中國機械工程學刊第四十一卷第五期第 585~594 頁(民國一百零九年)
Journal of the Chinese Society of Mechanical Engineers, Vol.41, No.5, pp 585~594 (2020)

Using MQTT IIoT Gateway to Implement Bi-direction Communication Manufacturing Information System for Smart Factory

Shang-Liang Chen*, Kuei-Ming Lee**and I-Ching Li**

Keywords: MQTT, Internet of Things, IIoT Gateway, Manufacturing, Smart Factory.

ABSTRACT

To improve the economic growth rate, all countries are committed to the concept of a smart factory. To upgrade the manufacturing equipment in a smart factory from automation to intellectualization, all the manufacturing information must be collected, analyzed, and used. To obtain the information, such as sensing, motion, actuation, etc., the concept of the Industrial Internet of Things (IIoT) must be imported into the factory. In a factory, the devices are made from different suppliers, so that information may not be easily exchanged between them. Hence a unified information format or communication protocol is very critical.

A bi-direction communication system for a smart factory is then designed and developed in this research. The communication protocol MQTT is installed in an embedded system, which is used as an intelligent IIoT gateway. The MQTT protocol is also installed into the controller of equipment, which will be used to collect sensing data, production information. The obtained data and information can be transmitted between different devices by using the MQTT protocol to achieve network communication between machines and IoT devices. By using the MQTT, a mobile application platform is also developed and implemented on the Android system, so that the required information can be obtained from heterogeneous communication interface easily.

INTRODUCTION

Nowadays, the concept of smart factory is developing rapidly. IIoT technology is used in a smart factory to collect real-time information from different manufacturing fields and analyze the collected equipments are made from different suppliers, so that information may not be easily exchanged between them. So how the information can be transmitted between various devices with a heterogeneous communication interface is an interesting problem for achieving the purpose of smart factory. Therefore, a network architecture based on the open communication protocol - MQTT (MQ Telemetry Transport) for transferring information between the equipment in the factory is proposed in this study. Here, the transmission architecture is planned according to functional layers and sensors are deployed on the machine tool. The Raspberry Pi is used as an IIoT Gateway for collecting and filtering sensing data. The sensing data will be transmitted by IIoT gateway to the machine controller or cloud server, which can assist users to monitor the status of the equipment. For engineers can ubiquitously monitor the machine, the application in this research is developed on a mobile Android platform. With the mobile platform, all the information and functional modules of the equipment can be monitored and managed.

Application of IIoT Communication Technology

IoT technology is developed for intelligent applications in recent years. A large number of sensors, actuators, computing units, etc. with different communication interfaces are imported into the system. Therefore, it is essential to select the appropriate communication protocol to transmit information among equipment effectively and to obtain information quickly and accurately in the system.

Markel Iglesias-Urkia et al. pointed out that the devices in the manufacturing system of Industry 4.0 need to be able to communicate with each other to achieve collaboration, monitoring, and control. After the introduction of Cyber-physical systems (CPS), the number of small devices in the production line will increase significantly. If the general network communication protocols such as SOAP and HTTP are used, there will be large energy consumption and high memory computing requirements. Many lightweight communication protocols such as MQTT,

Paper Received September, 2020. Revised Sepfember, 2020, Accepted October, 2020. Author for Correspondence: Shang-Liang Chen

** Professor, Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan 70101, R.O.C.*

*** Graduate Student, Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan 70101, R.O.C.*

CoAP, DDS, JMS, XMPP, MQTT-SN, and WebSockets are adopted by the IIoT. They will reduce the transmission delay, compared with SOAP or HTTP, and are helpful for equipment connection and industrial applications.

Ala Al-Fuqaha et al. summarized the current popular IoT communication technologies and compared CoAP, MQTT, AMQP, DDS, and other protocols. Each protocol has its own appropriate conditions and applications. Different devices usually use different protocols, then, a service that can automatically convert one protocol to another protocol is required.

MQTT Protocol

MQTT (MQ Telemetry Transport) is a Client/Server, Publish/Subscribe messaging transport protocol that belongs to ISO standard. MQTT protocol is based on TCP/IP, and the header of the MQTT packet compared with HTTP protocol is very small, so it is suitable for transmission in poor network conditions. The followings are the MQTT features.

1. It is suitable for communication in M2M and IoT applications.
2. Publish/Subscribe message pattern is used. There are three important roles in the protocol: (1) publisher (2) subscriber as the client, and (3) broker as the server.
3. It is lightweight, open, simple, and designed for easy implementation.
4. It is suitable for the unstable network with low bandwidth or high latency.
5. The "Topic" is used to classify messages. The IP address and port of the broker are also needed for transmission.
6. There are three kinds of qualities of service (QoS): at most once, at least once, and exactly once.

Different from the general client/server model, the MQTT protocol is a publish/subscribe model. The publisher sends specific information. The broker will receive the information and transmit it to the designated subscribers. In other words, the broker is responsible for all information delivery.

MQTT uses a topic-based approach to classify information, and each piece of information belongs to a topic. The subscriber subscribes to the required topic. After the publisher sends information to the broker, the broker will transmit information to the subscriber by the topic depends on IP address and port number.

The MQTT communication protocol has been widely investigated in previous studies. For example, Ala Al-Fuqaha et al. announced that MQTT is an optimal communication protocol for the IoT and M2M. D.C. Yacchirema et al. implemented an IoT gateway to convert the collected data from different sensors into a unified format through the MQTT protocol. They also pointed out that MQTT can transfer information in lightweight with limited resources and enable heterogeneous devices to communicate with each other. Yu-Hsuan Yen combined the IoT communication and laser guidance technology to improve the performance of existing Automated Guided Vehicle (AGV). The data packets of the status of AGV system are analyzed and converted into MQTT format packets so that the power and speed of the vehicle can be real-time monitored.

From the above researches, it can be concluded that if the MQTT protocol is applied to the IIoT gateway in this study, it can reduce the packet and bandwidth requirements. If it can be introduced into the equipment in the factory, it will improve the real-time monitoring performance.

Wireless Communication in the Factory

There are many equipment and devices usually used in a factory. Problems such as difficult wiring and poor communication efficiency are often encountered.

Shang-Liang Chen et al. proposed a Wi-Fi-based remote monitoring platform for cloud-based machines in a factory, using Wi-Fi to capture manufacturing information on the machine to achieve M2M in the plant. Collecting manufacturing production data in real-time, engineers can query and monitor the status of the machine ubiquitously. The method of Wi-Fi module deployment is installing a Wi-Fi Node on each machine to build a wireless connection function. Finally, the machine information is transmitted to the server for storage via the wireless network.

Ming-Huang Chen proposed a cloud platform for machine wireless remote monitoring based on Wi-Fi technology. The embedded Wi-Fi sensing system and various sensors are deployed on the machine to monitor the machine and manufacturing site. Besides, a dynamic measurement system is also designed to store important parameters such as current, pressure, and vibration signals of machines in a cloud database through a wireless network monitoring system.

Regarding the wireless network architecture in the above researches, a wireless environment for information transmission will be built in this study.

Application of Mobile Technology in Manufacturing

Due to the rapid development of mobile technology, the number of users who use smartphones and tablets to browse Internet services has gradually surpassed the number of users of computers. The trend of applying mobile technologies is also beginning to spread to manufacturing.

D. Mourtzi et al. developed applications for Android devices and introduced them to the automotive industry to test their applicability. Enabling Original Equipment Manufacturers (OEMs) to remotely manufacturing management to meet performance indicators such as cost, time, quality, reliability, and flexibility under mass customization. J. Wan et al. introduced mobile services and cloud computing technology into an intelligent manufacturing environment to build a personalized custom manufacturing system (PCMS). A large amount of product information is transmitted to the cloud through a wireless module, and the optimization decision is made by the cloud. Pin-cheng Lu introduced the motion control module based on the EtherCAT into the Android mobile device. The Client/Server architecture based on Socket communication protocol provides users to monitor and control machines in the factory.

From the above researches, mobile application technology can make the management and acquisition of various production information in the factory more convenient and ubiquitous.

A network architecture based on IoT communication protocols is proposed in this study. The following are the research purposes of this study.

1. An IoT communication protocol MQTT is introduced into network architecture in this system. The subscribe/publish model is used to construct the information transmission between the machine and the sensing module to achieve the heterogeneous interface communication.
2. The embedded device Raspberry Pi is used as the IIoT gateway to collect and analyze the sensing data. The analyzed information will be published to the devices which subscribe to it.
3. Android system is used to build a mobile platform and functional modules, including information monitoring, hardware system management, and information transport services. Through this platform, users can get the manufacturing information anywhere and handle problems immediately to achieve bi-direction interaction.

SYSTEM MODELS

The system architecture is divided into five functional layers, and the architecture diagram is illustrated in Figure 1. The key technologies are presented in the following.

1. Sensing Layer: The surrounding of the machine tool is planned to be equipped with various sensors. IIoT gateway is built based on Raspberry Pi, and the sensing data is received by it. After the data is collected and filtered, these data will be upload to the motion control layer.
2. Transport Layer: The way of information transporting between different layers is defined, including data transporting methods, transport entities, and protocol rules. MQTT is adopted in every layer for simplifying and accelerating data exchange.
3. Motion Control Layer: The control core of this layer is a universal controller, which can receive the sensing information collected by IIoT gateway, and display it on the human-machine interface.
4. Data Management Layer: The format and type of needed data, the authority of data access, and the entity of data storage are defined. The information from every layer is stored in a cloud data center.
5. Application Layer: An Android APP is designed to receive sensing and machine information, and the needed information can instantly be viewed.

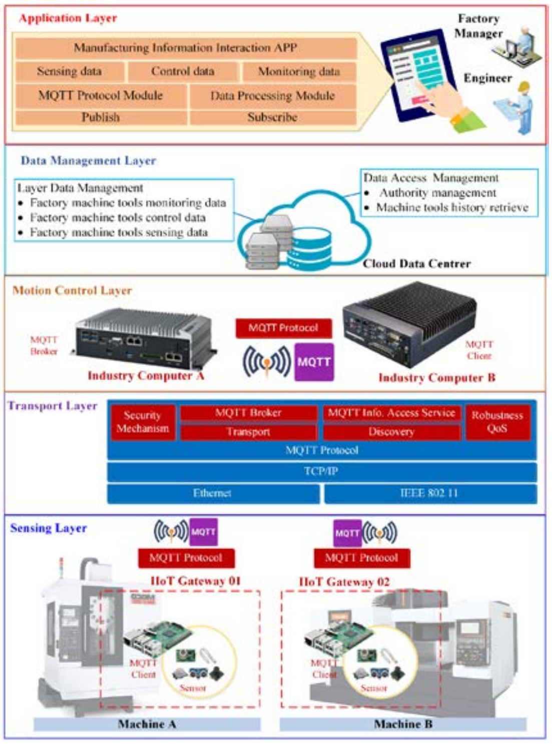

Fig. 1. The bi-direction communication manufacturing information system architecture diagram.

Design of IIoT Gateway Function Module

An IIoT gateway is developed at the sensing layer in the system architecture, which is responsible for data processing, analysis, and filtering of the collected sensing information. Embedded device units on the market are evaluated and key factors are considered, such as price, computing power, additional functions, peripheral modules, and opensource software resources for analysis. It is shown that most Makers currently use to develop IoT applications is Raspberry Pi. Therefore, Raspberry Pi will be used in this study as the core of the sensing layer.

The function module of the IIoT gateway includes five modules described as follows.

1. Transmission function module: Receive and transmit sensing data and control commands.
2. Advanced function module: According to the requirements of the sensors on IIoT gateway, advanced functions such as the abnormal alarm notification function is developed.
3. Parameter setting module: Various communication transmission parameter settings are provided.
4. Schedule execution module: The ordering of each function is designed to make the program execute in order.
5. Process control module: Responsible for module status record and debug during execution.

Design of Sensing Layer

The Raspberry Pi is used as IIoT gateway and set to be the client and the publisher of the MQTT protocol. The sensing information transmission situation is shown in Figure 2. After the sensor information is collected by the Raspberry Pi, it will be transmitted to the controller as the MQTT broker. At the same time, the controller can also act as a subscriber to receive and display sensing data.

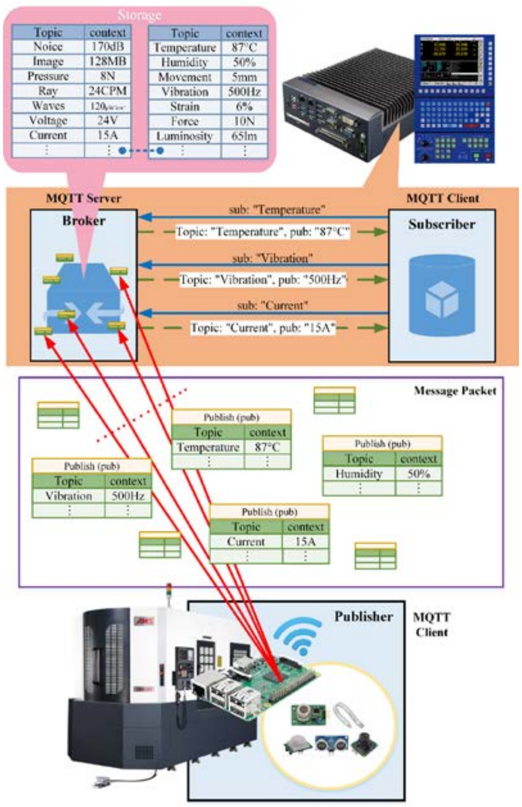

Fig. 2. Sensing information transmission situation.

The overall topic planning is shown in Figure 3. The topics respectively are factories, plants, all machines in each plant, the IIoT gateway configured on the machines, and the various sensors on the IIoT gateway.

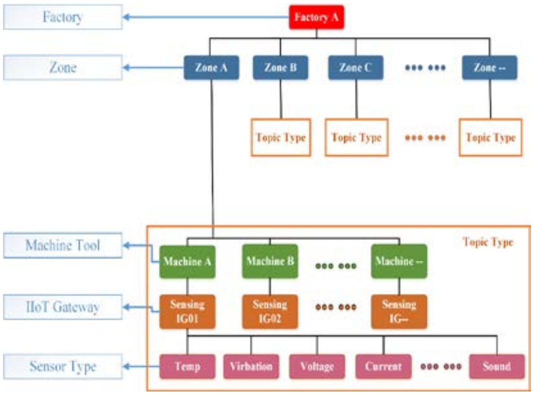

Fig. 3. Topic planning for information transmission.

Design of Motion Control Layer

The core of the motion control layer is a universal controller set in each factory, and there are two kinds of roles in the following. The architecture is shown in Figure 4.

1. Only as a client of the MQTT protocol, it is used to receive the sensing information released by the IIoT gateway.
2. The controller acts as both the broker and the client of the MQTT protocol. When controllers need sensing information from the Raspberry Pi, the controller, as the subscriber, will obtain the subscription information through the controller as the broker.

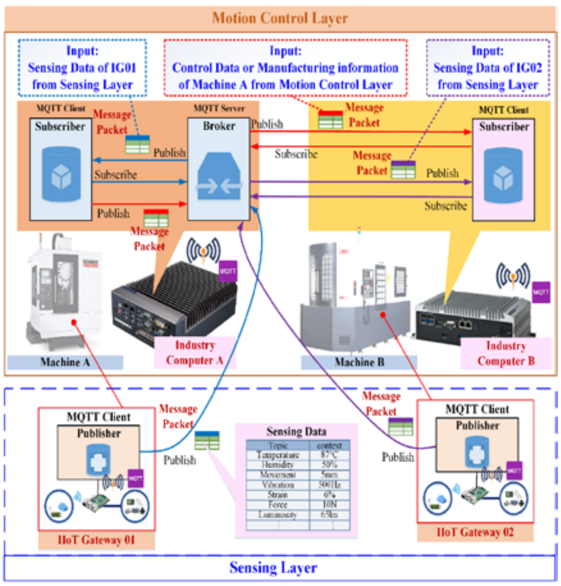

Fig. 4. The architecture diagram of the motion control layer.

Design of Data Management Layer

In this system, the data management layer is responsible for defining the types of information and the authority of viewing the information. A cloud database center is deployed in the factory, and a database is established to store data. The design of this layer is as follows.

1. Plant personnel management: The needed personnel information is recorded in the plant, and information reading permissions are assigned according to the personnel's position.
2. Plant information management: The number of plants and the types and information of the machines in the plant is recorded.
3. IIoT gateway management: The IIoT gateway information configured beside each machine is recorded, as well as the type and number of each sensor.
4. Sensing data management: All data received and transmitted by IIoT gateway is recorded.
5. Machine control data management: The relevant parameters of the machine and all the data transmitted by the machine controller is recorded.
6. Historical operation record management: The information of the query during the operation of the mobile application is recorded.
7. Abnormality management: The alarms of anomaly sensing information issued by IIoT gateway is recorded.
8. Processing record management: The processing actions after an abnormal event occurs on the mobile application is recorded.
9. MQTT setting management: The address of the broker in the plant and all subscription topics are recorded.

Design of Application Layer

To enable personnel to get the information of the system anywhere, a mobile information delivery platform on smart mobile devices is proposed in this research. The flow is shown in Figure 4. This platform is divided into two main modules, which are "sensing and motion control information management module" and "bi-direction communication manufacturing information module." The detailed functions of each module are described below.

Sensing and motion control information management module:

1. Permission management module: Login function is provided. According to the authority level of different personnel, some operation functions of different modules will be locked in the operation interface.
2. Sensing information reading module: When the user wants to know the sensing information of a specific machine, this module provides a selection interface to facilitate the monitoring of various sensing information.
3. Motion control information reading module: When the user wants to know the operating status of a specific machine, this module provides an operator interface to view the machine information of a specific item.
4. Setting management module: Functions of increase, decrease, or modify the basic data in the system database are provided for senior managers. The data that can be modified include personnel permission settings, factory machine management, IIoT gateway module management, and MQTT broker and subscription topic management.

Bi-direction communication manufacturing information module:

1. Bi-direction communication module: Users can subscribe and publish messages on topics with feedback information function. If an exception occurs in the system, notifications will be sent to users who subscribe to the topic. Status reports, exception information, and processing results will be recorded in the system.
2. Manufacturing information delivery module: Important information can be selected for subscription according to personal needs. After the setting is completed, the module runs in the background program. Needed information will be received continuously, and the user will be notified immediately. As a result, personnel can reduce the time of finding information on the production information page, and can directly use the mobile device to obtain the required machine information.

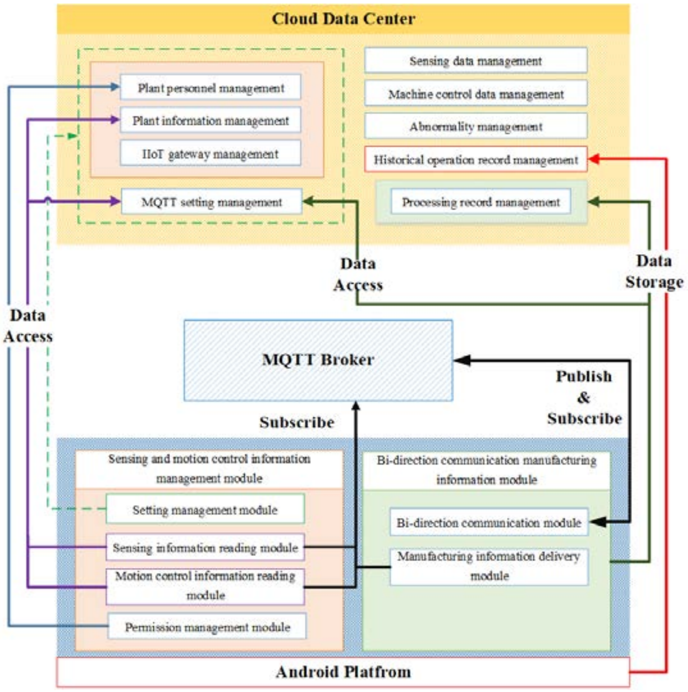

Fig. 5. The flow of mobile platform access to the database.

Evaluation and Selection for the Protocol Development Tools

There are many open sources or paid software that can be chosen to build the server and client of the MQTT protocol. Mosquitto is an opensource message broker that supports the development environment such as Windows, Linux, MAC OS X, and Raspberry Pi. Open-source MQTT client is provided by the Eclipse Paho project, and there are implementations in many kinds of language and libraries. Related resources are easy to acquire in GitHub, so we use the above two open sources to construct MQTT applications in this research.

RESULTS

The bi-direction communication manufacturing information system for smart factory is divided into three parts, the IIoT gateway, the integrated interface of the controller, and the application of the mobile device.

MQTT Protocol Based IIoT Gateway

The client of the MQTT protocol is built on the Raspberry Pi as the IIoT gateway. The Linux operating system Raspbian is used for function development on Raspberry Pi. An opensource project developed by the Eclipse team is used to implement the function of the MQTT client. It provides multiple programming languages and libraries for different development platforms, allowing users to develop on different platforms

The architecture diagram of the functional module for the MQTT protocol based IIoT gateway is illustrated in Figure 6. There are five functional modules, and these modules are defined as follows.

1. Data & Command Transmission module: Sensing data and control commands are received and transported by this module.
2. Advanced function module: According to the sensors on IIoT gateway, advanced functions such as the abnormal alarm notification function are developed.
3. Parameter setting module: The setting of the hardware parameter and the transmission parameter is provided.
4. Schedule execution module: The running procedure of every functional module is arranged. This module is executed as the main program.
5. Process control module: Responsible for recording the status and debugging of each module in the execution period.

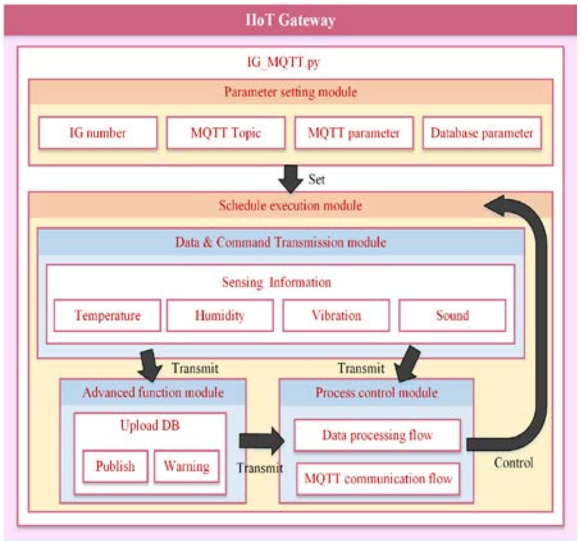

Fig. 6. The architecture diagram of the functional module for MQTT protocol based IIoT gateway.

The Integrated Interface of the Controller

There are MQTT broker and client functions on the controller. The EtherCAT Motion Control Platform (EMP) developed by the Institute of Mechanical Engineering in ITRI is used to construct the environment of the controller. Windows 7 and XP (based on RTX) systems, as well as Visual C ++ / VB / C # / VB.net are supported, and Motion Control Library (MCCL) compatible with EtherCAT protocol is provided.

Mosquitto message broker and Paho Client library are used to build MQTT functions on this industrial controller. The functions of MCCL library and MQTT library provided by Eclipse Paho to .Net (C #)-M2Mqtt (v4.3.0) are integrated into this system. Visual Studio 2015 is used to develop an integration interface of the controller system, and the MCCL is used to send control commands to the machine.

Interface functions include servo system drive, machine current position reading, encoder feedback position, home position returning, spindle speed adjustment, spindle control, and two sets of motion control command demonstrations. The MQTT communication transmission function module is integrated into the interface, which can subscribe and present data for the IIoT gateway beside the machine, send the machine motion control information to the required client, and store it in the database. The functional module architecture is shown in Figure 7.

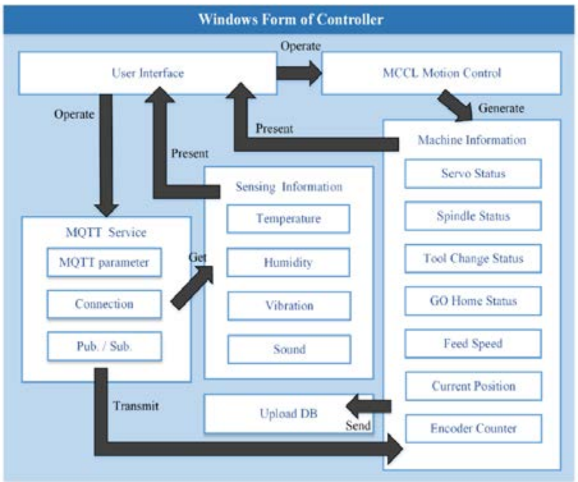

Fig. 7. The architecture diagram of integrated interface function modules of the controller.

Application for the Mobile Device

For the establishment of the mobile application, a service supporting Paho Java MQTT Client library-paho.android.service: 1.0.2 must be added to the dependencies, and SQL Server JDBC driver-JTDS (v1.2.7) should be added to connect the cloud database built by SQL Server 2012. The APP system architecture is shown in Figure 8. The Login Activity page provides the function for users to log in to this interactive platform. The data entered by the user is compared with the database, and the function of the system management module maybe will be locked when entering the MainActivity according to the user's different permissions. The MainActivity page is used to switch and present the Fragments of various functions. The contents of each Fragment are described as follows.

1. Sensing Fragment：Spinner drop-down menu is used to connect to the system database and read the machine data table (Factory, Zone, MachineTool) configured in each plant. When the user selects the machine to get the sensing information, its topic will be passed to the setSubscription method to be subscribed. The opensource drawing suite MPAndroidChart: v3.0.3 is installed, a new class ChartHelper is added to draw the sensing information into the chart, and visualize the sensing data dynamically.
2. Control Fragment：The same subscription topic method is used as a Sensing Fragment. The messageArrived method is implemented in the MqttCallback category to dynamically modify the value displayed on the TextView, according to the topic of the received information.
3. Management Fragment：View, add, modify, and delete functions on the data table.
4. Interaction Fragment：The interface is divided into three parts, connection settings, information delivery, and subscription information. Spinner is used to select the connected broker, and subscribe or publish in the system. The sent or received information will be updated in the ListView on the interface.
5. Active Fragment：Connection setting method is the same as Interaction Fragment's, but there is only subscription function. It can be executed in the background. When receiving the information, the Notification function will be performed, and the information will be presented in the notification bar of the mobile device.

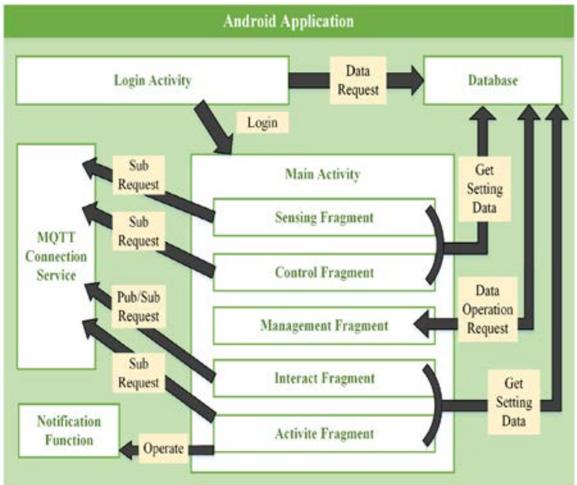

Fig. 8. The architecture diagram of the functional module for mobile application.

SYSTEM TESTING AND ANALYSIS

Tools of MQTT broker and client are used for connection and developed on Raspberry Pi, controller, and mobile platform. In the implementation section, first, install the MQTT broker and client packages on the Raspbian Linux system of the Raspberry Pi, and use the apt command to install the mosquitto and mosquitto-clients package. After the broker is activated, the local information transmission test can be performed.

Then the application package MQTTLens provided by Google's web browser on the Windows system can be installed to perform the MQTT publish/subscribe test. In addition, the IoT MQTT Dashboard application that supports the Android system is used to simulate the client for data transmission and server connection testing.

In this research, the applied situation of manufacturing information is designed according to the different topics. The system has been proved that it can be used to show values transmitted by each device correctly, and the testing results are demonstrated in Figure 9.

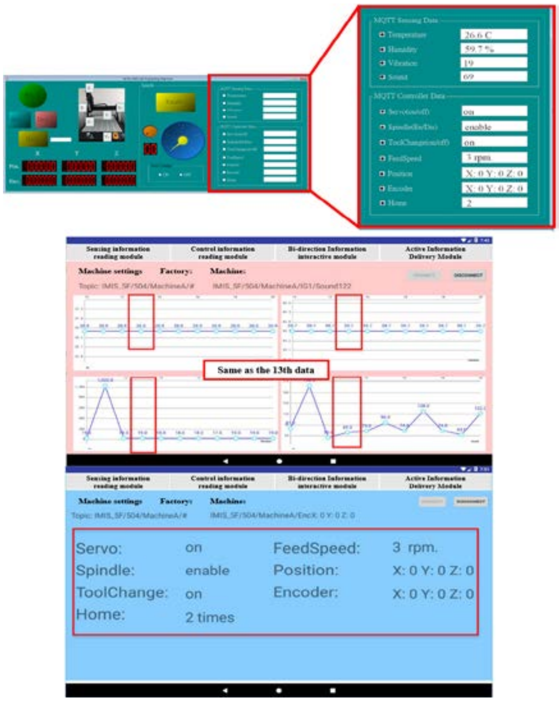

Fig. 9. Testing picture of the system in applied situation.

Implementation and Testing of Data Transmission at the Sensing Layer

The Raspberry Pi is used to build the IIoT gateway as the MQTT client for sensor data collection and publishing sensing information to the MQTT broker.

Implementation and Testing of Data Transmission at the Control Layer

MQTT broker is set up on the EtherCAT motion control platform, and an operating interface is designed for users to perform machine motion control. Besides, an MQTT information transmission module is implemented, which can be used to subscribe to the sensing information and publish machine control information.

Implementation and Testing of Data Transmission at the Application Layer

An application is implemented for a bi-direction communication manufacturing information system on an Android platform. This platform can be used to present sensing and machine control information, as well as to subscribe and publish specific information. After the user selects the machine to be monitored, the machine information will be received through the MQTT broker. After receiving the sensing information of the topic, the sensing information will be presented in the form of values and charts, and the displayed sensing information will be dynamically updated when the information receives.

DISCUSSION

Three network architectures based on MQTT protocol are discussed and shown in table1.
A. The MQTT broker is built on a machine controller
B. The MQTT broker is built on a Raspberry Pi
C. The MQTT broker is built on a management server

For each architecture, the implementation and application scenario are analyzed and organized into a comparison table shown in Table 1. If there is an importing requirement for the factory, this analysis can be referred to plan the blueprint of deployment and application.

Table 1. Comparison of MQTT protocol network architectures.

Architecture	A	B	C
Technical requirement	**High (Engineer responsible for machine maintenance is required)**	**Low (the embedded system is required to import)**	**Medium (network administrator is required)**
Construction cost (embedded device, network management device, MQTT software)	**Low**	**Low**	**Low**
The difficulty of maintenance (knowledge of hardware, number of nodes)	**High**	**Medium**	**Medium**
number of client connections (depend on system specifications)	**Medium to High**	**Low**	**High**
Application situation	**There is a machine tool with a small workload in the factory**	**It will not affect the original machine equipment or management system**	**The factory is equipped network management server**

CONCLUSIONS

In this research, the embedded device and the architecture of the IoT communication protocol are introduced into the factory. A base of the IIoT environment is built, which can help the factory to transform and develop intelligent applications. The results of this research are described as follows.

1. A network architecture based on MQTT protocol in the factory is constructed, and the communication bridge can be built between heterogeneous interfaces such as machine tools and IoT devices. The data can be standardized and lightweight to make information transmission faster.
2. The IIoT gateway is developed with Raspberry Pi to build an embedded sensing module. Sensing information besides the machine tool can be collected, and then be transformed into MQTT format and be sent to the subscriber.
3. The application of network architecture is implemented in the Android platform so that personnel can access to sensing and machine information anywhere. There is a manufacturing information delivery service to inform the personnel of the status of the manufacturing site. The personnel can also return some operations to achieve a two-way interaction between the human and the machine.
4. After the implementation and analysis of this system, three network architecture for different applied environments is discussed. The result of the discussion can as reference for the enterprise to import IIoT sensing technology and communication architecture into the factory.

ACKNOWLEDGMENT

Funding and technical support for this work were provided by Industrial Technology Research Institute (ITRI) and Ministry of Science and Technology, R.O.C (MOST). Under the contract B109-F007, MOST 107-2221-E-006-230-MY2, and MOST 109-2221-E-006-098-. Because of the support of ITRI and MOST, this study could be conducted smoothly.

REFERENCES

Accenture(2012). Mobile web watch survey.

Al-Fuqaha, Ala, 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.

D. C. Yacchirema Vargas and C. E. Palau Salvador, “Smart IoT Gateway For Heterogeneous Devices Interoperability,” IEEE Latin America Transactions, vol. 14, no. 8, pp. 3900−3906, 2016.

D. Mourtzis, M. Doukas, and C. Vandera. “Mobile apps for product customisation and design of manufacturing networks.” Manufacturing Letters, vol. 2, no. 2, pp. 30−34, 2014.

Iglesias-Urkia, Markel, et al. “Towards a lightweight protocol for Industry 4.0: An implementation based benchmark.” Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), 2017 IEEE International Workshop of. IEEE, 2017.

J. Wan, et al. “Mobile services for customization manufacturing systems: an example of industry 4.0.” IEEE Access, vol. 4, pp. 8977−8986, 2016.

Kim, Seong-Min, Hoan-Suk Choi, and Woo-Seop Rhee. “IoT home gateway for auto-configuration and management of MQTT devices.” 2015 IEEE Conference on Wireless Sensors (ICWiSe). IEEE, 2015.

Light, Roger. “Mosquitto: server and client implementation of the MQTT protocol.” Journal of Open Source Software 2.13 (2017): 265.

Locke, D., (2010), “MQTT V3.1 Protocol Specification.” International Business Machines Corporation (IBM) and Eurotech, pp 42.

Locke, Dave. “Mq telemetry transport (mqtt) v3. 1 protocol specification.” IBM developerWorks Technical Library 15 (2010).

Ming-Huang Chen(2013). “A Monitoring System Development for Machine Tool Factory Using Wi-Fi Wireless Sensor Network Technology.” National Cheng Kung University, Taiwan R.O.C.

Pin-Cheng Lu (2018). “Mobile Motion Control and Wireless Monitoring System for EtherCAT Machine Tool.” National Cheng Kung University，Taiwan R.O.C.

Shang-Liang Chen, Sin-Ru Wang, Yun-Yao Chen (2013). “Design of a Cloud Platform for Machine Wireless Remote Monitoring Based on Wi-Fi Technology.” TANET 2013, Taiwan R.O.C.

Sinha, Aniruddha, Shubham Sharma, and M. R. Mahboob. “An Internet of Things based prototype for smart appliance control.” 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2017.

Vargas, Diana Cecilia Yacchirema, and Carlos Enrique Palau Salvador. “Smart IoT gateway for heterogeneous devices interoperability.” IEEE Latin America Transactions 14.8 (2016): 3900-3906.

Yu-Hsuan Yen. “Implementation of IoT communication technology on Automated Guided Vehicle.” National Central University, Taiwan R.O.C., 2017.

具 MQTT 協定之工業物聯網閘道器實現廠區製造資訊雙向互動系統

陳響亮　　李桂銘　　李宜靜
國立成功大學 製造資訊與系統研究所

摘 要

為提升經濟成長率，各國政府無不致力於進行工業技術革新與轉型，使得現今智慧工廠的概念被廣泛討論。欲使工廠內各設備由自動化提升為智動化，需整合硬體與智慧化軟體技術，取得、分析並運算設備和生產線的各項資訊，如感測、動作控制、作動流程等。為此工廠必須導入工業物聯網(IIoT)架構，下至感測器、致動器，上至機台控制器、雲端運算伺服器等，各設備間皆需要進行資訊傳輸並應用至大數據分析，這使得統一的資訊格式與通訊協定更顯重要。

本研究開發一廠區製造資訊雙向互動系統，此系統著重於通訊架構的設計，於廠區內設備導入物聯網通訊協定 MQTT，並搭配裝設於廠區機台周邊、以嵌入式系統開發之 IIoT 閘道器。此閘道器可進行感測資訊之收集與過濾，以及機台控制資訊彙整之功能，達到機台與物聯網設備、機台與機台之間的聯網通訊。此外本研究以 Android 系統開發一行動化應用平台，該平台採用 MQTT 的發佈(Publish)、訂閱(Subscribe)模式進行應用功能設計，使廠區內的人員能更加便利與快速的獲取所需的機台相關資訊。