

Design and implementation of IOT driven environment monitoring , alerting and controlling mechanism for industries over secure MQTT protocol

Nitin K Gunari¹, Dr. Prashant V Joshi²

¹School of ECE, REVA University, Bengaluru, India

²Assistant Professor, School of ECE, REVA University, Bengaluru, India.

Abstract— *The industrial environment data is used for monitoring various industrial plants to maintain stable environment conditions for various operations. We need to have a monitoring system which can be moved to mobile or a desktop pc/laptop so that the plants can be monitored remotely. This paper shows the design, development and implementation of environment monitoring and controlling, alerting mechanisms for industries over secure MQTT protocol driven by IOT. Embedded system provides features like flexibility, easy installation and reasonable cost. This system uses Arduino UNO microcontroller which handles multiple processes based on multi-tasking and is coded with “C” and “C ++” language. The circuit consists of sensors and Wi-Fi module connected to microcontrollers. Industrial real time environment data like temperature, humidity, light, smoke, Intruder/obstacle detection is collected by sensors. This data is converted to desirable format and is sent to local MQTT server .The communication between MQTT server and client is secured with Secure Socket Layer(SSL)/Transport Level Security(TLS).By using Internet Protocol(IP) address , data is monitored using web browser from anywhere around the world using laptop or mobile. Appropriate actions are triggered as safety measures when environment parameters exceed desired values.*

Keywords—*ArduinoUNO, embedded system, industrial environment data acquisition, Internet of Things, , sensor interface.*

1. Introduction

As the advancement in the technology is happening industries are becoming more complex and are being used in harsh environment. Because of the complexity of the system and environmental conditions, human intervention to know the environment status in real time is difficult.

The resources, components and various operations in industry are to be kept and operated in particular environment and this environment data needs to be acquired in real time for monitoring. The parameters like temperature, smoke, light, obstacle detection, humidity are most important factors for the resources to deal with. Most resources work better when they are kept and operated in a suitable environment conditions. A system is designed and developed that collects this information from various resources in real time and can be monitored remotely from any place so that work is carried out efficiently and good quality products are produced. Appropriate actions are taken if environment parameters value exceed specified limit.

Microcontrollers have high throughput and multitasking capabilities and are used in embedded system applications. Based on the type of application being developed appropriate microcontrollers are used. More preferred is Arduino UNO. Arduino UNO microcontrollers are low power, low cost, have high efficiency, easily available and programmed and can be used for real time application purposes.

MQTT protocol is used for message communication between devices. This protocol is light weight, highly efficient and mainly used during resource and bandwidth constraints.

Secure MQTT solution using SSL/TLS is used in order to provide secure communication between server and clients so that data passed remains private and integral and provide authentication mechanism.

Based on IoT, the resources such as sensors, microcontroller and remote devices are configured and provided with wireless connectivity. The system makes use of phone or laptop as a remote device to view the status of system in real time and thus monitor the resources.

IoT is a very emerging and rapidly developing field which has made the way to connect various physical objects to connect with digital world. IoT has marked itself in innovation like smart automation in various fields like home, agriculture, medical, food industries and many others.

2. Problem definition and formulation

2.1 Problem definition

Most industrial resources or operations work better when they are kept and operated in a particular or specified environment conditions. A system is needed that collects industrial environment data in real time and this real time data need to be in easy and readable format and should be available on our devices to quickly monitor and analyze from anywhere in the world.

If environment parameters exceed desired levels, immediate safety measures need to be automated to avoid disasters.

Encrypted communication channel is required between devices, so that no attacker can alter, eavesdrop any part of the communication and only authenticated devices communicate.

2.2 Problem formulation

Scope and objectives:

- a. Retrieve environmental data of different industrial resources in real time using embedded system and IOT.
- b. Monitor through a standalone system, Laptop or remotely through a mobile device or laptop.
- c. Automate appropriate safety measures when environmental parameters exceed desired levels using devices like fan, bulb, and buzzer.
- d. Provide a message communication protocol that is light weight, easy to implement and efficient like MQTT.
- e. Provide data encryption, data integrity and authentication during communication like SSL.

3. Design tools used

3.1 Hardware Requirements

1. Arduino UNO microcontroller.
2. Temperature Sensor –LM35
3. Humidity Sensor- NSK 85 180914

4. Smoke Sensor-MQ135
5. Light Sensor: Light Dependent Resistor(LDR)
6. Infrared sensor (IR) sensor.
7. Wi-Fi Module-ESP 8266 for transmitting data to server.
8. USB cable for burning the code to microcontroller.
9. Laptop, mobile or standalone system to monitor the system resources.
10. Relay, Buzzer, bulb, cooling fan: These are controlling devices which are triggered to control the environment parameters during unfavorable environment conditions.

Relay: Controls bulb and fan by switching ON or OFF.

Buzzer: 5V Buzzer

Bulb: 8 W / 12 V

Cooling Fan: 12V, 3" DC Cooling Fan

3.2 Software Requirements

1. Arduino IDE: Development environment (IDE) for C/C++ programming for microcontrollers used to upload the code to board.
2. Flask: micro web framework of Python used in web application programming.
3. Java Script and Python: Used For server side scripting.
4. Adafruit IO server: Adafruit IO is a server that provides data connections. It requires very little programming. Client libraries include REST and MQTT API's which makes clients communicate with the server easily.
 - a. As mentioned above Adafruit IO client libraries provide support for MQTT as well as secure MQTT and can be used in client side like in Arduino platform by including the library in Arduino IDE.
 - b. Special kind of topics are provided by MQTT API to which sensor data or feed is published and the same topic is used by subscribers. Below is the example.
 Topic: (username)/feeds/(feed name or key)
 Here username: is the username created in io.adafruit.com. This acts as a security feature by restricting access to feeds / data that is published.
 and feedname or key : is the name of the feed or key
 - c. We are using Adafruit MQTT client library for MQTT protocol implementation. This library works on Arduino. We can download the library and use it in Arduino. Below header files are used from library. Adafruit_MQTT.h , Adafruit_MQTT_Client.h
 - d. Below are the details required to connect MQTT client or device to Adafruit IO server.
 Host: The server required to be connected i.e. io.adafruit.com
 Port: 1883 for plain MQTT, 8883 for secure MQTT. We are using 8883 for secure MQTT connection which has SSL implemented.
 Username: Created under io.adafruit.com
 Password/Key: Is the key of Adafruit IO .It is mainly Client ID.

e. In order to connect to secure MQTT SSL /TLS support, we are using Wifi secure client class where we invoke a method which is as below

Verify (fingerprint, host).Here fingerprint and host are the parameters.

Fingerprint: It is a 160 bit string which is in hexadecimal that identifies the certificate. It is a unique value to a certificate. Fingerprint is computed from the data of the certificate using an algorithm like SHA 1.

Host: is the server to which client wants to connect.

4. Design and implementation

4.1 Block Diagram

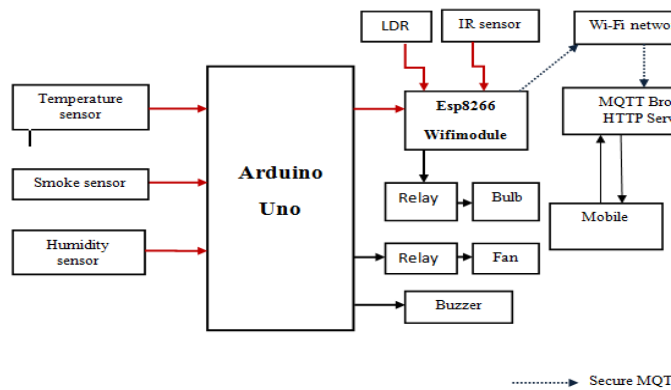


Fig.1.System Block diagram

System is divided into five modules as below:

1. Data acquisition
2. Data conversion
3. Data transmission
4. MQTT Broker data acquisition and transmission Data monitoring
5. Data Security
6. Data Display
7. Controlling mechanism

Data acquisition: The Industrial real time data is captured by the sensors. Sensors connected to Arduino Uno /and Wifimodule.

Data conversion: The data in analog format, it's converted into digital data by microcontroller.

Data transmission: The digital data is transmitted to MQTT broker Wi-Fi Network through Wi-Fi module.

MQTT Broker data acquisition and transmission: Local MQTT Broker and Web server is set up. Messages received from Microcontrollers are sent to subscribed clients i.e. Laptop or mobile devices by MQTT Broker. Provides a URI by using which the concerned users can view the data and monitor.

Data Security – SSL security is implemented between clients and Broker communication in order achieve data integrity and device authentication.

Data Display: The Data from web server which is streamed is viewed using the URI with the help of UI client like web browser from Computer or mobile.

Controlling mechanism: Appropriate controlling devices are triggered when environment parameters exceed desired limits.

4.2 Working Principle

Embedded systems have different designs as per their applications. Modular design approach is followed for this project. The overall system is composed of a microcontroller, sensors mainly temperature, humidity, smoke, LDR, IR, Wi-Fi module, MQTT broker, end user monitoring devices like mobile or laptop. The circuit consisting of sensors, microcontroller, Wi-Fi module is placed in the environment which needs to be monitored. The microcontroller acts like the control unit. Microcontroller is programmed with C/C++ using which the microcontroller to interact with sensors and Wi-Fi module. Micro controller processes the sensor data and sends it to Wi-Fi module.

Any change in temperature and humidity in the environment is detected by temperature sensor and humidity sensor respectively, whereas smoke sensor detects gases emitting from decaying food products or any smoke from fire. Also LDR is used in locations where light intensity needs to be monitored. Any obstacle or intruder is detected by IR sensor. The output voltage of sensors changes with changes in environment and output voltages are sent to ADC unit of the microcontroller. The microcontroller processes these voltages using the program. The output of the microcontroller is sent to MQTT broker by Wi-Fi module. The parameters are viewed using URI of the server in mobile or laptop.

Below controlling devices are triggered automatically when environment parameters exceed desirable limits.

If temperature exceeds 32 degrees, then fan is ON.

If light intensity decreases, bulb is made ON.

If smoke or intruder or fire is detected then Buzzer is made ON.

Here microcontroller, laptop or mobile devices acts as clients and MQTT broker installed in local machine acts as server.

Entire communication between sending client and MQTT server takes place through secure MQTT protocol.

MQTT protocol: MQTT is IoT messaging connectivity protocol. It is a publish/subscribe protocol which works on top of TCP.

This protocol supports small devices, and data is transmitted to very far distances, sometimes intermittent networks. This protocol is used for connected devices where there are bandwidth constraints and the devices have constrained resources.

In MQTT, messages are sent through publish subscribe architecture asynchronously.

MQTT topology consists of MQTT server and client. MQTT client and server communicate through different control packets like Publish, Connect. Data transmitted over network through MQTT packet takes less bandwidth.

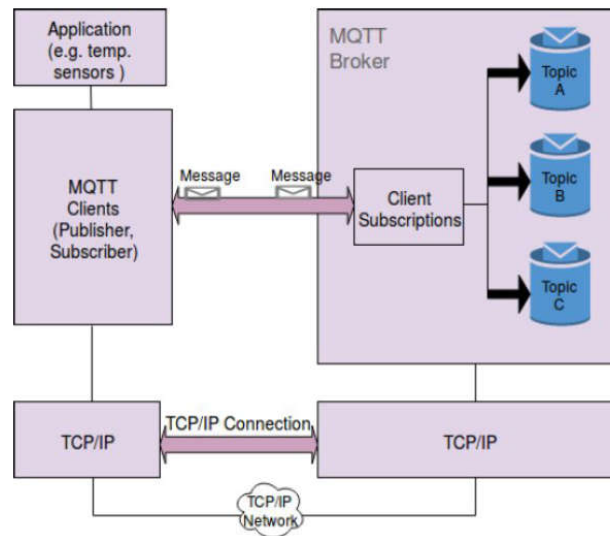


Fig 2: MQTT protocol architecture

Figure 2 shows the architecture of MQTT protocol.

MQTT architecture has 3 components –

1. Publisher
2. MQTT broker
3. Subscriber

Publisher– Is a client that connects to broker and publishes messages like data from sensors to MQTT broker.

Message contains actual data and topic associated with data.

Topic is a string which is used by broker to filter messages of each client. Topics may contain levels, one or more .Topics are very light weight because it doesn't require initializing a topic before publishing or subscribing to it.

In this project micro controllers acts as a Publisher.

MQTT Broker-Is the main part of MQTT protocol. It receives messages from publishing clients, filters them based on topics and decides who is interested in it and then sends messages to subscribed clients. From figure 2, we can see that there are 3 messages with topics A, B and C.

If a client is interested in Topic A, then it subscribes to it. If another client subscribes to all 3 topics, then are 3 messages are sent to that client. Authentication and authorization of clients is also done by broker. It can have one or more clients. Can be locally installed server or cloud based server. In this project HBMQTT is used as broker and local server is used.

Subscriber- Is a client that receives subscribed message from broker. In this project mobile devices and laptops subscribe to messages.

Entire communication happens over TCP connection.

Secure MQTT: SSL provides encrypted channel between server and client. SSL is the cryptographic algorithm which uses handshake mechanism to create secure communication channel. In handshake mechanism various parameters are negotiated like using certificates to authenticate server.

Plain TCP which is default connection doesn't use encrypted secure communication. Username and password passed from client to server for authentication can be easily stolen by attackers if plain MQTT is used. MQTT brokers can use SSL in place of plain TCP. For secure MQTT connection port MQTT 8883 is used. By using private key, public key, server certificates of Adafruit IO server SSL connection is established between client and server. The certificates and keys are used for authentication and encryption purpose.

SSL handshake: fig 3 shows SSL handshake. When client connects to 8883 port of the server, server sends its server certificate and public key to client. Server side certificate is signed by Certifying Authority, saying that the keys belongs to the server itself. Since client already has fingerprint of server certificate, it verifies the fingerprint against finger print of server certificate. Once both match, authentication process is completed. Client sends secret key information encrypted with public key of the server and at the server side private key is used to decrypt the secret key information. Messages that are exchanged between client and server use this secret key to encrypt messages.

In this way secure communication is created between client and server.

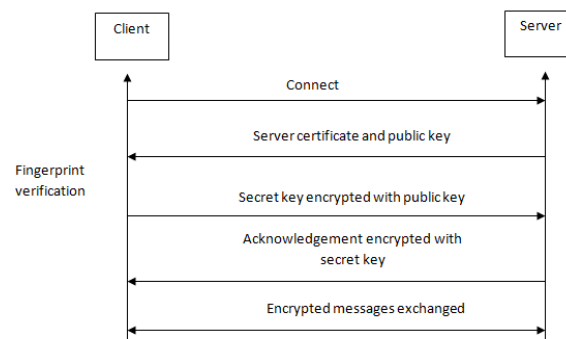


Fig 3: SSL handshake

5. Advantages, disadvantages and applications

5.1 Advantages

1. Real time data acquisition.
2. Industrial environment data can be monitored from anywhere around the world.
3. Automatic environment controlling and alerting mechanism.
4. MQTT is light weight messaging protocol, very much suitable for low bandwidth and fragile networks.
5. Secure communication channel with efficient messaging protocol.
6. Low cost

5.2 Disadvantages

1. Data acquired is not stored
2. The circuit is wired.
3. Internet facility is required
4. SSL/TLS resource consumption is higher compared to normal TCP.
5. CPU and bandwidth consumption is more in the process of SSL
6. RAM consumption is also high, since TLS/SSL needs additional buffers for every MQTT connection.
7. Message Expiry: In MQTT, there is no message expiry, so if message is sent to the broker and no one picks it then broker is overloaded with messages. This degrades overall performance.

5.3 Examples of Real time applications of MQTT / Secure MQTT

1. Facebook has used MQTT aspects for online chat.
2. Microsoft Azure IoT uses MQTT protocol for transmitting messages from measurable instrument devices
3. Amazon Web Services is using Amazon IOT on MQTT / Secure MQTT.
4. Trace Lora MQTT, is a rail application uses secure MQTT for transmitting messages.

5.4 Major Applications of this IoT based concept

1. Monitoring water purity, internal and external water temperature, CO2 concentration and light intensity on the surface of water in real time.
2. In healthcare domain objects like people, equipment, and medicine can be tracked and monitored in real time.
3. In mining industry IoT technologies are used to identify mine disaster signals and receive early warning thus improving safety of underground work.
4. IoT can be used in logistics companies for monitoring of objects in real-time from source to destination across entire supply chain which includes manufacturing, shipping, distribution etc.
5. Used in the fire fighting safety field to detect fire and raise warning to prevent fire disasters.

6. Results

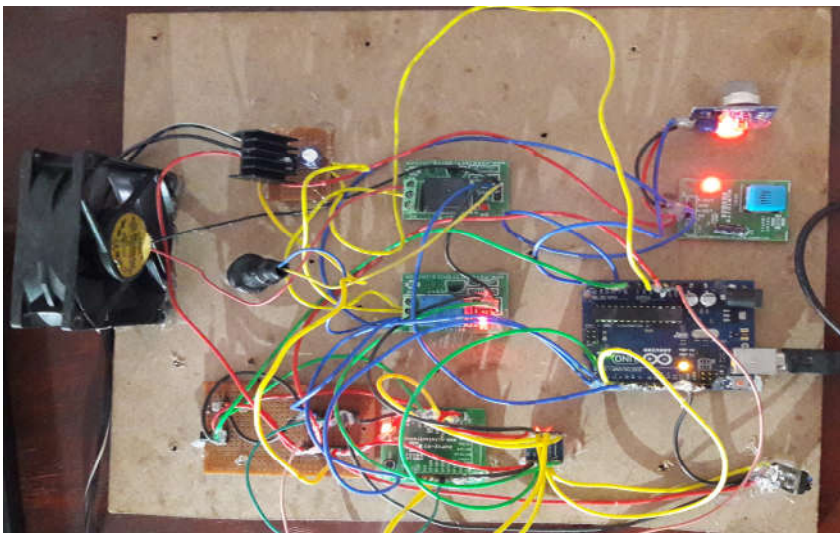
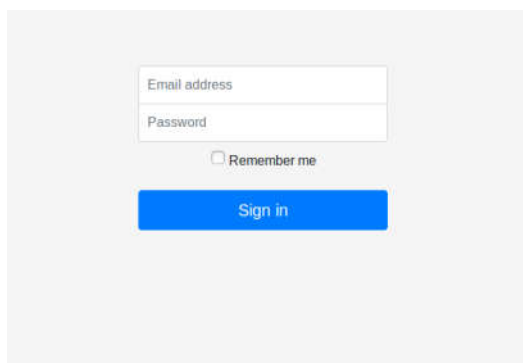


Fig.4.The circuit consisting of sensors, microcontroller, Wi-Fi module, controlling and alerting devices



A login form with two input fields: "Email address" and "Password". Below the fields is a checkbox labeled "Remember me". At the bottom is a blue button labeled "Sign in".

Fig.5.UI webpage showing login credentials



Secure MQTT

temp	hum	smoke	IR	LDR
32.3	Normal	detected	No obstacle	Normal Light

Fig.6.UI webpage showing output of sensor values

7. Conclusion

Environment data acquisition, monitoring and controlling in real time is very important for any industry to function properly and produce good quality products and increase its profits. Also communication protocol and security forms an important aspect in IOT model .This project has implemented IOT driven environment monitoring, alerting and controlling mechanism for industries over secure MQTT protocol.

In this project prototype, sensor data is collected in real time and is transmitted over secure channel to the server and remote devices. Automatic actions can be triggered based on sensor data values. This design can be implemented in other IOT projects involving critical applications like in medicine, military, satellite domains respectively where data sent is very crucial .

8. Acknowledgment

I would like to thank my guide and project coordinator Dr. Prashant V Joshi , Assistant Professor, school of ECE, REVA University for his valuable guidance for completion of the project. Finally I wish to thank and acknowledge the help given by my parents and friends which helped for the completion of the project successfully.

References

- [1] Qingping Chi, Hairong Yan, Chuan Zhang, Zhibo Pang, and Li Da Xu, Senior Member, IEEE, “A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment.”,IEEE Transactions on Industrial Informatics,Vol No 10 ,Issue No 2 , May 2014
- [2] Li Da Xu,Senior Member, IEEE, Wu He, and Shancang Li, “Internet of Things in Industries: A Survey”,IEEE Transactions on Industrial Informatics,Vol No 10 ,Issue No 4 , November 2014.
- [3] Kortuem, G, Kawsar, F, Sundramoorthy, V and Fitton, D “Smart objects as building blocks for the internet of things”,IEEE Computer Society,Vol No 10 ,Issue No 4 , November 2009
- [4] Harish Ramamurthy, B. S. Prabhu and Rajit Gadh, “Wireless Industrial Monitoring and Control using a Smart Sensor Platform”,IEEE Sensors Journal , Vol No 4 ,Issue No 1 , April 2007
- [5] Ioan Ungurean, Nicoleta-Cristina Gaitan and Vasile Gheorghita Gaitan , “An IoT architecture for things from industrial environment” ,IEEE International Conference on Communications , May 2014.
- [6] Zhuming Bi, Senior Member, IEEE, Li Da Xu, Senior Member, IEEE, and Chengen Wang, Senior Member, IEEE, “Internet of Things for Enterprise Systems of Modern Manufacturing”,IEEE Transactions on Industrial Informatics,Vol No 10 ,Issue No 2 , May 2014.
- [7] Dipa Soni , Ashwin Makwana , “A survey on MQTT: A protocol of Internet of Things(IOT)”,ResearchGate , April 2017 .
- [8] Radhika Munoli , Prof. Sankar Dasiga,” Secure Data Transmission for Iot Applications ”, IJARCCCE , Vol. 5, Issue 7, July 2016.
- [9] <https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl>
- [10] <https://docs.aws.amazon.com/iot/latest/developerguide/protocols.html>
- [11] <https://www.kontron.com/downloads/datasheets/lora-mqtt-rail-datasheet-11-2017.pdf?product=148064>
- [12] <https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs>
- [13] https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_7.1.0/com.ibm.mq.doc/sy10660_.htm
- [14] <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-io.pdf>