SUGISAKA MASANORI

Journal of Robotics, Networking and Artificial Life
Vol. 9(2); September (2022), pp. 202–208
ISSN (Online):2352-6386; ISSN (Print):2405-9021
https://alife-robotics.org/jrnal.html

Research Article

Web-based IoT and Robot SCADA using MQTT protocol

Jr-Hung Guo, Tzu-Yuan Lin, Kuo-Hsien Hsia
Department of Electrical Engineering, National Yunlin University of Science & Technology, 123 University Road, Section 3,Douliou, Yunlin 64002 , Taiwan, R.O.C

ARTICLE INFO

Article History
Received 27 October 2021
Accepted 02 August 2022

Keywords
Internet of things (IoT)
Robot
MQ telemetry transport (MQTT)
AES
Supervisory control and data acquisition (SCADA)

ABSTRACT

Internet of Thing (IoT) and robot technology is a very popular research topic. Especially in the application of Industry 4.0, it is the most basic and important part. However, related security and application system development often have many problems. For example, the integration of various communication interfaces and data formats, and the rapid establishment of application systems. Therefore, this paper uses MQTT protocol and AES encryption technology to develop a Web-based SCADA system. This Web-based SCADA uses drag-and-drop operation, and users can quickly build a WYSIWYG (What You See Is What You Get) application system. Moreover, with the characteristics of MQTT message transmission, users can choose to transmit the device's message as it is or to process the data before transmitting when creating the SCADA system. And these messages will be encrypted by AES, making the whole system safer and more efficient. Finally, by collecting different communication protocols and integrating multiple communication interfaces, this SCADA can be connected to many PLCs and other equipment, so that the industry can build application systems quickly and at low cost..

1. Introduction

Internet of Thing (IOT) and robot technology is a very popular research topic recently, especially in the application of Industry 4.0, it is the most basic and most important part. There are many production equipment may be designed without taking into account the function of automatic data collection. Or the equipment has a data collection function, but the equipment manufacturer is unwilling to disclose the communication protocol or data format. The same robotic system also suffers from the above problems. All these have caused the industry to be hindered in the progress of Industry 4.0. Therefore, many industries or government agencies have put forward related plans one after another, hoping to quickly solve the problem of equipment networking communication protocols and data format standards. For example, the international automation company Rockwell also provided a complete set of software and hardware solutions in 2019 [1]. The internationally renowned IBM also released the open source visual interface development tool Node-RED [2]. Provides various APIs, including Internet services, and calls using various communication protocols, such as MQTT, MODBUS, OPC-UA, etc. On embedded systems, Node-RED provides the function of controlling GPIO, and uses MQTT or HTTP protocols to communicate with the cloud to build an Internet of Things system.

After solving the communication protocol, how to build this information into an application system is another complicated problem. In the past, it was usually necessary to write code, set up the host, manage the database system, etc., which usually required a lot of manpower and expense. Therefore, how to quickly build an application system with a lower technical threshold

Corresponding author's E-mail: jrhung@yuntech.edu.tw, m10812021@yuntech.edu.tw, khhsia@yuntech.edu.tw

and cost has become a research topic for many scholars. For example, NDUKWE, Cherechi [3]; etc., which use LoRa to develop a low-cost SCADA monitoring system for small-scale renewable energy. SINGH [4] and others have built an energy-saving smart home system based on IoT. JAYASHREE [5] and others introduced in detail the relevant knowledge of the enterprise to build the Internet of Things system. From the above examples, we can know that the IoT is an important link in Industry 4.0 and many automation applications. But for the monitoring system of the robot, there are very few related researches. The main reason is that each robot system is very different, so at present almost all the monitoring systems are provided by the original manufacturer. However, this will also cause great problems in the follow-up Industry 4.0 system integration. Therefore, this paper integrates the AGV (Automated Guided Vehicle) path planning system commonly used in the industry into the SCADA system of this research. And through the open API, different AGV robot systems can be integrated.

However, due to the inability to integrate communication protocols and data formats, and the complex construction of visual monitoring systems, Industry 4.0 and even the application of artificial intelligence will be greatly hindered. Therefore, this paper is expected to integrate IoT modules, and through common and open software and hardware, various applications can be quickly built. In addition, the web version of the SCADA system is also used to allow users to quickly build the system. Moreover, the IoT module and SCADA system of this study will also integrate artificial intelligence algorithms, as well as various database connections, various industrial communication standards and other protocols. After users build their own systems, they can also extend the connection with ERP, MES, APS and other systems. So that the application of Industry 4.0 and artificial intelligence can be implemented quickly and at low cost to the industry.

2. System Architecture

In the application of the IoTs, because the communication interface and data format of each IoTs device are different, it is very troublesome to collect these data, and to effectively apply and store these data. Therefore, this paper selects the MQTT protocol as the entire Internet of Things data structure. When the data transmission content of the IoTs module can clearly know the corresponding I/O or sensor data, we can clearly distinguish these data, and can effectively publish and apply. If the data transmission content of the IoT module cannot clearly distinguish the corresponding I/O or sensor, the data transmitted by the IoT device can still be completely transmitted. And use artificial intelligence algorithms to parse out the corresponding I/O or sensor data.

MQTT is a lightweight protocol and a protocol designed for the Internet of Things. Therefore, the network bandwidth it requires is very low, and the hardware resources required are also low. It is very suitable for IoT environments with low power consumption and limited network bandwidth, such as smart home appliances or medical devices. MQTT uses the Publish/Subscribe mechanism to transmit data, which contains 4 main elements, Publisher, Subscriber, Topic, and Broker. Among them, the Internet of Things module is the Publisher, and the information that the Internet of Things module sends out is the Topic. These information Topic are not sent directly to the demand-side Subscriber, but are sent through the forwarding station Broker. Therefore, the Publisher and the Subscriber are not directly connected. Therefore, the information and communication security of the entire Internet of Things system must be controlled through a Broker. At present, most brokers have SSL (Secure Sockets Layer) encryption mechanism when transmitting data. But SSL is too complicated for IoT devices. Therefore, this paper uses AES (Advanced Encryption Standard) technology to ensure the security of all communication and data in the communication of the IoT module and the communication of the relay station (Broker).

In this architecture, we use the WebSocket [6] network transmission protocol, which is often used in Internet applications. WebSocket makes the data exchange between the client application or browser and the server easier, especially allowing the server to actively push data to the client. In the WebSocket API, the browser and the server only need to complete a one-time handshake operation, and a sustainable connection can be established between the two, and two-way data transmission can be carried out. WebSocket and HTTP are different communication protocols. Although both belong to the application layer of the OSI model, the transport layer is also a TCP protocol. But WebSocket and HTTP are not the same communication protocol. HTTP uses the "request and response" mode for communication, while WebSocket uses two-way communication. But in order to make it compatible with the HTTP protocol, WebSocket works through HTTP ports 80 and 443, and supports HTTP proxy and middleware. The WebSocket protocol supports the interaction between a Web browser or other client applications and a Web server, and has better communication efficiency. In order to facilitate real-time data transmission between the client and the server. The server can be implemented in a standardized way without the client requesting first. And to allow messages to pass back and forth to each other while remaining connected. In this way, a two-way continuous conversation can be carried out between the client and the server. Communication can be done through TCP port 80 or 443. This is beneficial in an environment where a firewall prevents non-Web network connections, with better flexibility, higher security, and efficiency. In the robot and AGV part, since there is not necessarily a network interface, the communication interface and use AES encryption function can be converted through the IoT module of this study. If the robot and AGV originally have a network communication interface and an AES encryption mechanism, they can be directly connected to the system of this study. The entire communication architecture is shown in Fig. 1.

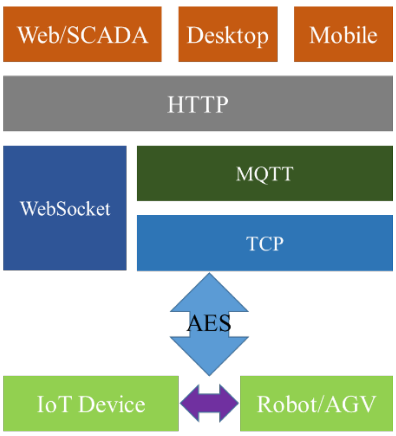


Fig. 1. Block diagram of the communication architecture between IoT devices and robot and application systems.

2.1. *MQTT data structure*

MQTT allows devices and devices to exchange data with each other through topics, so theme design is very important. Since this system will be connected to many controllers, sensors, and actuators, in the design principle of Topic, consider the use of a four-level design, the format is shown in Table 1.

Table 1. MQTT communication data structure.

MQTT serial number	device name	Input and output	Data type
0x00000000 to 0xFFFFFFFF	device name, max 64 character.	Input or Output	sensors, connection status, video streaming, etc.

The description of this structure is as follows.

- MQTT serial number layer: The first layer is named after the device ID, called MQTT ID, starting from 01.
- Device name layer: Since there are many IoTs devices, robot or data sources of other devices in the system, this layer can be used to distinguish different devices or data sources. For example, this paper uses the IoTs module to include An STM32 terminal module and an ESP32CAM can be named stm32 and esp32 respectively, as shown in Fig. 2.
- Input and output layer: This layer can be used to represent data output or input, and are named after control and data respectively.
- Data type: The last layer refers to the format of the subscription data, including sensors, connection status, video streaming, etc., which are respectively sensor, connection, and stream.

The complete MQTT message of this IoT and robot example is shown in Table 2, 3, 4.

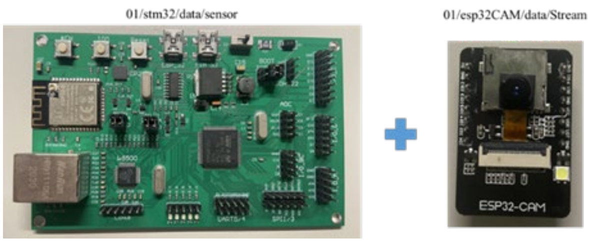


Fig. 2. Examples of IoT modules (STM32 control board and ESP32 imaging device).

Table 2. MQTT comm. data structure.

Topic Name	content
01/stm32/data/sensor	6-channel ADC, temperature and humidity and I/O status on the STM32 module.
01/stm32/data/connection	Wi-Fi and LAN connection information, such as Wi-Fi SSID and password.
01/esp32CAM/data/stream	ESP32CAM image
01/robot/data/connection	Robot or AGV comm. Interface.
01/robot/data/type	Robot or AGV comm. data type, like command, sensor, location or others.

Table 3. MQTT comm. data structure: control for IoT.

Topic Name	Payload	Function
01/stm32/control	setpagedata	Turn on sending Wi-Fi data.
01/stm32/control	statusON	Turn on sending sensor data.
01/stm32/control	statusOFF	Turn off sending all data.
01/stm32/control	reset_io	Reset all I/O.
01/stm32/control	OUTPUT1_ON~ OUTPUT5_ON	Set I/O to high level.
01/stm32/control	OUTPUT1_OFF~ OUTPUT5_OFF	Set I/O to low level

Table 4. MQTT communication data structure _control for robot or AGV.

Topic Name	Payload	Function
01/robot/raw data	Robot comm. raw data	Robot comm. raw data.
01/robot/connect	Robot connect	Robot connect.
01/robot/control	Robot control command	Robot control.
01/robot/status	Robot status	Robot status like work, error or others.
01/robot/locate	Robot path or axis data	Set or return robot location or axis data.
01/robot/stream	Robot stream data	Set or return robot sensor or stream data.

In the communication part of the robot or AGV, because each system is too different, we provide the function of retaining the original robot communication data in the communication structure of MQTT. This allows many different robots and AGV systems to be quickly connected to the system of this paper. Finally, we design the common functional data architecture of robots and AGVs, so that the functions of the robot can be quickly integrated into the SCADA system of this paper, such as control commands, positions, paths, etc. The complete robot and AGV, MQTT communication architecture is shown in Table 4.

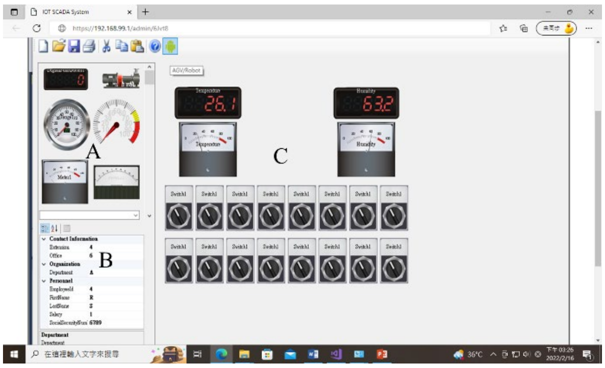
In the design of the SCADA system, we use Web-base and drag-and-drop operation. And in the design of the screen components, try to adopt the design close to the actual object, and through the parameter setting method, to make the operation simple. After the user's screen is designed, what is presented is the final used screen. Fig. 3(a) is the screen of SCADA in the design mode of this thesis. The screen is mainly divided into three blocks, namely:

(A) Instrument icon area: This area provides icons of common industrial instruments and equipment provided by the system.

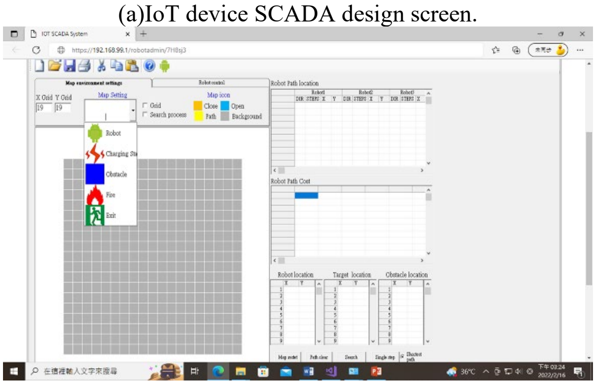
(B) Parameter setting area: This area is for setting the communication parameters, display parameters, MQTT related settings, etc. of the instrument icon.

(C) SCADA design area: Using the drag-and-drop method, place the icons to be displayed and controlled in this area, and you can quickly build the WYSIWYG monitoring system.

In the SCADA part of AGVs and robots, this paper is currently dominated by mobile robots. Users can arbitrarily set robots, obstacles, target points, etc. in the system, and can calculate the path of each robot, or use it to monitor the state of the robot. The whole picture is shown in Fig. 3(b).



(a)IoT device SCADA design screen.



(b)AGV/Robot SCADA screen.

Fig. 3. SCADA design mode screen.

3. Experimental Results

This paper has developed a Web-based SCADA system that allows users to quickly establish a SCADA monitoring system by drag-and-drop operation and parameter setting. We actually apply this system to industrial production lines. Fig. 4 shows that the original production equipment does not have the function of network communication, and needs to monitor the temperature and humidity of the environment. Therefore, we use a self-developed IoT module to collect the signal with an external sensor, and then send it back to the large-scale LED signage and SCADA system on site. Fig. 5 shows the results of the SCADA implementation developed in this paper. Considering that the entire screen information will not be too confusing, the data displayed in the center of the screen can be selected by the user. Or when there is a problem with the field device, the data of the problematic device will be displayed automatically.

In the AGV monitoring section, Fig. 6 is the AGV robot used in this study. This AGV can actively or passively plan the path, and transmit the relevant data and status through the network. Therefore, we integrated this AGV into the SCADA of this study, and the whole operation picture is shown in Fig. 7. In the mobile robot SCADA part of this research, we use the grid display method.
The main reason is to consider that the entire SCADA will not use too many resources when operating on the web page. If the 3D virtual reality is used as the interface of the web page, it will consume more resources. And if many AGVs are monitored at the same time, there will be a problem of how to display the status of different AGVs.

Fig. 4. SCADA system application environment and IoT installation status.

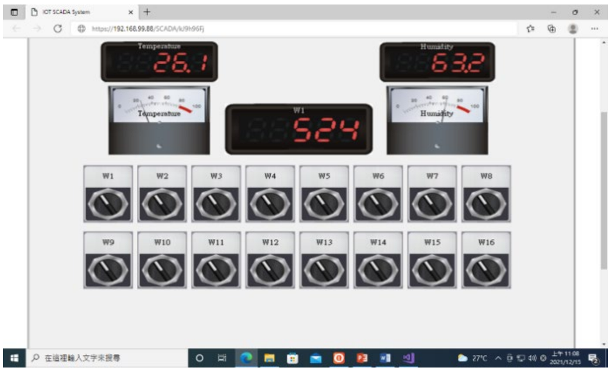


Fig. 5. SCADA screen.

Fig. 6. AGV System.

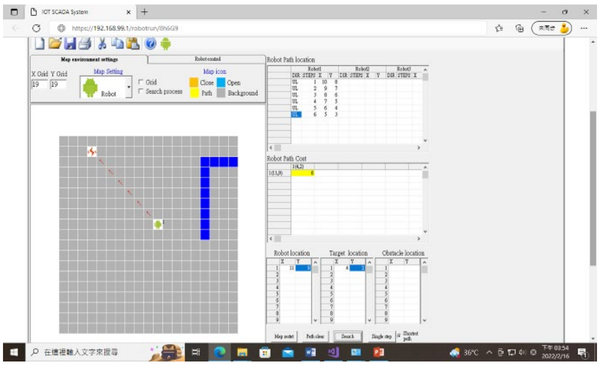


Fig. 7. AGV SCADA screen.

4. Conclusions

This thesis uses the MQTT protocol to develop a Web-base SCADA system, which has built many instruments and devices commonly used in industry. And it allows users to quickly build a SCADA system by drag-and-drop and parameter setting. In the connection with IoT devices, we support multiple communication protocols. If the device data can be clearly distinguished and distinguished, these data will be processed into MQTT messages and used on the SCADA system we designed. If the content of the device's data cannot be clearly known, the SCADA system will still store it in the database and try to parse the content. Finally, we actually applied this SCADA system to the production line, and the overall operation was in good condition. In addition, we also integrated the monitoring of AGV mobile robots. And through the open MQTT protocol, different AGVs or mobile robots can be integrated into the SCADA system of this study. The feasibility of the SCADA system in this study can be verified through the actual test.

In the future, we will integrate the robotic operating system (ROS). Because the ROS also uses the MQTT architecture. And we will add image streaming and recognition functions. In addition to providing general data monitoring, this system can also perform large-scale image streaming data transmission and image recognition functions. Let this system be applied to more industries.

References

1. Rockwell also provided a complete set of software and hardware solutions
2. IBM "Nodered"
3. NDUKWE, Cherechi; IQBAL, M. Tariq; KHAN, Jahangir. Development of a Low-cost LoRa based SCADA system for Monitoring and Supervisory Control of Small Renewable Energy Generation Systems. In: 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2020. p. 0479-0484.
4. SINGH, Prithvi Pal; KHOSLA, Praveen Kumar; MITTAL, Mamta. Energy conservation in IoT-based smart home and its automation. Energy conservation for IoT devices, 2019, 155-177.
5. JAYASHREE, L. S.; SELVAKUMAR, G. Getting Started with Enterprise Internet of Things: Design Approaches and Software Architecture Models. Springer Nature, 2020.
6. WebSocket,2021/12/08

Authors Introduction

Dr. Jr. Hung Guo

He is an Assistant Professor at the Department of Electrical Engineering, National Yunlin University of Science & Technology, Douliou, Taiwan. He received his Ph.D. Degree from National Yunlin University of Science & Technology, Taiwan in 2012. His research interests include sensor network, intelligent systems, intelligent robot. and technical and vocational education

Mr. Tzu-Yuan Lin

He was born in 1997. He received the B.S. degrees in Automation Engineering from National Formosa University of Science & Technology Yunlin, Taiwan, in 2019, and M.S. degrees in Electronic Engineering from National Yunlin University of Science & Technology, Yunlin, Taiwan, in 2021. Currently, he is a software engineer in Smart Creating Co., Ltd. His research interests include Artificial Intelligence, Internet of Things, SCADA, Embedded Systems and Cloud Computing.

Dr. Kuo-Hsien Hsia

He received the Ph.D. degree in electrical engineering from the National Sun Yat-Sen University, Taiwan, in 1994. He is currently an Associate Pofessor of National Yunlin University of Science and Technology in Taiwan. His research interests are in the area of mobile robotics, fuzzy control and image processing.