IoT-Based UPS Monitoring System Using MQTT Protocols

Padlan Alqinsi
Department of Electrical Engineering
UIN Sunan Gunung Djati Bandung
Jl. A.H Nasution 105, Cibiru - Bandung 40614, Indonesia
padlanalqinsi@live.com

Nanang Ismail
Department of Electrical Engineering
UIN Sunan Gunung Djati Bandung
Jl. A.H Nasution 105, Cibiru - Bandung 40614, Indonesia
nanang.is@uinsgd.ac.id

Ian Joseph Matheus Edward
School of Electrical Engineering and Informatics
Bandung Institute of Technology
40132 Bandung, Indonesia
ian@stei.itb.ac.id

Wahyudin Darmalaksana
Research And Publication Center
UIN Sunan Gunung Djati Bandung
Jl. A.H Nasution 105, Cibiru - Bandung 40614, Indonesia
yudi_darma@uinsgd.ac.id

***Abstract*— Uninterruptible power supply (UPS) monitoring system using message queuing telemetry transport (MQTT) protocol is a solution to solve UPS monitoring issues in a large infrastructure. This paper discusses the proof of the small size of message data by using the MQTT protocol on IoT-based communications. Based on IoT concept, the system described in this paper used Arduino microcontroller connected to the Internet via an Ethernet shield. The system used MQTT as a communication protocol that was designed for a lightweight communication. This system was intended to display UPS monitoring data in real-time on a web page stored on raspberry pi which roles as a web server, MQTT broker, MQTT subscriber and database. UPS parameter could be monitored using a web-based application. There were some differences in data obtained from sensor with the measurement results of measuring instruments specified for each parameter: the difference of input voltage was equal to 0.20%, while differences of output voltage, output power and output current reached 1.34%, 0.17% and 20%, respectively.**

Keywords—UPS; MQTT; Monitoring System; IoT

I. INTRODUCTION

UPS has an important role in a system that requires a backup of the voltage source to keep the system running even if the main voltage source is interrupted[1]. When the main voltage source of a system is interrupted, the UPS switches the main voltage source to UPS battery without disconnecting power supply source to the system. Hence, the system is still running as when it gets the main voltage source [2]. For this reason, many industrial infrastructures use UPS to keep the system running despite the power source failure [3].

When the main power source is off, the UPS cannot continuously be a source of power for the system because UPS has certain limitations, such as battery capacity and the amount of power that can be generated by UPS[4]. Power supplied by the UPS as a backup power source stops if the battery used by UPS has run out of capacity. This will cause the system to stop suddenly because there is no source of power that sustains the system. Such incidents can result in the damaging of electronics components of the system. For a system that has a large infrastructure, the failure of one of the infrastructures will lead to disruption or even complete failure of the system. Therefore, the UPS can only backup the system during certain times or conditions. However, it is very difficult to monitor the UPS manually as it takes time and cost a lot. Hence, if there is a problem, it is very difficult to troubleshoot the problematic UPS. To solve this problem, this paper proposes a monitoring system using a Machine to Machine communication (M2M) based on the technology of the Internet of Things (IoT) so that technicians do not need to go to where the UPS is installed. They simply need to open a web browser from any device such as laptops, mobile phones and other gadgets that have internet access [5] to view the status or parameters of the UPS that they handle in Real Time. IoT technology has been applied for monitoring needs in various fields such as smart home[5], transportation, environmental monitoring [6], control of electrical power [7], agriculture [8] and others.

The use of MQTT as a communication protocol can save power and bandwidth[9]. MQTT is a lightweight protocol [10] which has a small message size. MQTT architecture using Publish / Subscribe is more suitable for use in IoT than other protocols that use Request / Response because the client on MQTT does not require a request update, resulting in bandwidth savings as well as increase in battery life of the device [11]. This paper discusses the proof of the small size of message data by using the MQTT protocols on IoT-based communications.

II. DESIGN AND WORKFLOW

This section describes the design, workflow and the message format used in the IoT-based UPS monitoring system described in this paper. System design includes types of hardware the system uses, while system workflow describes the workflow of each hardware from measuring the UPS parameter to showing the result in a web-based application. The message format describes the structure of the message used in the system.

978-1-5386-6163-5/18/$31.00 ©2018 IEEE

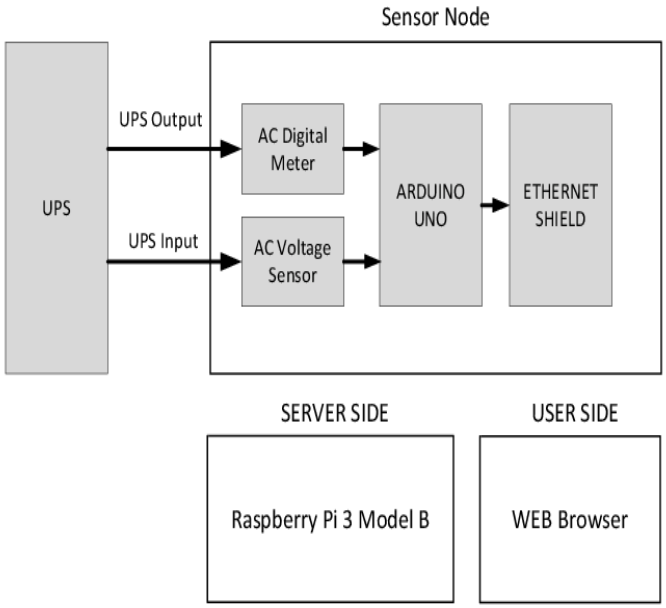

Fig. 1. System Design Diagram

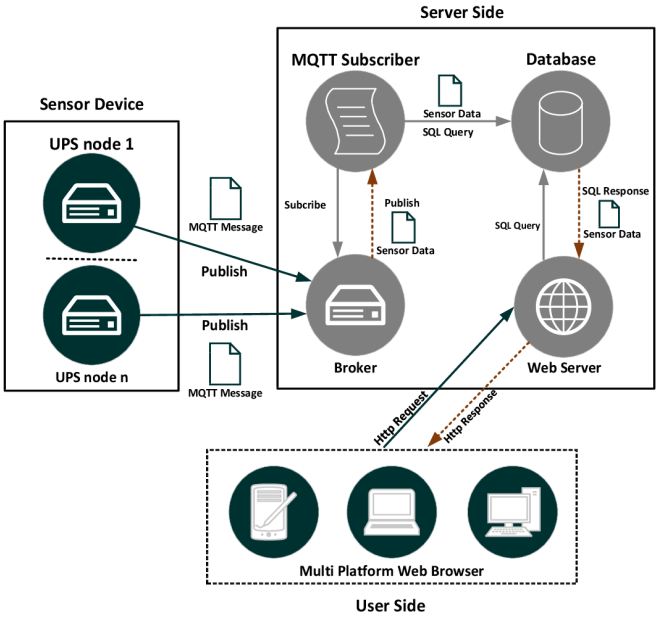

Fig. 2. System Workflow Diagram

A. *System Design*

The system consisted of 3 main parts: Sensor Node, Server Side and User Side. Sensor node consisted of two sensors that read input and output parameters of the UPS and an arduino microcontroller board connected to the internet via the Ethernet Shield. Raspberry Pi 3 Model B was used as MQTT Broker, Web Server and Database Server on Server side.

In the sensor node, two types of sensor were used: PZEM 004 as digital meter and ZMPT101B for voltmeter. Digital meter was used to measure output parameters of UPS which include Output Voltage, Output Current, as well as Output Power. On the other hand, the source voltage (Input Voltage) was measured by ZMPT101B. Raspberry Pi on the server side was installed with Mosquitto MQTT Broker, MySQL Database, and Apache web server; while on the user side, the monitoring software was displayed as a web-based application. This application could be accessed through a web browser on their computer or smartphone that were connected to the network. Figure 1 shows the diagram of the system designed.

B. *System Workflow*

In the designed system, sensors device act as MQTT Publisher which published sensor data to MQTT Broker that was installed on Raspberry Pi. Server-side (Raspberry Pi) had several roles such as MQTT Broker, MQTT Subscriber, Database and Web Server. MQTT Subscriber stored published data on topic to the database. The web page would then retrieve data to be displayed from the database. Figure 2 illustrates how the system worked.

C. *Message Format*

The sensor node published the sensor value to a topic in the MQTT broker. The published sensor value was then converted to a string data type that had 5 numbers separated by a space character. For example, a message of "1 221 220 200 3" published by the sensor node were meant as values available in Table I.

TABLE I. MESSAGE FORMAT

Message = 1 221 220 200 3	
Parameters	***Value***
Device ID	1
Input Voltage	221 Volt AC
Output Voltage	220 Volt AC
Output Power	200 Watt
Output Current	3 Ampere

III. RESULT AND DISCUSSION

In this paper, there were 3 aspects tested, which were sensor testing, MQTT Broker testing and software testing. In the sensor test, the value measured by the sensor were compared to measurement results by using measuring instruments. There were 4 parameters compared, i.e. input voltage, output voltage, output power and output current. The MQTT Broker was tested by running a script to publish messages to the topic used in the system. The script contained the sensor data and also the device ID that publishes the message. This script would then run on 3 different devices at the same time, so it could be assumed that the message received by the broker was a message published from the actual sensor node. The aspect tested during the web page test was if the web feature were able to show the sensor data from multiple devices.

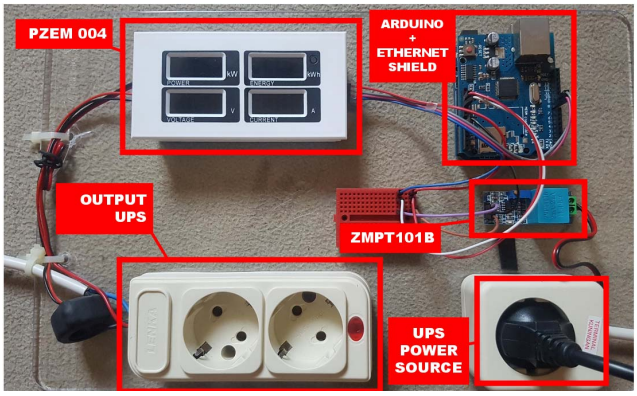

Fig. 3. Sensor Node Implementation

A. *Sensor Measuring Test*

After the system had been successfully implemented, the sensor was tested by taking 10 samples of comparison of results from UPS parameter reading values using sensors and measuring instrument. Input voltage was detected by ZMPT101b while voltmeter was used as the measuring instrument. On the other hand, PZEM 004 and voltmeter were used as a sensor and measurement instrument for the output voltage, respectively. For the output power, the sensor used was PZEM 004 and the measuring instrument was a digital watt meter. PZEM 004 was also used as sensor for output current, but clamp meter was used as the measurement instrument of the current. Figure 4 – Figure 7 shows measured value using sensor and measuring instrument.

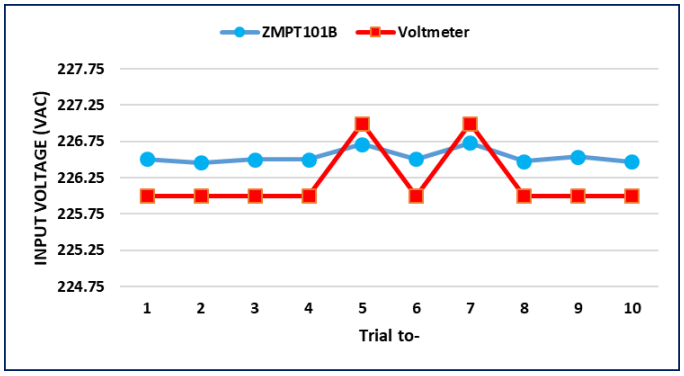

Fig. 4. UPS Input Voltage Measurement

Fig 4. shows the comparison graph of input voltage measurement using ZMPT101B and voltmeter. Based on the graph, the average difference of measurement is 0.20%. The figure shows that the results of measurements with sensors that have a fair value measurement difference.

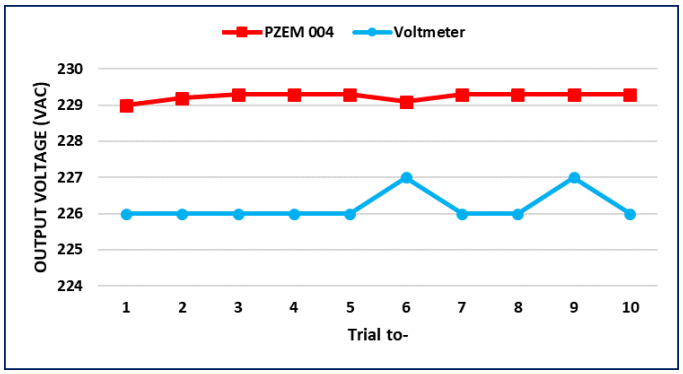

Fig. 5. UPS Output Voltage Measurement

Fig 5. shows the graph of comparison of output voltage measurement using PZEM004 and voltmeter. Based on the graph, the average difference of measurement is 1.34%.The figure shows that the results of measurements with the sensors have a good value.

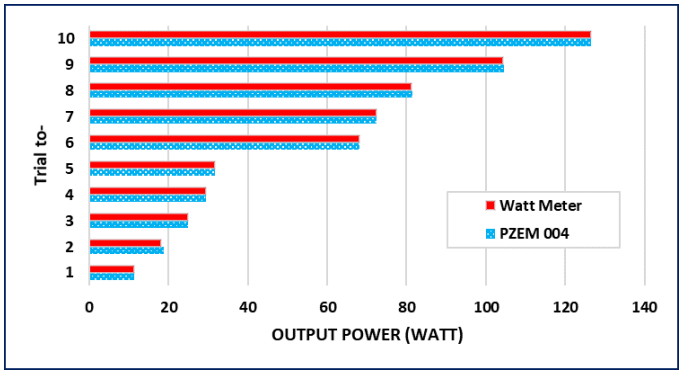

Fig. 6. UPS Output Power Measurement

Fig 6. shows the measurement comparison graph of the output power using PZEM004 and the digital watt meter. Based on the graph, the average difference of measurement is 0.17%. The figure shows that the results of measurements with sensors that have a fair value measurement difference.

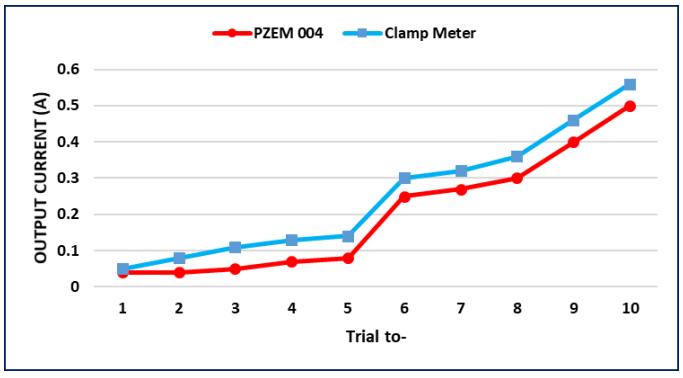

Fig. 7. UPS Output Current Measurement

Fig 7. shows a graph of comparison of output current measurement using PZEM004 and clamp meter. Based on the graph, the average difference of measurement is 20%. These figures shows that current measurement using PZEM004 not too good to be used.

B. MQTT Protocols

After the sensor node had worked properly, the next aspect tested was the MQTT Broker. In this system, the broker was installed in Raspberry pi 3 model B. The MQTT Broker was then tested by showing the message received by the Broker. The test is shown in Figure 8.

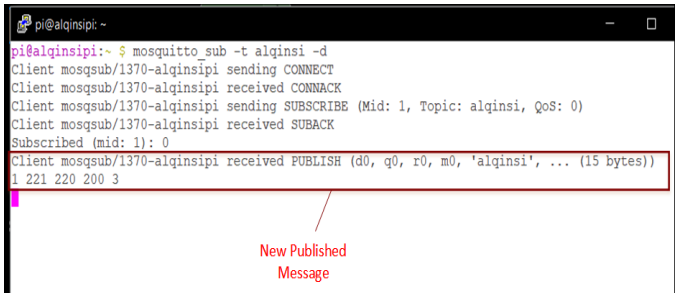

Fig. 8. Message Published to MQTT Broker

After the broker had received the message correctly, the next test conducted was MQTT Broker concurrency testing. If the broker could receive a message from a sensor node, it would be tested by sending multiple messages simultaneously. The messages were not sent by the sensor node, but by a script that was assumed to be the sensor node. Each message sent by the script had a different device ID that represented the messages coming from different sensor nodes.

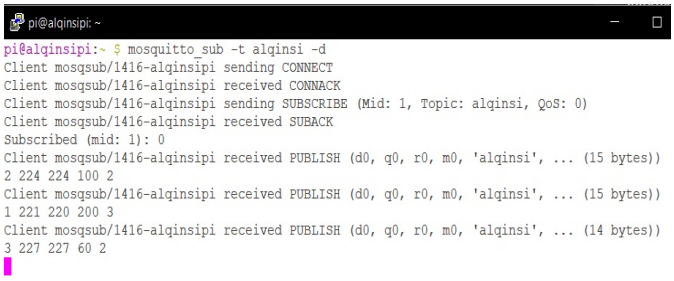

Fig. 9. Multiple Device Published a Message

Based on Fig 9. the use of MQTT protocol on this system only requires 14bytes up to 15bytes for one time delivery. This has an impact on the low power consumption of the system.

C. Testing the Web Page

Data from each sensor node were displayed in a web page. The web page was designed to be able to choose which sensor nodes will be displayed in accordance with the device ID. The home view of the designed web page can be seen in Figure 10, while Figure 11 and Figure 12 show UPS parameters with UPS 1 and UPS 2. The web pages in Figure 11 and Figure 12 were displayed in Google Chrome.

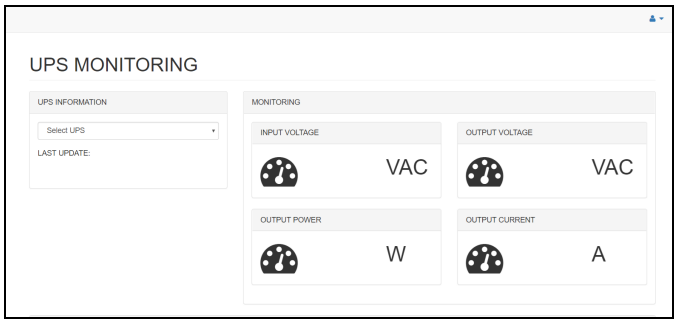

Fig. 10. Web Page Interface

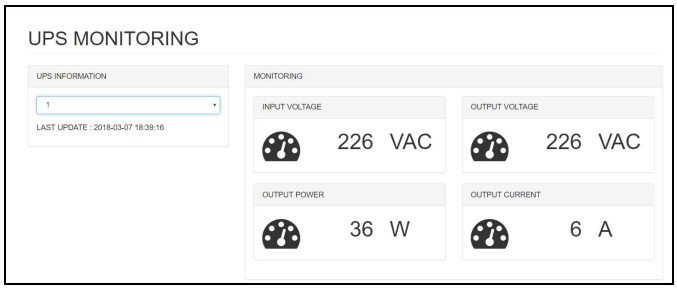

Fig. 11. Select UPS with device ID 1

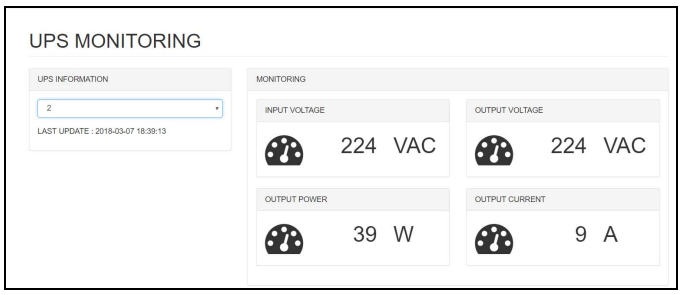

Fig. 12. Select UPS with device ID 2

IV. CONCLUSION

The developed system allowed the displaying of multiple UPS parameters in Real Time. The parameters that were monitored include the input voltage, output voltage, output power and output current with the size of the transmitted data are only about 15 bytes. The parameters were displayed in a web page in order to make them possible to be accessed in various devices that have a web browser. The designed system is low-cost because it uses lightweight MQTT protocol but it is powerful enough to handle multiple devices at once.

REFERENCES

[1] R. Wang, H. Wang, Y. Deng, and F. Zou, "The Design of UPS Battery Online Monitoring Instrument," *Adv. Sci. Technol. Lett.*, vol. 53, no. Isi, pp. 16–19, 2014.

[2] S. B. Bekiarov and A. Emadi, "Uninterruptible power supplies: classification, operation,dynamics, and control," *APEC. Seventeenth Annu. IEEE Appl. Power Electron. Conf. Expo. (Cat. No.02CH37335)*, vol. 1, no. c, pp. 597–604, 2002.

[3] P. G. Krishna, "UPS Parameter Monitoring and Controlling Using IOT and GSM," *Int. J. Pure Appl. Math.*, vol. 116, no. 6, pp. 133–139, 2017.

[4] L. Fu and B. Zhang, "The Design and implementation of a UPS Monitor and Control System.," *Int. Conf. Comput. Sci. Netw. Technol.*, vol. 2, pp. 1322–1325, 2011.

[5] T. S. Gunawan, I. Rahmithul, H. Yaldi, M. Kartiwi, and N. Ismail, "Prototype Design of Smart Home System using Internet of Things," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 1, pp. 107–115, 2017.

[6] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with China Perspective," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, 2014.

[7] N. A. Hidayatullah and D. E. J. Sudirman, "Desain dan aplikasi internet of thing (iot) untuk smart grid power system," 2017.

[8] N. Fajrin, I. Taufik, N. Ismail, L. Kamelia, and M. A. Ramdhani, "On the Design of Watering and Lighting Control Systems for Chrysanthemum Cultivation in Greenhouse Based on Internet of Things," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, 2018.

[9] I. Špeh and I. Heđ, "A Web - Based IoT Solution for Monitoring Data Using MQTT Protocol," *2016 Int. Conf. Smart Syst. Technol.*, pp. 249–253, 2016.

[10] J. E. Luzuriaga, J. C. Cano, C. Calafate, P. Manzoni, M. Perez, and P. Boronat, "Handling mobility in IoT applications using the MQTT protocol," *2015 Internet Technol. Appl. ITA 2015 - Proc. 6th Int. Conf.*, pp. 245–250, 2015.

[11] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things," *Trans. IoT Cloud Comput.*, vol. 3, no. 1, pp. 11–17, 2015.