International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 57, No. 5, October, 2024, pp. 1503-1511
Journal homepage: http://iieta.org/journals/jesa

Smart Cloud Computing System for Environment Based on MQTT Protocol and Node MCU (ESP8266)

Anas Abduladheem Kamil[1*], Alaa J. Mousa[2], Laith A. Abdul-Rahaim[3]

[1] Babylon Technical Institute, Al-Furat Al-Awsat Technical University, Babylon 51001, Iraq
[2] Department of Technical Communication Engineering, Al-Shaab University, Baghdad 10001, Iraq
[3] Electrical Engineering Department, University of Babylon, Babylon 51001, Iraq

Corresponding Author Email: anas.altaee.bib11@atu.edu.iq

https://doi.org/10.18280/jesa.570526

Received: 5 September 2024
Revised: 7 October 2024
Accepted: 14 October 2024
Available online: 28 October 2024

Keywords:
MQTT, Internet of Things (IoT), heating, air conditioning appliances (HVAC)

ABSTRACT

In the age of high-tech automation and wireless technology, it is possible to make various devices come together in a given environment, enhancing comfort, energy savings, security, and affordability. We can observe environmental indicators remotely and manage relevant equipment with embedded systems' tiny, low-energy consumption. These devices' power consumption and network bandwidth utilization are critical. Thus, a low-power device must transfer data using minimal redundant protocol. Since Wi-Fi is so widely available, linking all devices within the space through one standard gateway makes sense. This study gives an overview of Message Queuing Telemetry Transport (MQTT), an ideal lightweight protocol for such applications because of its low power and bandwidth requirements. The proposed prototype uses MQTT on the ESP8266, a Wi-Fi-based expansion board. In the case of ESP8266, sensors and actuators are connected directly using Cayenne-based MQTT brokers to provide remote control and monitoring. It allows for efficient and effective communication within the network, ensuring stable data transmission and device management while minimizing power consumption and bandwidth usage. The project has shown how possible it is to use MQTT in the ESP8266 to support smart environments, leading to better IoT-integrated solutions.

1. INTRODUCTION

An intelligent environment monitors the environment from a distance and executes the intended actuation. Home appliances like TVs, glow lamps, propellers, etc., are connected through the same environmental gateway and assigned distinct IP addresses through intelligent environments. Any computer, laptop, or mobile phone can access and control environmental devices remotely. This can increase internal security, drastically reduce electricity loss, and improve living circumstances. The rapid advancement of technology has made machines appear intelligent in recent times. Computing power and intelligence support real-world devices so they can be configured appropriately. Remotely observing and controlling the equipment is becoming more accessible because of sensors connected to built-in gadgets and low-power wireless communication. IoT network is entirely of this. IoT concept has grown to encompass many applications, including intelligent transportation, intelligent healthcare, intelligent environments, intelligent industries, etc. [1-5]. When using intelligent gadgets for domestic purposes quickly gained traction, the concept of an intelligent environment was initially viewed as a better management strategy [6]. However, this idea has evolved beyond just turning on lights; it now includes intelligent device management, warming control systems, safety systems, remotely controllable devices, reinforcing power exhaustion, etc. [7-9].

To put it another way, IoT is a grid of wirelessly connected objects that communicate with one another and self-regulate according to predetermined rules. Nevertheless, the resources of these devices are constrained. For this reason, protocols with little weight (such as MQTT, CoAP, etc.) are used to transport data over wireless networks [10-14]. The remarkable increase in energy depletion speeds up the request, resulting in a corresponding increase in the monetary quantity of energy. This resulted in the creation of a formal request to implement intelligent environment applications with an emphasis on energy consumption in residential buildings. By the way, it generates low carbon output, desire reduction, and energy conservation episodes that have a fixed impact on the residents' energy exhaustion patterns [15, 16]. The intelligent environment is widely used to regulate security locks, lighting, HVAC appliances. Intelligent environmental enforcement heavily relies on wireless sensor networks, regardless of the purpose. Radio modules come in various species, the most well-known being Wi-Fi, ZigBee, 3G, GSM, Bluetooth, etc. Wi-Fi is profitable through various technologies because of its greater bandwidth, extensive coverage, ease of development, etc. [17]. However, Wi-Fi is chosen as the connection style in the prototype due to the growing number of Wi-Fi hotspots and scope sufficient to achieve the desired control and observation, and the devices are controlled out of MQTT protocol and proceeded using ESP8266. A summary of the hardware and software used.

1.1 Cloud computing in intelligent environments

Cloud computing plays a crucial role in deploying and managing intelligent environments. By leveraging cloud infrastructure, developers can offload heavy computational tasks, data storage, and complex analytics to powerful servers in data centers. Allows IoT devices to focus on data collection and primary processing, reducing hardware requirements and power consumption.

The cloud facilitates seamless integration and interoperability between IoT devices and systems. With cloud-based platforms, developers can create centralized management systems that provide a unified view of the entire intelligent environment. These platforms can also enable real-time monitoring, remote control, and automation of devices, enhancing the overall functionality and user experience [10, 11].

1.2 MQTT protocol for efficient communication

The MQTT protocol enables reliable communication between cloud services and IoT devices. It operates through a publish-subscribe model, simplifying network design. MQTT is lightweight and suitable for Node MCUs, reducing bandwidth and message transmission overhead. It also supports Quality of Service levels for reliable message delivery in intelligent environments.

1.3 Core IoT implementations

The Node MCU board is a versatile development board using the ESPO8266 microcontroller for IoT projects. It features a 32-bit RISC CPU, Wi-Fi, and multiple I/O's, making it suitable for various application domains. Its small size and Wi-Fi module enable connecting to the internet and other devices, enabling advanced intelligent settings.

2. RELATED WORKS

Intelligent environments are employed to provide energy efficiency, comfort, and security preferences. As an outcome, this project set up a Small Intelligent Environment System using an ESPA8266 microcontroller and WLAN for control. It can control and regulate appliances such as lights, humidity, and room temperature. Experimental results show that such functions as convenient control and control observation of the system can be done through an internet-connected device [18].

Sophisticated trends in communicative technologies and improved embedded systems' costs have created the context for intelligent environments with cloud computing and the IoT [19]. One technology that has found its use in this regard is the MQTT protocol, which is a robust and reliable method of transmitting real-time data between connected devices [20].

Therefore, the Internet of Things has changed how humans interface with objects in their environment by embedding connectivity and intelligence into them [21]. In this context, Node MCU has achieved a high position as a small, low-cost solution for Internet of Things applications [22].

Based on the ESP8266 microcontroller, the Node MCU can acquire data from the environment, process data, and transmit the obtained information to a cloud system through the MQTT protocol [23-25]. In addition, it eliminates the demand for the longest connection distance of traditional monitoring systems and the immediately dynamic and convenient part of the monitoring [26]. The study [27] covers IoT challenges and advances, particularly in device monitoring and control. The authors identify the absence of software and apps to handle diverse systems and hardware as a major IoT issue. Due to the growing number of IoT devices, device accessibility must be integrated. Another important element of IPv6 is that M2M concerns are better addressed than other issues. Scholars believe IPv6's numerous addresses will enable M2M communication. This context explains why smart cloud computing design and architecture are important.

Kumar and Dezfouli [28] recommended analyzing the MQTT protocol in detail. It is underlined for M2M and IoT applications in the study [28]. According to the authors, the protocol for querying message queues will remain popular due to its friendliness, simplicity, and low service fee on low-powered edge computing devices. A key point in MQTT protocol papers and articles is that the protocol is designed to help one client at a time. Therefore, it performs poorly when several client data arrive concurrently. The authors' argument is valid because Infinite Message Queuing Telemetry Transport (MQTT) is effective. After all, only one message is a focus, but edge devices waste battery and network bandwidth when processing many fuses.

The study [29] discusses using low-cost sensor networks and web-based computing resources to detect airborne contaminants. The authors emphasize that air quality monitoring indoors and outdoors is becoming more important, especially in airports, where structural factors like large numbers of people indoors cause outdoor pollution to spill. The authors briefly mention a low-cost sensor's text stabilization. The authors believe these technologies can usually solve environmental monitoring problems without major cost inputs. This is a drawback since it permits resource-poor people to adopt cheaper, breathable air quality monitoring technologies. The text stabilization of a low-cost sensor is briefly highlighted.

The study [30] discusses the security issues associated with numerous common implementations of the MQTT communication protocol. MQTT is one of the key protocols used in IoT, especially for less capable devices; hence, this research is crucial for developing smart cloud computing technology. The researchers studied how brokers and customers saw the MQTT standard and identified characteristics affecting profound application states.

3. PROPOSED SYSTEM

The proposed system is a comprehensive and secure home automation solution we developed. The proposed system utilized multiple sensors and mechanisms to achieve its objectives. The Smart Cloud Computing System for Environments incorporates temperature, humidity, and gas sensors, as illustrated in the picture. A C++ programming language interfaced the sensors with an Arduino ESP8266 Wi-Fi module. The router supplies an internet access point for the ESP. It employs the JSON programming language format to connect Arduino to the internet, access point, and web server. Three sensors—temperature, gas, and motion—are employed, and the ESP 8226 Wi-Fi Arduino has been programmed with the requisite codes for each sensor.

The ESP 8226 Wi-Fi Arduino has been configured with multiple libraries to ensure compatibility with the intended system. The sensor values are readied for transmission to the cloud server database after establishing an access point (utilizing our home network as the default). The cloud database was constructed using Apache and MySQL formats. The database and web server's location were stored on the computer's hard drive. The database connects to a web server via an HTTP request, which the server executes to retrieve measured information values. This process occurs in two phases: data collection in the first phase and result presentation in a categorized format using JSON in the second phase. The displayed information will indicate the current values of the sensors. The data will be refreshed every second to represent a real-time system. The results illustrate the sensors' rapid responsiveness and the system's design. The home automation system is seen in Figure 1.

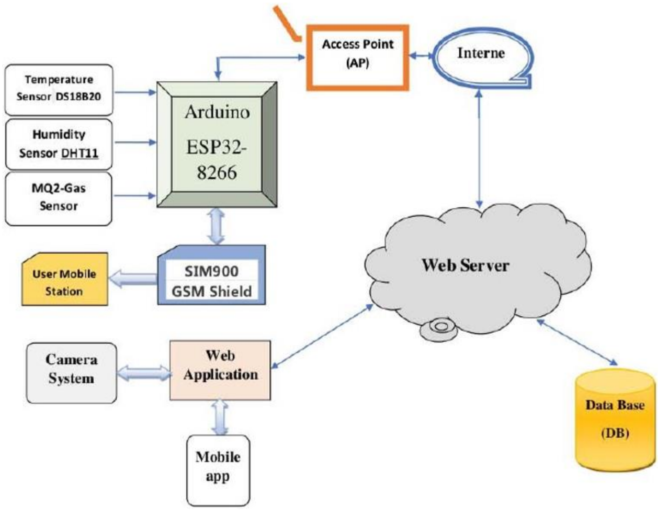


Figure 1. Proposed system

4. SYSTEM REQUIREMENTS

The Smart Cloud Computing System consists of three major layers: the Physical Layer (Sensors and Actuators), the Network Layer (ESP8266 and Wi-Fi Router), and the Cloud Layer (MQTT Broker and Cayenne Platform). The Physical Layer connects sensors like the DHT11 Sensor and LDR Sensor to the ESP8266, while the Network Layer manages data communication between the sensors and the cloud platform. The ESP8266 Node MCU is the gateway and MQTT client, collecting sensor data and controlling relays and LEDs. The Wi-Fi Router provides internet connectivity, enabling the ESP8266 to interact with the MQTT broker and the Cayenne platform. The Cloud Layer represents the smart cloud computing aspect of the system, with the MQTT Broker for efficient communication and the Cayenne Platform for data storage and analysis.

The system's data flow can be visualized as a continuous loop between sensors, the ESP8266, the MQTT broker, and the Cayenne Platform. The system can automatically decide whether to activate relays based on LDR readings or manually control them. The Cayenne platform includes a notification engine to send alerts when certain conditions are met.

4.1 Software requirements

The proposed system hardware and software are implemented in this work.

4.1.1 MQTT

Low-weight MQTT uses grid bandwidth with a two-byte header [31]. TCP-compliant MQTT sends node messages to servers. Internet of Things nodes with modest capabilities benefit from message-oriented MQTT. MQTT was established by IBM in 1999 and acknowledged by OASIS [32]. MQTT publish/subscribe. MQTT connections involve customers, servers, or public brokers as in Figure 2. MQTT enforcement messages contain data. MQTT allows grid-connected devices to share enforcement data. Clients publish and subscribe to MQTT. The publisher publishes enforcement data upon subscription. Servers link MQTT clients. Related clients can share enforcement data. Smartphones, sensors, etc., use MQTT. After verifying message transmission, MQTT clients notify brokers. MQTT brokers combine data. Enforcement information is sent to other MQTT clients.

MQTT helps clients protect their rights by focusing any misunderstandings on the broker. Publishers and subscribers are isolated since they don't have to acknowledge support or enforcement. Quality of service shared and configured control packets before delivering application data. MQTT control packets have payloads and changeable and fixed headers. See Figure 3. The MQTT control packets sent between clients and servers are CONNECT, CONNACK, PUBLISH, PUBACK, PUBREC, PUBREL, SUBSCRIBE, SUBACK, etc. While offline, the MQTT server buffers and provides all data to the client when the session is enabled [33].

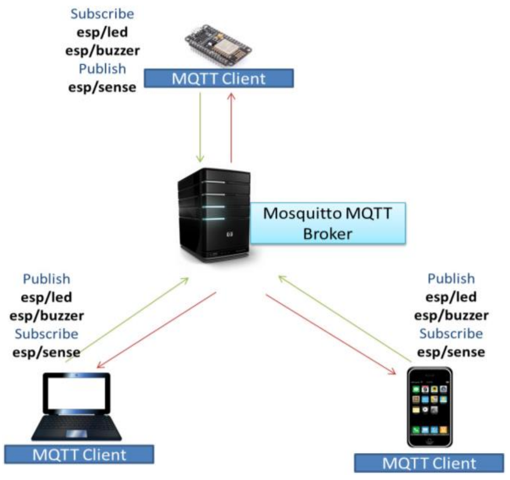


Figure 2. MQTT network

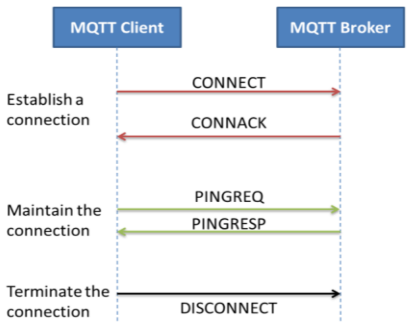
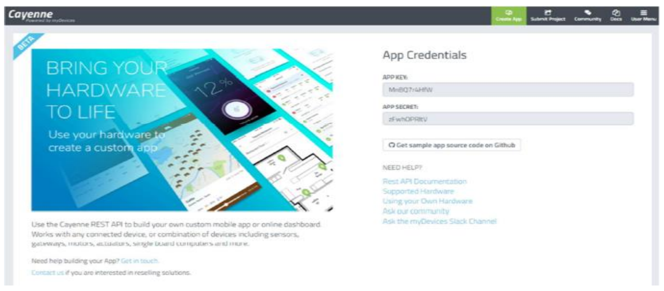


Figure 3. MQTT control signal

4.1.2 The Cayenne IoT platform

IoT scheme builder Cayenne is for engineers, designers, developers, and students to utilize in different IoT implementations. Cayenne accelerates IoT expansion. Cayenne offers iOS and Android apps and cloud-based online enforcement. A popular GUI in web and mobile versions is the dashboard. Evin Bromer founded 'My Devices' (Cayenne IoT rostrum) in 2013. We can remotely monitor and handle physical equipment using Cayenne mobile apps. Cayenne dashboard widgets may view data, construct principles, tab events, define declaration rules, and more. Cayenne connects motors, sensors, valves, lights, relays, and universal actuators. Cayenne works with Raspberry Pi, Arduino, ESP8266, LoRa, and more boards [16]. The dashboard can create prioritization principles when the ESP8266 is connected to the Cayenne cloud program and has sensors. These guidelines may need actions after a certain readability. A simple example is actuating a propeller when the ambient temperature reaches a certain degree. The mechanism can send email and text notifications. The dashboard's publication guidelines can be adjusted. The platform is user-friendly and features a drag-and-drop GUI for project development. IoT and Industry 4.0 are ready, allowing task monitoring and control [34]. The Quality-of-Service approach visualizes heating supervision statistics. As in Figures 4 and 5, the platform has various widgets and data visualization tools and supports multiple IoT devices. It supports HVAC supervision, water level and temperature monitoring, and telemetry solutions. It has a dynamic public IP for IoT hosting and device templates for quick integration. The platform offers security, encryption, and authentication support for various application domains.


Cayenne
Create App
Submit Project
Community
Docs
User Menu
BETA
BRING YOUR HARDWARE TO LIFE
Use your hardware to create a custom app
12%

App Credentials

APP KEY: MnBQ7r4HfW

APP SECRET: zFwhOPRtV

Get sample app source code on Github

NEED HELP?

Rest API Documentation
Supported Hardware
Using your Own Hardware
Ask our community
Ask the myDevices Slack Channel

Use the Cayenne REST API to build your own custom mobile app or online dashboard. Works with any connected device, or combination of devices including sensors, gateways, motors, actuators, single board computers and more.

Need help building your App? Get in touch.

Contact us if you are interested in reselling solutions.

Figure 4. IoT cayenne scheme builder

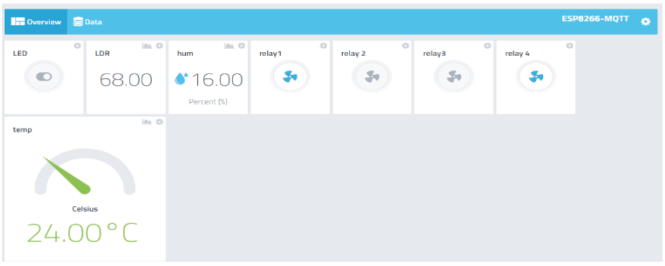


Figure 5. GUI dashboard

4.2 Hardware requirements

The ESP8266 is an affordable development board with integrated Wi-Fi, I/O, I2C, Universal Asynchronous Receiver-Transmitter, Analog to Digital Converter, and Pulse Width Modulation for quick prototyping of Internet of Things products, as shown in Figures 6 and 7. It can be developed using Lua-based ESP or Arduino IDE, and its low cost makes it suitable for intelligent environment systems. Figure 8 shows the Light Dependent Resistors (LDRs), widely used sensors for light measurement due to their simplicity, low fabrication cost, and quick response. However, they have performance limitations like low sensitivity and high noise. Researchers have improved LDR performance using innovative materials and designs, such as semiconductors in carbon compound-based sensor designs and hybrid resistors with thin film transistors. This has expanded applications in consumer electronics, life sciences, intelligent building renovation, and advanced manufacturing. New algorithms for data processing also show promise in improving LDR performance. Successful applications could lead to new business creation and revolution in IT.

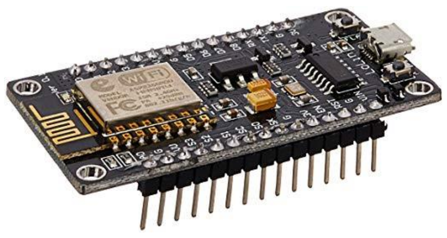


Figure 6. ESP8266

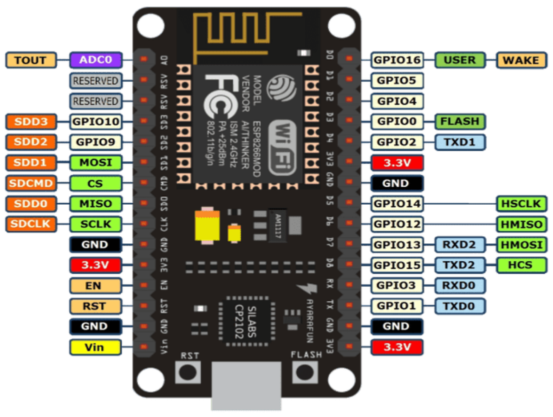


Figure 7. ESP8266 pins

Figure 8. LDR sensor

5. SYSTEM ENVIRONMENT

This section describes a system that uses sensor data to control devices attached to readings, comparing it to threshold values in the Arduino Uno's memory. The data is sent via USB to a broker, which displays it on a computer screen using C# programming. The broker sends packet data to the cloud via Wi-Fi, using HTTPS protocol for data protection. When a sensor detects abnormal readings, Fog Computing alerts the homeowner, and readings are saved in the cloud via MS SQL database as shown in Figures 9 and 10.

5.1 Temperature and humidity

The humidity and temperature controls will only work when people are there, saving electricity and shutting down the equipment when no one is. The system will activate the cooling system if the temperature or humidity is above 32°C or 70% while humans are present. It turns off when the temperature dips below 30°C and humidity hits 60%. These percentages can be adjusted to meet user demands. Figure 11 shows how this condition will be executed automatically. The temperature reading will also be presented on the broker's (Fog) computer screen via the application interface, transferred immediately to the cloud, and displayed in real-time in the web application, as shown in Figures 12 and 13.

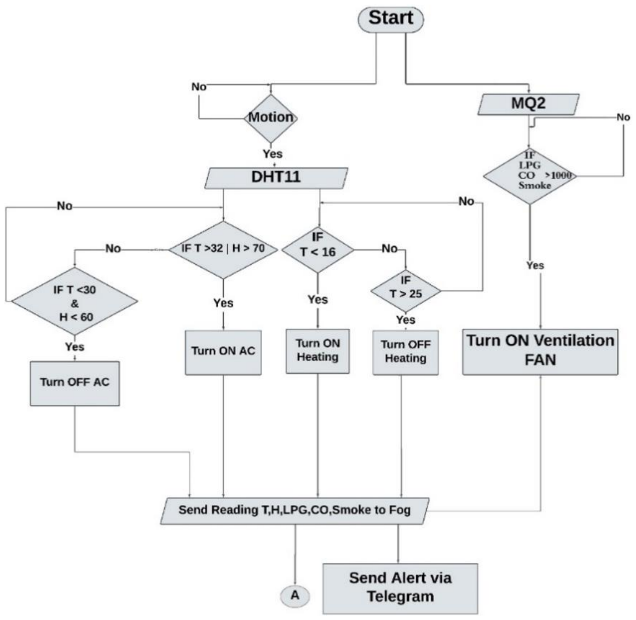


Figure 9. End node system operation flow chart

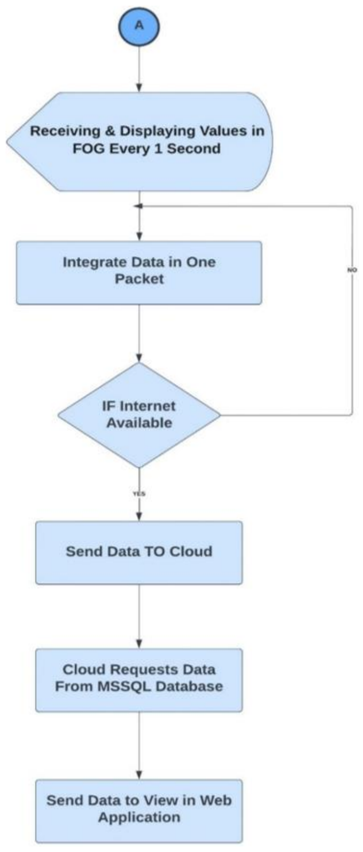

A
Receiving & Displaying Values in FOG Every 1 Second
Integrate Data in One Packet
NO
IF Internet Available
YES
Send Data TO Cloud
Cloud Requests Data From MSSQL Database
Send Data to View in Web Application

Figure 10. Broker node system operations

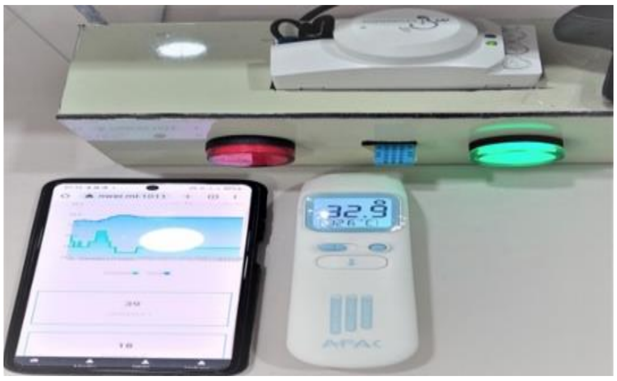
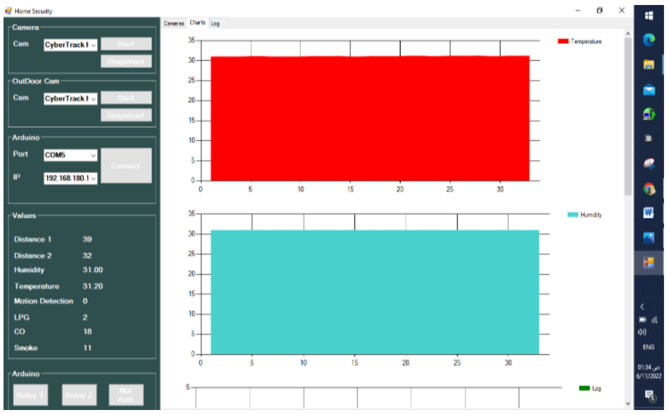


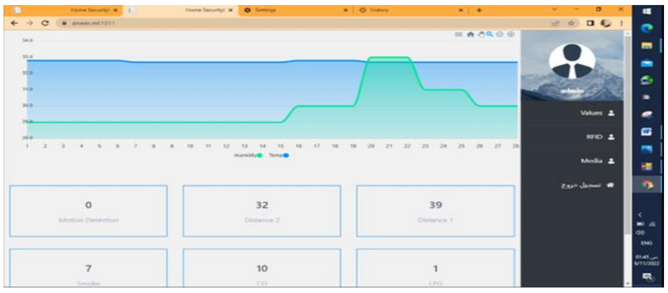
Figure 11. Temperature more than 32°C



Values

Distance 1	39
Distance 2	32
Humidity	31.00
Temperature	31.20
Motion Detection	0
LPG	2
CO	18
Smoke	11

Figure 12. Temperature measured from fog user application



0	32	39
Motion Detection	Distance 2	Distance 1
7	10	1
Smoke	CO	LPG

Figure 13. Temperature measured from cloud web application

The (Fluke 59 MAX/59 MAX+) Infrared Thermometer device in the suggested system was used to calibrate the measured temperatures from the DHT11 temperature sensor. Figure 13 illustrates how closely the results were with a low error rate. Using the percentage error calculator, the error amount is 0.078866%. As shown by the following equation:

$$\text{Percentage Error} = \frac{|\text{Approximate} - \text{Exact}|}{\text{Exact}} \times 100\%$$

5.2 LPG, CO, and smoke detection

Smoke, carbon monoxide, and liquid petroleum gas (LPG) are all hazardous gases; thus, it is critical to recognize them and take preventative measures. As indicated in Figure 14, the ventilation fan will activate automatically when these gases are present in specific amounts. The homeowner's telegram account will receive a warning message, as depicted in Figure 15. Under all circumstances, the fog user interface and GUI will present the readings for each percentage, as shown in

Figures 16 and 17. Figure 18 shows us the GUI's reading rate of gases and smoke.

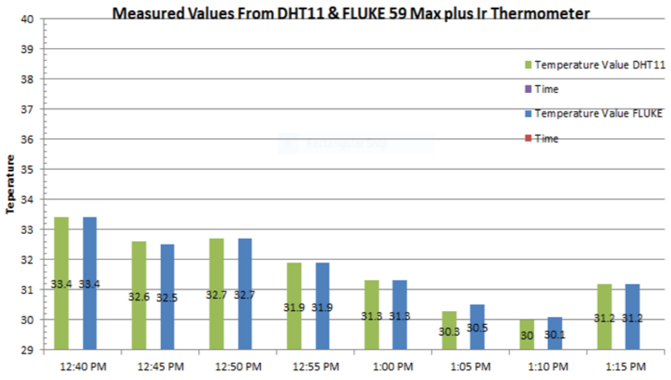


Figure 14. Temperature calibration results

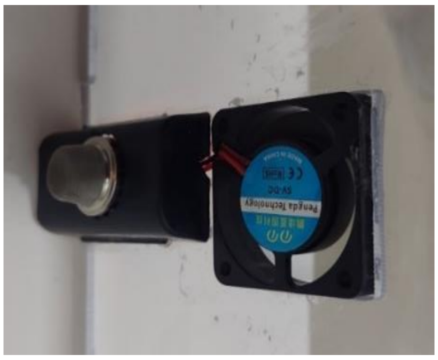
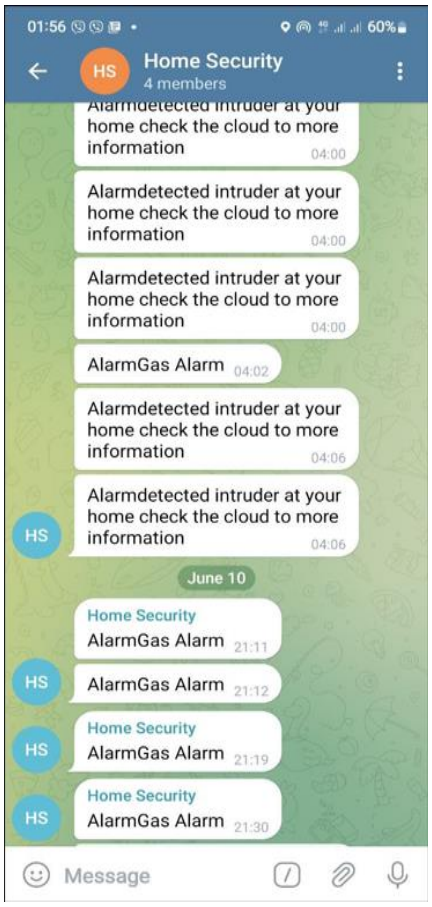


Figure 15. The ventilation fan automatically working



01:56 60%

Home Security
4 members

Alarmdetected intruder at your home check the cloud to more information 04:00

Alarmdetected intruder at your home check the cloud to more information 04:00

Alarmdetected intruder at your home check the cloud to more information 04:00

AlarmGas Alarm 04:02

Alarmdetected intruder at your home check the cloud to more information 04:06

Alarmdetected intruder at your home check the cloud to more information 04:06

June 10

Home Security
AlarmGas Alarm 21:11

AlarmGas Alarm 21:12

Home Security
AlarmGas Alarm 21:19

Home Security
AlarmGas Alarm 21:30

Message

Figure 16. Gas alarm message via telegram application

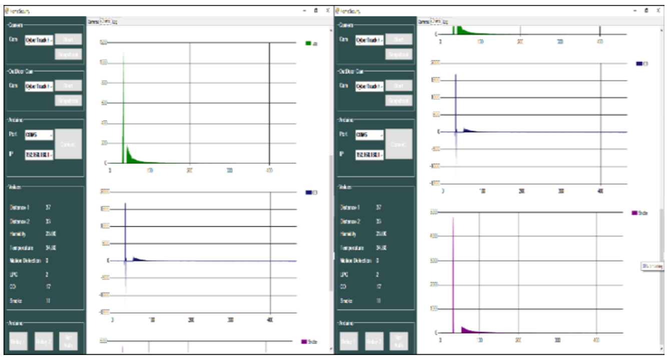
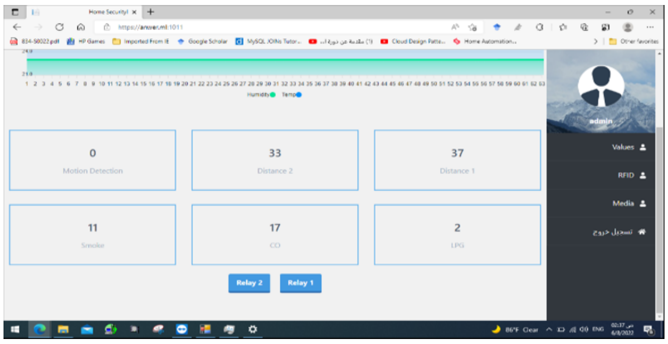


Figure 17. Reading rate of gases and smoke from the fog node



0 Motion Detection
33 Distance 2
37 Distance 1
11 Smoke
17 CO
2 LPG

Figure 18. The reading rate of gases and smoke from the GUI

5.3 Humidity and temperature sensor

The humidity and temperature sensor (DHT11) is a reliable and consistent humidity and temperature sensor that uses digital signal acquisition. It comprises a capacitive or touch humidity sensing element, a negative temperature coefficient (NTC) sensor, and an 8-bit microcontroller. The sensor provides accurate data, a short response time, and is immune to electrical noise. Key features include humidity detection, temperature measurement, digital output, and an 8-bit microcontroller. Applications include HVAC systems, dehumidifiers, weather stations, smart environments, and automotive, aiming to provide a more comfortable interior environment for vehicle passengers.

The DHT11 sensor is a popular and affordable solution for measuring temperature and humidity in various environmental monitoring applications. It uses a single-wire and single-byte method of measuring, making it easy to use and low-cost. The DHT11 sensor module, used since 2007, can be directly interfaced with 8, 16, or 32 microcontrollers depending on the humidity and temperature variant. The sensor uses humidity-sensitive capacitance to produce a signal, taking the response of the capacitive humidity sensor and the reading of the NTC thermistor. The moisture affects a layer of lithographed polyimide on alumina, which functions as a capacitor. The microcontroller measures the time required for the capacitor to be loaded and compares it to the voltage needed for the reference circuit. The DHT11 sensor operates with 2.5-5.5 V digital systems and uses a single-wire protocol, sending a serial bitstream with temperature and humidity. The latest DHT11s have four pins for calibration and operate as usual.

Lastly, DHT11 is a suitable sensor for measuring humidity and temperature functions in different applications based on its high perfervidity and comparatively low price. The resistive type of humidity measurement block, NTC temperature measurement block, and an 8-bit microcontroller help achieve accurate, fast, and interference immune performance. This product would be a worthy addition to any environment control system used for industrial, commercial, or residential purposes.

6. LIMITATIONS

6.1 Limitations of MQTT

MQTT, a cloud computing protocol, has limitations in smart city applications due to its reliance on a reliable network and compatibility with SSL, a licensed protocol. This can affect performance and battery life, especially for low-cost boards. Addressing these drawbacks is crucial for designing large smart cloud solutions in smart cities.

6.2 Limitations of Node MCU

The Node MCU, a microcontroller unit for smart environments, has limitations such as limited processing power, minimal memory, and poor internet connectivity. These limitations can affect computation in memory-intensive tasks and application development. The Node MCU's connection to internet-based cloud systems can cause communication issues or poor protocol conversion. The embedded data transmission speeds can vary due to limited computational capabilities. Security risks from adversaries can also affect performance. Despite these limitations, the Node MCU is a powerful yet user-friendly microcontroller capable of processing large amounts of data but could be more accessible for beginners [35].

6.3 Limitations of smart cloud computing systems in IoT environments

The rise of internet-connected services and applications necessitates the development of lower-level networks, including local, global, edge, and cloud networks. These systems generate and exchange massive amounts of data but have limitations such as limited resources, communication protocols, security threats, and efficient device management. In IoT environments, the critical limitations include resource constraints, security threats, efficient device management, and intelligent local device response [36].

7. SYSTEM PERFORMANCE ANALYSIS

7.1 System implementation

The Arduino IDE is used to program the ESP8266 module as an MQTT client, enabling the integration of sensors into an IoT ecosystem. The ESP8266 is the central node, directly interfacing with sensors that gather environmental data. The data is transmitted to the Cayenne IoT platform via Wi-Fi, enabling remote monitoring and control of various parameters. A DHT11 humidity and temperature sensor measures the environment's air temperature and humidity levels, which are then displayed on a user-friendly dashboard. The Cayenne platform provides data storage capacity and detailed insights, enabling continuous monitoring and historical data analysis. A light-dependent resistor sensor monitors ambient light intensity, enabling lighting control automation. This intelligent lighting control reduces power consumption in residential and industrial settings. The project also incorporates a notification engine for real-time updates. Figure 19 shows the system connection in simple form.

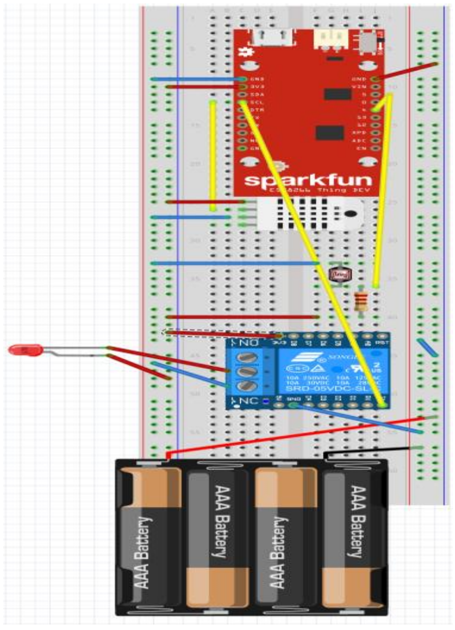


Figure 19. System connection in simple form

7.2 System results

Response times

-The system's response time, from detecting abnormal sensor readings to issuing alerts, averaged around 1-2 seconds in trials conducted under normal Wi-Fi conditions.

-When dealing with critical sensors (e.g., gas leaks), response times were well within acceptable safety thresholds, ensuring rapid alerts to homeowners.

Power consumption

-In idle mode, the Node MCU (ESP8266) consumed 70 mA of current, while during active data transmission, power consumption rose to 120 mA.

-The system's overall energy efficiency was optimized by putting the sensors and communication modules into sleep mode when not actively transmitting.

Data transmission rates

-The broker-to-cloud data transmission rate via Wi-Fi averaged 30-40 KB/s, sufficient for transferring sensor readings at regular intervals without network congestion.

-Data loss was minimal, and the MQTT protocol's lightweight nature ensured reliable data delivery even under moderate network load.

8. CONCLUSION

MQTT excels in environments with low bandwidth and minimal energy consumption as a lightweight protocol. This makes it ideal for applications requiring efficient wireless internet access beyond traditional Wi-Fi networks. One such implementation involves the use of MQTT clients on ESP8266 microcontrollers. By employing ESP8266, a prototype for an intelligent environment system based on MQTT has been developed.

This system allows for the remote monitoring and control of sensors and actuators connected to the ESP8266 via a shared environment gateway. Consequently, existing infrastructure can be leveraged to enhance environmental appliances, making them more innovative and efficient. The result is an intelligent, convenient, and energy-effective environment management system.

In summary, applying MQTT on ESP8266 provides a robust solution for creating energy-efficient and easy-to-manage intelligent systems, contributing to more innovative and sustainable environmental control.

REFERENCES

[1] Bangare, P.S., Patil, K.P. (2023). Internet of things based secured data transmission protocol for agriculture application. Ingénierie des Systèmes d'Information, 28(2): 419-424. https://doi.org/10.18280/isi.280217

[2] Abdul-Rahaim, L.A., Ali, A.M.A. (2015). Remote wireless automation and monitoring of large farms using wireless sensors networks and the internet. International Journal of Computer Science & Engineering Technology (IJCSET): 6(3): 118-137.

[3] Zahra, M.M.A., Mohsin, M.J., Abdul-Rahaim, L.A. (2020). Artificial intelligent smart home automation with secured camera management-based GSM, cloud computing, and Arduino. Periodicals of Engineering and Natural Sciences, 8(4): 2160-2168. https://doi.org/10.21533/pen.v8i4.1718

[4] Al-Gburi, M.K., Abdul-Rahaim, L.A. (2022). Secure smart home automation and monitoring system using the Internet of Things. Indonesian Journal of Electrical Engineering and Computer Science, 28(1): 269-276. https://doi.org/10.11591/ijeecs.v28.i1.pp269-276

[5] Hamzah, A S., Abdul-Rahaim, L.A. (2022). Smart homes automation system using cloud computing based enhancement security. In 2022 5th International Conference on Engineering Technology and its Applications (IICETA), Al-Najaf, Iraq, pp. 164-169. https://doi.org/10.1109/IICETA54559.2022.9888307

[6] Pal, S., Ghosh, S., Bhattacharya, S. (2017). Study and implementation of environment monitoring system based on MQTT. Environmental and Earth Sciences Research Journal, 4(1): 23-28. https://doi.org/10.18280/eesrj.040105

[7] Son, J.Y., Park, J.H., Moon, K.D., Lee, Y.H. (2011). Resource-aware smart home management system by constructing resource relation graph. IEEE Transactions on Consumer Electronics, 57(3): 1112-1119. https://doi.org/10.1109/tce.2011.6018863

[8] Zamora-Izquierdo, M., Santa, J., Gomez-Skarmeta, A. (2010). An integral and networked home automation solution for indoor ambient intelligence. IEEE Pervasive Computing, 9(4): 66-77. https://doi.org/10.1109/mprv.2010.20

[9] Ahmed, S.H., Kim, D. (2016). Named data networking-based smart home. ICT Express, 2(3): 130-134. https://doi.org/10.1016/j.icte.2016.08.007

[10] Jones, T.N., Christian, G.R., Felix, P. (2021). Wireless sensors network for monitoring linear infrastructures using MQTT protocol on Raspberry Pi with nRF24l01 and node-red. Instrumentation Mesure Métrologie, 20(5): 247-254. https://doi.org/10.18280/i2m.200501

[11] Shatti, A.H., Hasson, H.A., Abdul-Rahaim, L.A. (2021). Automation conditions of mobile base station shelter via cloud and IoT computing applications. International Journal of Electrical & Computer Engineering, 11(5): 4550-4557. https://doi.org/10.11591/ijece.v11i5.pp4550-4557

[12] Hamzah, A.S., Abdul-Rahaim, L.A. (2022). Smart home automation system using cloud computing based enhancement security and environment. In 2022 2nd International Conference on Advances in Engineering Science and Technology (AEST), Babil, Iraq, pp. 334-339. https://doi.org/10.1109/AEST55805.2022.10412921

[13] Almhana, H.A., Samir, M.J., Abdul-Rahaim, A.A. (2018). Monitoring and controlling home automation system based on GSM/GPRS Technique. Journal of Engineering and Applied Sciences, 13(18): 7545-7554. https://doi.org/10.3923/jeasci.2018.7545.7554

[14] Abdul Kareem, A.H., Ahmed, B.M., Abdul-Rahaim, L.A. (2022). Design and implementation of humidity and temperature automation and monitoring for public building comfortable climate based on cloud platform. Przeglad Elektrotechniczny, 98(4): 94-100. https://doi.org/10.15199/48.2022.04.21

[15] Kamil, A.A., Nasr, M.S., Alwash, S. (2020). Maximum power point tracking method for a photovoltaic system based on enhanced particle swarm optimization algorithm under partial shading condition. International Journal of Intelligent Engineering and Systems, 13(6): 241-254. https://doi.org/10.22266/ijies2020.1231.22

[16] Hanon, W., Salman, M.A. (2024). Smart controller integrated with MQTT broker based on machina learning techniques. Journal Européen des Systèmes Automatisés, 57(1): 87-94. https://doi.org/10.18280/jesa.570109

[17] Li, L., Xiaoguang, H., Ke, C., Ketai, H. (2011). The applications of wifi-based wireless sensor network in internet of things and smart grid. In 2011 6th IEEE Conference on Industrial Electronics and Applications, Beijing, China, pp. 789-793. https://doi.org/10.1109/iciea.2011.5975693

[18] Mathews, S.P., Gondkar, R.R. (2019). Protocol recommendation for message encryption in MQTT. In 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, pp. 1-5. https://doi.org/10.1109/icondsc.2019.8817043

[19] Kodali, R.K., Soratkal, S. (2016). MQTT based home automation system using ESP8266. In 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Agra, India, pp. 1-5. https://doi.org/10.1109/r10-htc.2016.7906845

[20] Kashyap, M., Sharma, V., Gupta, N. (2018). Taking MQTT and NodeMcu to IoT: Communication in internet of things. Procedia Computer Science, 132, 1611-1618. https://doi.org/10.1016/j.procs.2018.05.126

[21] Gheorghe-Pop, I.D., Kaiser, A., Rennoch, A., Hackel, S. (2020). A performance benchmarking methodology for MQTT broker implementations. In 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, pp. 506-513. https://doi.org/10.1109/qrs-c51114.2020.00090

[22] Li, Y. (2013). Design of a key establishment protocol for smart home energy management system. In 2013 Fifth

International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, pp. 88-93. https://doi.org/10.1109/cicsyn.2013.42

[23] Jabbar, W.A., Kian, T.K., Ramli, R.M., et al. (2019). Design and fabrication of smart home with internet of things enabled automation system. IEEE Access, 7: 144059-144074. https://doi.org/10.1109/access.2019.2942846

[24] Kamil, A.A., Mousa, A.J., Aljuboury, A.S., Abdul-Rahaim, L.A. (2024). Control system design for failure starting of diesel power block for cell on wheels communication tower based on cloud service system. Journal of Internet Services and Information Security, 14(3): 275-292. https://doi.org/10.58346/JISIS.2024.I3.017

[25] Castellanos, W., Macias, J., Pinilla, H., Alvarado, J.D. (2021). Internet of things: A multiprotocol gateway as solution of the interoperability problem. arXiv preprint arXiv:2108.00098. https://doi.org/10.48550/arXiv.2108.00098

[26] Alsahlani, A.Y.F., Popa, A. (2020). Analyzing of LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, pp. 1837-1844. https://doi.org/10.1109/SSCI47803.2020.9308139

[27] Ugwuanyi, S., Irvine, J. (2019). Intelligent internet of things (IoT) node demonstrator for device monitoring and control in the oil and gas sector. arXiv preprint arXiv:1902.10255. https://doi.org/10.48550/arXiv.1902.10255

[28] Kumar, P., Dezfouli, B. (2019). Implementation and analysis of QUIC for MQTT. Computer Networks, 150: 28-45. https://doi.org/10.1016/j.comnet.2018.12.012

[29] Calvo, I., Espin, A., Gil-García, J.M., Fernández Bustamante, P., Barambones, O., Apiñaniz, E. (2022). Scalable IoT architecture for monitoring IEQ conditions in public and private buildings. Energies, 15(6): 2270. https://doi.org/10.3390/en15062270

[30] Di Paolo, E., Bassetti, E., Spognardi, A. (2023). Security assessment of common open source MQTT brokers and clients. arXiv preprint arXiv:2309.03547.

[31] Dinculeană, D., Cheng, X. (2019). Vulnerabilities and limitations of MQTT protocol used between IoT devices. Applied Sciences, 9(5): 848. https://doi.org/10.3390/app9050848

[32] Gentile, A.F., Macrì, D., Carnì, D.L., Greco, E., Lamonaca, F. (2024). A performance analysis of security protocols for distributed measurement systems based on internet of things with constrained hardware and open-source infrastructures. Sensors, 24(9): 2781. https://doi.org/10.3390/s24092781

[33] Wu, C.Y., Huang, K.H. (2020). A framework for off-line operation of smart and traditional devices of IoT services. Sensors, 20(21): 6012. https://doi.org/10.3390/s20216012

[34] Ovalles, F.O., Bolivar, A.E., Rodriguez, A.J. (2018). Use of an embedded system with WIFI technology for domotic control of conventional environments. Journal of Physics: Conference Series, 1126(1): 012070. https://doi.org/10.1088/1742-6596/1126/1/012070

[35] Rahmadiansah, A., Utama, S.W., Wirayusa. (2019). Study of low-cost IoT application for industry 4.0 using NodeMCU ESP8266 module. AIP Conference Proceedings, 2088(1): 020036. https://doi.org/10.1063/1.5095288

[36] Upadhyay, S., Kumar, M., Upadhyay, A., Verma, S., Kavita, Kaur, M., Khurma, R.A., Castillo, P.A. (2023). Challenges and limitation analysis of an IoT-dependent system for deployment in smart healthcare using communication standards features. Sensors, 23(11): 5155. https://doi.org/10.3390/s23115155