Development of a MQTT-Modbus Gateway for Interconnection of Field Networks and Applications in Smart Grids

Sivakumar.G[1], Senthil Kumar.S[1], Vijayaragavan S P[1]*, Lakshmi S[1]
[1]Department of Electrical and Electronics Engineering, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India
Vijayaragavan.eee@bharathuniv.ac.in

Abstract. The Industrial Internet of Thing (IIoT) mainly focuses on integrating and utilizing data completely within the production environment. To enhance IIoT the data from various devices and protocols are converted which help in providing a proper communication between the devices. There are still a lot of serial Modbus-based devices in use in industrial settings today. In modern industries serial Modbus-based communication is considered as one of the widely used one for maximum number of devices. Nearly all of the main manufacturers use MODBUS RTU in their new product as it is one of the mostly used industrial protocol for linking devices. The Modbus protocol has master-slave architecture where only device control and others follow. In other hand MQTT is a type of communication method which is well suited for IOT and they work on publish and subscribe model where devices send message(publish) and receive message(subscribe) based on specific topic. In this research, we used a low- cost Raspberry Pi single-board computer along with RS485 add-on board that contains the required ICs for data transfer and together they create a Modbus to MQTT gateway for Industrial IoT cloud applications. In addition,the system uses a Modbus RTU interface to over RS485 to provideefficient two-way communication between field devices and cloud applications.

Keywords:Smart Grids, Internet of Things, MQTT,Industry 4.0, intelligent technologies

1. Introduction

The IoT will change the way we live by making almost everything smart and connected. According to researchers, it will boost the growth of cloud computing because it allows data to be collected,stored, and processed compared to previous methods. Interoperability is a major problem for IIoT, but many serial-based devices that use the Modbus protocol are still in use in industrial settings. Regardless of the manufacturer, Modbus enables communication between a wide variety of devices linked to the same network [1]. Web pages are used to configure the gateway, which is based on an embedded web server. Updates can be made wirelessly when needed. The suggested gateways offer a quick and safe method for developing IIoT applications both today and in the future. Additionally, it can be integrated into a

S. P. Vijayaragavan et al. (eds.), *Proceedings of the Global Conference on Sustainable Energy Systems, Smart Electronics and Intelligent Computing (GCSESEIC 2025)*, Advances in Engineering Research 297,
https://doi.org/10.2991/978-94-6239-654-8_54

system to benefit from the IIoTenvironment while the operational technology that is now in use is retained. Many different sectors and Internet of Things environments use the MQTT protocol these days. Almost all IoT platforms accept MQTT and HTTP as their primary protocols for data transmission. Therefore, IIoT is grown in waythat makes easier for devicesfrom different manufacturers to work together smoothly [2]. To make edge devices IIoT and Industry 4.0 ready, data from various devices and protocols must be converted. The IIoT is advanced in a complementary manner by MQTT and Modbus. A gateway needs to accommodate a large number of protocols. The gateway also handles the data collection and protocol conversion to ensure that the data is in right format for the further process. To solve these problems various studies have tried to connect data from serial- directly to the private or public IoT cloud applications. [3]. In this research, we developed a prototype Modbus/MQTT gateway using a Raspberry Pi which is a low-cost single-board computer and an add-on board for transferring data. This system provides more efficiency when compared to the previously discussed methods and system. It might make the process of connecting industrial facilities' serial equipment to cloud platforms for IIoT applications easier. Data from serial-based devices, like RS-485, is then securely transmitted to publicly and privately hosted cloud services (Amazon AWS [4], Alibaba Cloud, powered by the MQTT messaging protocol and other ready-to-run cloud platforms. This helps industrial plants to analyse their data and encouraging innovation to improve speed in the IIoT. And it can also be added with the existing system without any major changes so the they can make take advantage of the IIoT technology.

It is particularlysuitable forcloud-basedtechnologies and services that deals with data collection and process control. Our analysis showed amismatch in data packet mapping between CoAP and OSGP, which is because both use different applications and protocols. Through this mapping, information can be obtained and sent to SGs via IoT systems used in industrial and residential plants. In this paper we present the integration of the CoAP and OSGP communication protocols as part of a solution called CoAP and OSGP Integration for the Internet of Things (COIIoT). Based on the existingliterature, we knowthat for the first time CoAP and OSGP have been adapted, based on the systematic mapping car. In the first case, MODBUS TCP is used independently to construct the IIoT environment; that is, it is regarded as an IoT protocol. According to the authors, in first scenario the system works effectively when the request-response pattern is required and which makes them suitable for industrial applications [5]. A new paradigm for electrical power systems was presented by SGs, and this technology is being developed to fulfil the growing need for electricity. Efficiency, intelligence, quality, adaptability, dependability, sustainability, and resilience are just a few of the attributes that are inherent to this idea [6]. SGs are sophisticated CPSs that use a number of embedded intelligent technologies to increase power grid safety and dependability.

The authors address the factors to be taken into account when designing communication architecture for smart grids (SGs) in, taking renewable energy sources into account. According to , three factors need to be considered in this situation: Possibility: the grid may employ many IP address schemes and protocol standards; Scalability: it describes both the transmission networks and the data flow from distributed generation; Reliability: this indicates that certain performance indicators, particularly those related to latency and packet loss rate, must be defined[16,19]. Then, in the context of the SGs, the significance of communication infrastructure and associated technologies including communication protocols is quite clear.

2. Materials and Methods

$$Ttotal = TModbus + TGateway + TMQTT \quad (1)$$

$$Reff = N \cdot STcycle \quad (2)$$

The proposed gateway consists of a Raspberry Pi minicomputer and an RS485 add-on board, as shown in the left side of the figure [17]. The RS-485 network shown on the right side of the diagram consists of multiple nodes connected in a bus or line structure with 120-ohm resistors placed at both ends to minimize the signal reflections Equation (1,2). The serial devices like Programmable Logic Controllers (PLCs), Remote Terminal Unit(RTUs) and sensors use Modbus protocol to communicate and share data with each other [18]. By using MQTT as an IIOT protocol and Modbus as a local interface for device management, the system extends the accessibility of device data to cloud applications. MQTT and Modbus work together as a complimentary pair to enhance the overall IoT framework. Th proposed gateway provides a smooth protocol conversion between Modbus RTU and MQTT which make them more table for IIoT applications that depend on MQTT or external cloud services like Alibaba Cloud and Amazon AWS (Fig 1).

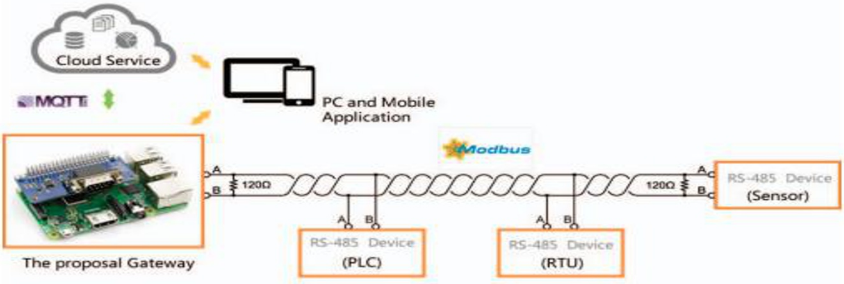

Fig.1. High-level overview of Modbus/MQTT Gateway design.

3. Implementation Details

3.1 Hardware Implementation

The main hardware diagram of the proposed system is shown in Fig 2. It consists of a Raspberry Pi board and an RS485 add-on board where the Raspberry Pi board uses Raspberry Pi (RPi) 3B+ [24] to provide built-in feature like RJ-45 Ethernet, Wi-Fi, Bluetooth and external USB connectivity. The board contains a 64-bit Arm Cortex-A53 (ARMv8) four core CPU clocked at 1.4 GHz and 1 GB LPDDR2 SRAM that provide an efficient platform for knowing advanced on-device data processing capabilities.

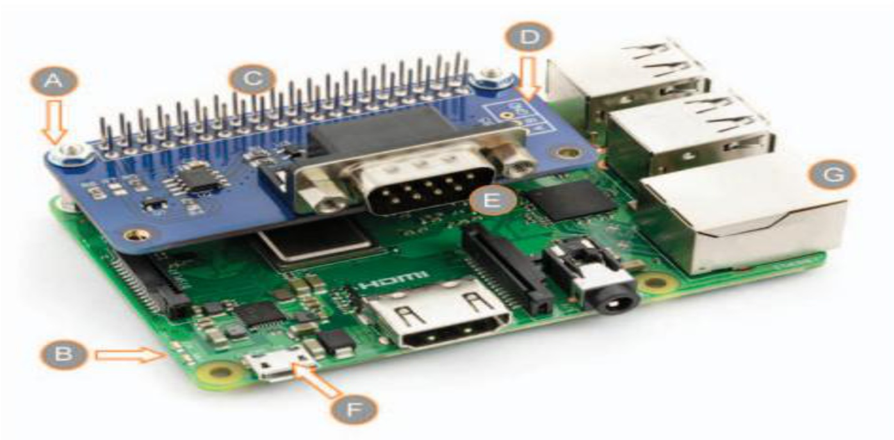

Fig 2: Hardware layout of proposed system

The RS485 add-on board follows the RS485 standard and its connected to the Raspberry Pi via a 40 pin GPIO connector. The RS485 half-duplex communication which is widely used in industries are implemented by using a 9-pin D-type connector and a 3-W external screw terminal that is directly connected to the UART Rx and Tx pins. The Raspberry Pi 3 provides serial communication using two UART controllers namely pl011 UART and tiny UART. In this project, for RS485 communication the mini UART is linked to the 40 pin GPIO header's TXD (GPIO 14) and RXD (GPIO 15). Bluetooth wireless modules is used with Pl011 UART, however any module can be connected to a GPIO port if needed [7]. Using a micro-USB cable and a 5 V power source designed to deliver a minimum of 2,500 mA of electricity, the RPi 3 is externally powered. In terms of protocol analysis, the Modbus and works on master-slave model where the master device sends a request to a slave and waits for a reply. This architecture is particularly effective for traditional multipoint serial networks as it allows the master station to completely control the data transmission. Since Modbus RTU is the most widely used Modbus implementation, this study mainly focuses on the MODBUS RTU fundamentals and application aspects [8].

3.2 Structure of an RTU frame:

- Device Address: It is a valid slave address ranges from 0 to 247 in decimal format. This address helps to identify which slave is responding based on the its unique address in the response frame's address field.

- Function Code: A valid code ranges from 0 to 255 in decimal format. The slave device performs based on the code information.

A lightweight communication protocol called Message Queuing Telemetry Transport (MQTT) was created for SCADA and distant networks. It emphasizes dependable connections with low-level embedded devices with little overhead (2-byte header), as well as the ability to efficiently use network resources and deliver dependable transactions [9]. The publish/subscribe design architecture used by MQTT totally decouples the publisher and subscriber, when a publisher client sends a message to a broker on a designated topic (such as "plc/Coil Status"), the broker replies with a status code and an acknowledgment message. Every subscriber client that is connected to the broker receives messages from the broker and specified monitors topic for incomingdata [10]. Essentially, the subject are message queues that are kept up to date by the broker with subscriptions during a session(Fig 3).

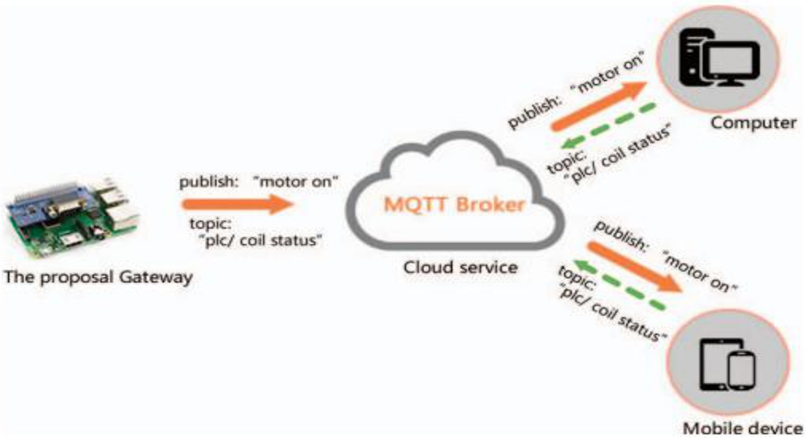

Fig 3: Overview and architecture of MQTT implementation

All messages are mediated by the MQTT server, which is operated on a private or public cloudlike Alibaba Cloud and Amazon AWS [14]. When the Rules Enginewith inthe cloud service identifies a new message published on a Topic [11], acommand is sent tosave the JSON object in the database.The MQTT publisher client on the Raspberry Pi send messages to the arbitrary plc/Coil which contain the motor status values.Any MQTT subscribers who have already subscribed to this topic receive these messages from the broker. The data and the significance of their delivery determine the QoS level to be used. The notion of quality of service (QoS) is made possible via MQTT. In MQTT, there are three QoSlevels [12,15].

4. Results and Discussion:

The results of implementing aweather station networkusing MQTT based network over MODBUS arediscussed in this section. Every weather station sends its data to the Raspberry Pi via MQTT, where the received data is saved in a MySQL database [13]. The received Information is then convertedfrom MQTT to MODBUS and transmitted to SCADA using the settings created in NODERED (fig 4). Numerous testing and sensor calibrations were done in preparation for the installation of the weather stations. Additionally, a data transmitting stage was put in place connecting the SCADA System to the Raspberry Pi Table 1.

Table 1. comparison of Modbus RTU vs MQTT

Feature	**Modbus RTU (RS485)**	**MQTT (IoT Cloud)**
Architecture	Master–Slave	Publish–Subscribe
Communication Medium	Serial RS485	TCP/IP over Ethernet/Wi-Fi
Data Format	Binary Registers	JSON / Key-Value Payloads
Scalability	Limited (≤ 32 devices per bus)	High (thousands of clients)
Reliability	Deterministic polling	QoS levels (0,1,2) for reliability
Suitability	Local industrial networks	Cloud-based IIoT applications

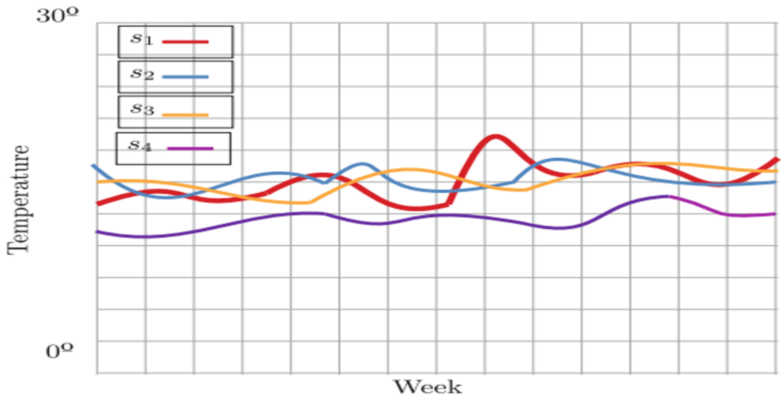

Fig 4: Results data transmission (Temperature)

5. Conclusion:

In this work, a plug-in printed circuit board with integrated circuit chips and components for data and signal processingwere used to develop the prototype. Along with that a low-cost credit card sized mini-computer Raspberry Pi is also used. The complete detail about the various software and hardware design used are discussed. Using Modbus protocol, the data from serial-based devices RS-485 is thensend to public and private cloud service through MQTT messaging protocol in JSON data format. This data can be then viewedanytime and anywhere from standard web browser or even on a smartphone. The gateway works with a wide variety of PLCs and RTUs, both contemporary and vintage. It is simple to acquire data, statistics, alerts, and alarms in real time as the value reaches a certain level. In order to promote innovation and expedite their IIoT initiatives, users can examine it without needing to change newly installed or even existing automation equipment. Although there were a few little issues with this project, the proposed system was only intended to be a gateway prototype for IIoT applications from the start. The gateway should also provide wide and flexible protocol support. In the future, we will concentrate on enabling cloud connectivity using cellular networks such as 3G, 4G or GPRS in areas without Ethernet network. Additionally, we work on connecting a various deviceusing more compatible industrial protocols like PPI and OPC-UA. It will facilitate the smoother integration of cloud, SCADA, IT, and PLC systems. As a result, end users will have greater freedom to install the apps that work best for their companies.

Funding

The authors state that no financial support was received for this research or publication of this paper.

Competing Interest

The authors declare that they have no conflicts of interest.

Reference

1. Encalada, P., Isabel, S.: "Diseño, desarrollo e implementación de una estación meteorológica basada en una red jerárquica de sensores, software libre y sistemas embebidos para la empresa ELECAUSTRO en la Minicentral Gualaceo utilizando comunicación MQTT y MODBUS" thesis (2018)
2. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: "Internet of things (IoT): a vision, architectural elements, and future directions". Future Generation Comput. Syst. 29(7), 1645–1660 (2013)
3. Zhang, J.; Hasandka, A.; Wei, J.; Alam, S.; Elgindy, T.; Florita, A.R.; Hodge, B.M. "Hybrid Communication Architectures for Distributed Smart Grid Applications". Energies, 11, 871. 2018
4. Cao, Z.; O'Rourke, F.; Lyons, W.; Han, X. "Home Energy Management System Incorporating Heat Pump Using Real Measured Data Sensors, 19, 2937 (2019).
5. Leite, J.B.; Mantovani, J.R.S. "Development of a Self-Healing Strategy WithMultiagent Systems for Distribution Networks". IEEE Trans. Smart Grid, 8, 2198–2206. 2017
6. Sacoto-Cabrera, E., Rodriguez-Bustamante, J., Gallegos-Segovia, P., ArevaloQuishpi, G., Le´on-Paredes, G.: "Internet of Things: informatic system for metering with communications MQTT over GPRS for smart meters". In: 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), pp. 1–6. IEEE (2017)
7. Vimos, V., Sacoto, E., Morales, D.:" Conceptual architecture definition: implementation of a network sensor using Arduino devices and multiplatform applications through OPC UA". In: 2016 IEEE International Conference on Automatica (ICAACCA), pp. 1–5. IEEE (2016)
8. Zhang, Y., Mukherjee, M., Wu, C., Zhou, M.T.:" Advanced industrial networks with IoT and big data". Mobile Netw. Appl. 24, 1–3 (2019)
9. Tantitharanukul, N., Osathanunkul, K., Hantrakul, K., Pramokchon, P., Khoenkaw, P.:" MQTT-topics management system for sharing of open data". In: 2017 International Conference on Digital Arts, Media and Technology, pp. 62–65. https://doi.org/10.1109/ICDAMT.2017.7904935, (2019).
10. Lea, P.: "Internet of Things for Architects: Architecting IoT Solutions by Implementing Sensors, Communication Infrastructure, Edge Computing, Analytics, and Security". Packt Publishing, Birmingham (2018)
11. WelleDonker, F., van Loenen, B.: "How to assess the success of the open data ecosystem?" Int. J. Digit. Earth. 10, 284–306 (2017)
12. Vicente-Saez, R., Gustafsson, R., Van den Brande, L.:" The dawn of an open exploration era: Emergent principles and practices of open science and innovation of university research teams in a digital world". Technol. Forecast. Soc. Change. 156, 120037 (2020)

13. Van Schalkwyk, F., Willmers, M., McNaughton, M.: "Viscous open data:The roles of intermediaries in an open data ecosystem". Inf. Technol. Dev. 22, 68–83 (2016)
14. Quiñonez, S., Zapata, A., Canto, P.: "Competencia digital en niños de educación básica del sureste de México." ricsh.org.mx+1RevistaIberoamericana de lasCienciasSociales y Humanísticas" 9(17), 289 – 311 https://doi.org/10.23913/ricsh.v9i17.199(2020)
15. Calyam, P., Rajagopalan, S., Seetharam, S., Selvadhurai, A., Salah, K., Ramnath, R.: VDCAnalyst.: "design and verification of virtual desktop cloud resource allocations". Comput. Networks, 68, 110–122 (2014)
16. S.P.Vijayaragavan, Pati, Debasish; Karthik, B; "Design and Implementation of a Photovoltaic Energy Management System with Battery Backup, Programmable Device Circuits and Systems", v-6, i-7, pp- 197-199, 2014.
17. Jayalakshmi, V., Jafar Ali, S.P.Vijayaragavan, "Implementation of Optimal Power Flow in Microgrids with Energy Storage", International Journal of Advanced Research inElectrical, Electronics and Instrumentation Engineering, v-4, i-2, pp-1073-1079, 2015
18. S.P.Vijayaragavan and T. Jayanthi "Design and simulation of a fuzzy logic controlled hybrid wind- solar system" Malaya Journal of Matematik, Vol. S, No. 2, 3111-3116, https://doi.org/10.26637/MJM0S20/0795(2020)
19. S.P.Vijayaragavan and V. Vijayalakshmi "Solar power and wind power management system using power line communication "Malaya Journal of Matematik, Vol. S, No. 2, 1876-1884https://doi.org/10.26637/MJM0S20/0488, 2020