving the command, the microcontroller processed the message and updated the GPIO output accordingly, resulting in immediate toggling of the LED state. This behavior confirmed the reliability of the publish–subscribe mechanism for command delivery [1], [2]. The actuation process was observed to be responsive and consistent, indicating stable communication between the cloud broker and the embedded device. The separation of data topics and control topics further improved system organization and ensured that sensing and actuation tasks operated independently without interference.

The bidirectional nature of communication demonstrated in this system forms the foundation of many real-world IoT applications, including smart home automation, remote equipment management, and industrial control systems [8], [10]. Cloud-based actuation allows users or automated platforms to initiate actions based on sensor readings, predefined rules, or external triggers. By implementing this interaction using MQTT, the system maintains efficiency while enabling real-time responsiveness [1].

Overall, the successful execution of remote actuation validates the ability of the proposed architecture to support closed-loop IoT systems. The integration of sensing, cloud communication, and device control illustrates how simple hardware components can be orchestrated through lightweight protocols to achieve reliable and scalable automation solutions [9], [12].

Conclusion

This work presented a comprehensive IoT-based environmental monitoring and control system that integrates embedded hardware, cloud-based communication, and remote actuation using standardized technologies. The implementation demonstrated how the ESP32 microcontroller, [3],

[4], combined with the MQTT publish–subscribe protocol and the HiveMQ broker, can support real-time data acquisition and bidirectional communication in a reliable and efficient manner [12].

The project successfully validated the core objectives of IoT system design, including device connectivity, cloud integration, data exchange, and remote control. The use of lightweight messaging ensured low communication overhead while maintaining responsiveness, making the system suitable for resource-constrained devices [1]. The successful execution of remote actuation further illustrated the practicality of closed-loop IoT architectures, which are widely used in automation and smart infrastructure [8], [10].

Beyond its functional outcomes, the project emphasizes scalability and extensibility as key design strengths. The modular architecture allows additional sensors, actuators, or client platforms to be integrated with minimal changes, enabling adaptation to diverse real-world applications [9].

Overall, the system serves as a solid foundational framework for future enhancements such as secure communication, persistent data storage, analytics, and intelligent decision-making. The concepts and methodologies explored in this work provide valuable insights into modern IoT architectures [8] and contribute meaningfully to academic learning and practical understanding in the field of Computer Science and embedded systems.

From an academic perspective, this work bridges theoretical concepts with practical system implementation, reinforcing the importance of hands-on experimentation in understanding IoT architectures. The process of integrating sensing, networking, and actuation components provided deeper insight into design trade-offs, protocol selection, and performance considerations in real-world embedded systems [9]. Such experiential learning is essential for developing robust solutions in emerging technology domains.

In future iterations, the system can be enhanced by incorporating secure authentication mechanisms, encrypted communication channels, and cloud-based data persistence for long-term analysis. Integration with additional sensing modules and intelligent decision-making algorithms can further extend its applicability to smart environments and industrial monitoring, and cyber-physical systems [8], [10]. These directions position the proposed system as a meaningful steppingstone toward advanced IoT development.

The development and evaluation of the proposed system also highlight the importance of reliability and robustness in IoT deployments. Through continuous operation and testing, the system demonstrated stable communication between the device and the cloud broker, ensuring consistent data transmission and command execution [5]. This reliability reinforces the suitability of MQTT-based architectures for real-time IoT applications operating under constrained computational and network resources [1].

The architecture and technologies used in this project align closely with industry-standard practices in modern IoT solutions. The use of a publish–subscribe communication model [1], cloud-hosted brokers [5], and modular embedded hardware [3], [4] mirrors real-world IoT deployments in domains such as smart homes, industrial automation, and environmental monitoring [8], [9]. As a result, the system provides practical exposure to design methodologies commonly adopted in professional and industrial environments.

Overall, this project not only achieved its technical objectives but also contributed significantly to the development of essential skills in embedded programming, cloud communication, and system integration. The insights gained from designing, implementing, and refining the system strengthen the foundation for further exploration in Internet of Things, distributed systems, and advanced computing applications. This work thus represents a meaningful academic and practical contribution in the evolving IoT landscape.

References

[1] A. Banks and R. Gupta, “MQTT Version 3.1.1,” OASIS Standard, 2014.

[2] OASIS, “MQTT Version 5.0,” OASIS Standard, 2019.

[3] Espressif Systems, “ESP32 Technical Reference Manual,” Espressif Systems Inc., 2023. [Online]. Available: https://www.espressif.com/

[4] Espressif Systems, “ESP32 Datasheet,” Espressif Systems Inc. [Online]. Available: https://www.espressif.com/en/products/socs/esp32

[5] HiveMQ, “HiveMQ MQTT Broker Documentation,” [Online].Available: https://www.hivemq.com/

[6] Adafruit Industries, “DHT11 Temperature and Humidity Sensor Datasheet,”

[Online]. Available: https://www.adafruit.com/

[7] K. Ashton, “That ‘Internet of Things’ Thing,” RFID Journal, 2009.

[8] R. Buyya et al., “Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[9] L. D. Xu, W. He, and S. Li, “Internet of Things in Industries: A Survey,” IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[10] M. Al-Fuqaha et al., “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[11] Espressif Systems, “ESP-IDF Programming Guide,” Espressif Systems Inc. [Online]. Available: https://docs.espressif.com/

[12] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “MQTT-S — A Publish/Subscribe Protocol for Wireless Sensor Networks,” Proc. COMSWARE, IEEE, 2008.