Jurnal Ilmiah Setrum

Volume 13, No.2, Desember 2024 p-ISSN : 2301-4652 / e-ISSN : 2503-068X

Data Communication between Omron CP1L-E PLC and Arduino Portenta H7 using Modbus TCP Protocol

Nizar Abdul Mathin*, Asep Andang, Firmansyah MSN

Electrical Engineering, Faculty of Engineering, Siliwangi University, Tasikmalaya, Jawa Barat, Indonesia

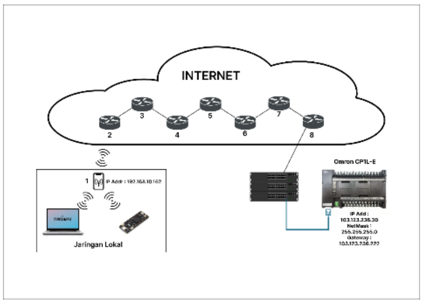
Informasi Artikel

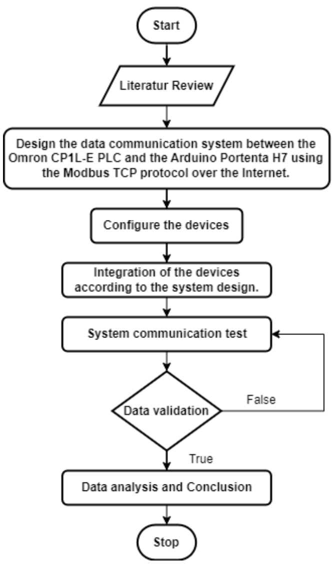
Naskah Diterima : 15 Oktober 2024
Direvisi : 23 November 2024
Disetujui : 28 November 2024

doi: 10.62870/setrum.v13i2.29148

***Korespondensi Penulis :**
nizaram67@gmail.com

Graphical abstract





Abstract

Technology is advancing rapidly in the industrial sector. This encourages PLC, a control device identical to the industrial world, to communicate with other devices, such as Arduino Portenta H7, a microcontroller device. This research aims to analyze data communication between the two devices via the Internet using the Modbus TCP protocol. In this research, Portenta H7 which performs as a client communicates with Omron CP1L-E as a server via the Internet using the Modbus TCP protocol. Data is sent with a variation of the request pause effect time to analyze its effect on communication performance. Quality of Service (QoS) measurements are carried out for 24 hours for each scenario. The results show that increasing the pause time affects communication performance. In the experiment with a 60 ms delay, bottleneck does not happen and the quality of service is in the good category based on TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks) standard.

Keywords: *Modbus TCP, throughput, delay, jitter, packet loss*

Abstrak

Perkembangan teknologi pada dunia industri semakin pesat. Hal ini mendorong PLC yang merupakan perangkat kontrol yang identik dengan dunia industri untuk dapat berkomunikasi dengan perangkat lain, seperti Arduino Portenta H7 yang merupakan perangkat mikrokontroler. Penelitian ini bertujuan untuk menganalisis komunikasi data antara kedua perangkat tersebut melalui internet menggunakan protokol Modbus TCP. Dalam penelitian ini, Portenta H7 yang berperan sebagai client berkomunikasi dengan Omron CP1L-E sebagai server melalui internet menggunakan protokol Modbus TCP. Data dikirim dengan variasi waktu jeda request untuk menganalisis pengaruhnya terhadap kinerja komunikasi. Pengukuran Quality of Service (QoS) dilakukan selama 24 jam untuk setiap skenario. Hasilnya menunjukan peningkatan waktu jeda berpengaruh pada kinerja komunikasi. Pada percobaan dengan jeda 60 ms, bottleneck tidak terjadi dan kualitas layanannya termasuk ke dalam kategori baik berdasarkan standar TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks).

Kata kunci: Modbus TCP, *throughput*, *delay*, *jitter*, *packet loss*

1. INTRODUCTION

As a control device identical to the industrial world, PLCs are now required to communicate with other devices, such as Arduino, a microcontroller [1], [2]. Arduino Portenta H7 is a variant issued by the Arduino AG Company. This device is produced for systems with high-performance needs, such as in the industrial domain [3], [4].

However, communication between the two devices still has challenges that must be overcome. One of them is the difference in communication protocols frequently used by both devices [5]. PLC commonly uses the Modbus protocol which is specialized for industry, while Arduino frequently uses communication protocols such as UART, I2C, SPI, etc.

For this reason, efforts are made to utilize the Modbus TCP communication protocol in this research. This protocol is a development form of the Modbus Protocol that can run over TCP/IP networks [6], [7]. This research is proposed to analyze the performance of the protocol on Portenta H7 that communicates with the Omron CP1L PLC. Testing is done by varying the time pause between requests on Portenta H7, to determine its effect on service quality based on standards from TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks).

2. LITERATURE REVIEW AND METHODOLOGY

2.1 Literature Review

In this research, some of the main theories that form the basis are the theory of the Omron CP1L-E PLC, Arduino Portenta H7, Modbus TCP Protocol, and Wireshark. The following is a description of the theories that support this research.

A. OMRON CP1L-E Programmable Logic Control (PLC)

Programmable Logic Control (PLC) is an electrical device that works digitally, has programmable memory, and stores commands to perform specific tasks such as logic, timing, counting, arithmetic, and sequencing to control machines or processes through analog or digital input/output modules [8]. The basic parts of a PLC are as follows:

1. Central Processing Unit (CPU), to retrieve instructions from memory, decode them, and execute the instructions. It performs systematic and logical functions and detects signals outside of the CPU.
2. Programming Memory (PM), in charge of storing instructions, programs, and data
3. Programming Device (PD), is a device used to type, enter, modify, and monitor programs in PLC memory.
4. Input/Output module, serves as a link between the PLC and the outer I/O assembly.

OMRON SYSMAC CP1L-E PLC is one of the PLC products of the OMRON Company. This PLC is designed with a high process speed, so the application of this PLC can be used in systems that require a fast response [9]. This PLC is equipped with an easy-to-use interface, making it easy for users to learn and operate. It is also equipped with serial and Ethernet communication ports for connectivity.

B. ARDUINO PORTENTA H7

Arduino Portenta H7 is a microcontroller that uses dual-core STM32H747 as the main processor [3], [4]. This processor has two cores, namely Cortex M7 at 480 MHz and Cortex M4 at 240 MHz [10]. The dual-core system is an advantage that Arduino Portenta H7 has. This makes the Arduino Portenta H7 able to run tasks in parallel, this is the advantage of the Portenta H7 compared to other Arduino microcontrollers [3]. In addition, this microcontroller is designed to be applied in high-end industrial systems, PLC systems, laboratory needs, and robotic controllers.

C. MODBUS TCP PROTOCOL

Modbus is an international standard data communication protocol published by the Modicon Company in 1979 and is often used in Programmable logic control (PLC). There are two classes of devices in this protocol, namely Modbus Master and Modbus Slave. The master has an active role in initiating communication either by reading data, writing data, knowing the status of the slave, and sending requests or queries [11]. The query contains function code and query data. While the slave only responds to requests from the master. If the query data from the master is received by the slave without error, the slave will send its response to the master. The response is in the form of a function code and response data. If there is an error in the query received by the slave, the slave will send an error message to the master consisting of an exception function code and an exception code.

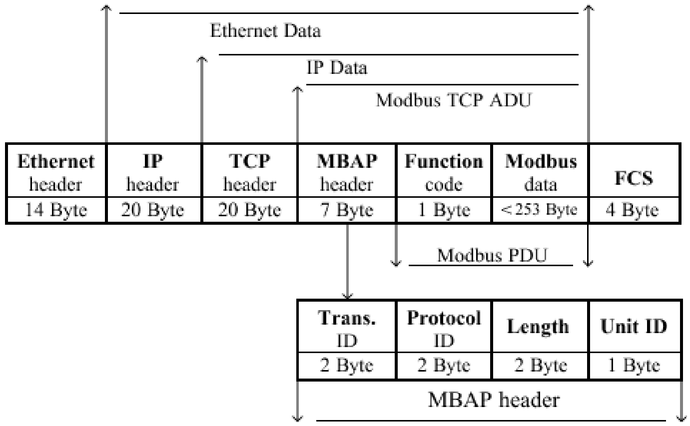


Figure 1. The frame structure of Modbus TCP

Figure 1. illustrates the frame structure of Modbus TCP. Modbus TCP is the Modbus RTU protocol that uses the TCP interface that runs on Ethernet media and runs at the TCP/IP application layer [6], [7]. This protocol combines a physical network in the form of Ethernet with a networking standard (TCP/IP) and a standard method for representing data (Modbus as an Application protocol). Modbus TCP collaborates standard Modbus data frames into TCP frames without checksums. Then the Modbus address frame is replaced with the unit identification in Modbus TCP and becomes part of the Modbus Application Protocol (MBAP) header. The data packet is then added to an IP address and sent from the client to the server according to the given IP address.

Modbus TCP uses a Client and server network topology that must be connected during the data exchange process, this exchange is carried out via port 502. IEEE 802.3 Ethernet is a commonly applied network protocol and is an open standard that is supported by many manufacturers and whose infrastructure is widely available. Therefore, TCP/IP is used as the basis for access to the World Wide Web. Modbus TCP shares the same physical and data link layers as IEEE 802.3 Ethernet and uses the same TCP/IP protocol suite [12].

When data communication on Modbus TCP occurs, there is a change in the data structure at each layer passed. This change process is called encapsulation [6], [12]. At the application layer, the Modbus protocol builds a frame format called application data unit (ADU) which consists of MBAP and Modbus Protocol Data Unit. According to [12], at the transport layer, the ADU is added to a TCP header so that it can be sent over the TCP network. The frame at this layer is called TCP Frame. At the network layer, the TCP Frame is added with an IP Header for data addressing so that it can go to the desired address, the frame in this process is called an IP Frame. At the data link layer, the IP Frame is added with the Ethernet Header and Frame Check Sequence (FCS), and its name is changed to Ethernet Frame. At the physical layer, Ethernet frames are sent through physical networks such as cable, fiber optic, and similar transmission media.

D. Wireshark

Wireshark is a measurement tool used to measure parameters in data communication on a network. Wireshark is software that functions as a network analyzer. Wireshark works by capturing data packets traveling in the system [13], [14], [15].

2.2 Methodology

A. Research Flowchart

The research flowchart describes the flow of research on designing a data communication system model between the Omron CP1L-E PLC and Arduino Portenta H7 using the Modbus TCP protocol via the internet media. The flow can provide an overview of the research methodology used in this study.

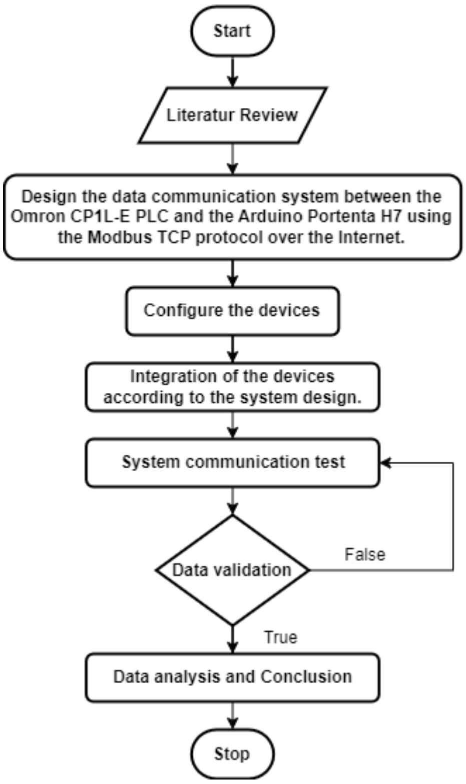


Figure 2. Research Flowchart

Figure 2. Research Flowchart t shows a methodical way to create an Internet-based Modbus TCP protocol data transmission system between an Arduino Portenta H7 and an Omron CP1L-E PLC. To collect pertinent data, the procedure starts with a Start node and moves on to a Literature Review phase. The flowchart then describes the communication system's actual design. Device configuration, device integration in accordance with the system design, and system communication test procedures come next.If the communication system indicates that both devices can communicate for 24 hours or until one of the devices is unable to continue communicating because of an error brought on by a bottleneck before 24 hours, the data is deemed legitimate. Following data validation, which validates the system, data analysis and conclusion are the last steps that lead to the Stop node.

B. Experiment Scenario

Data gathering begins with setting up devices such as Omron CP1L-E, Portenta H7, and a PC with Wireshark. Wireshark is used to monitor data communication between Portenta H7 which sends a request with function code 0x16 (write multiple registers) to write dummy data to 123 register addresses on Omron CP1L via the Internet. Data validation is carried out after the test runs for 24 hours with variations in the pause between requests of 0 ms, 15 ms, 30 ms, 45 ms, 55 ms, 60 ms, and 65 ms to determine the performance characteristics of the system based on QoS standards according to TIPHON. The topology of the system is described in Figure 3. The system topology.

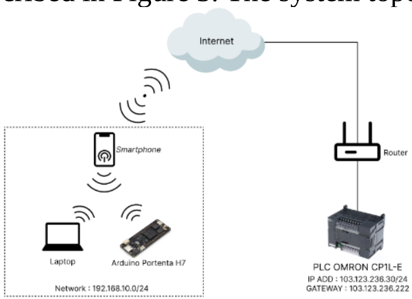


Figure 3. The system topology

C. Quality of Service (QoS) Parameters Based on TIPHON Standard

1) Throughput

Throughput is the actual data transfer rate where the total number of packet arrivals in a certain time interval is divided by that time interval [13], [15], [16]. The categories of Throughput are described in Table 1. The categories of throughput below.

Table 1. The categories of throughput

Category	***Throughput***	***Index***
Very good	*> 2.1 Mbps*	*4*
Good	*1.2 - 2.1 Mbps*	*3*
Medium	*338 - 1200 Kbps*	*2*
Bad	*0 - 338 Kbps*	*1*

Source: [15]

2) Delay

Delay (Latency) is the time it takes for data to travel the distance from the source device to the destination device [15]. In this study, the delay used is the Round Trip Time (RTT) delay. Because the Modbus TCP work system is based on Request and Response, the RTT delay can be used to assess how fast the responsiveness of communication is [17]. The delay categories are shown in Table 2. The categories of delay below.

Table 2. The categories of delay

Category	***Delay (ms)***	***Index***
Very good	*< 150*	*4*
Good	*150 - 300*	*3*
Medium	*300 - 450*	*2*
Bad	*> 450*	*1*

Source: [15]

3) Jitter

Jitter is a value that shows the quantity of variance or fluctuation of delay in data communication [13], [15], [16]. The jitter categories are shown in Table 3. The categories of jitter below.

Table 3. The categories of jitter

Category	***Jitter (ms)***	***Index***
Very good	*0*	*4*
Good	*0 - 75*	*3*
Medium	*75 - 125*	*2*
Bad	*125 - 255*	*1*

Source: [15]

4) Packet Loss

Packet Loss is a condition that represents the total number of packets lost in data communication [13], [15], [16]. Packet loss is the percentage between packets received and packets sent. Packet Loss categories are shown in Table 4. The categories of packet loss below.

Table 4. The categories of packet loss

Category	***Packet Loss (%)***	***Index***
Very good	*< 3*	*4*
Good	*3-15*	*3*
Medium	*15-25*	*2*
Bad	*> 25*	*1*

Source: [15]

3. RESULT

Communication performance analysis on the Modbus TCP communication system between the Omron CP1L-E PLC and Arduino Portenta H7 via the Internet is carried out based on standards from TIPHON (Telecommunications and Internet Protocol Harmonization Telecommunications Standards Institute).

3.1 Communication Testing

In testing the communication between Portenta H7 and Omron CP1L-E via the Internet. The effect of variation on the length of observation time is different. This is because, in some experiments with a short time pause, it tends to bottleneck and cause communication to stop. This is explained in Table 5.

Table 5. Results of the communication test

Pause Request	***Observation Time (s)***	**Description**
0 ms	198.860855	bottleneck
15 ms	237.297724	bottleneck
30 ms	23026.79154	bottleneck
45 m	26152.39305	bottleneck
55 ms	67526.35867	bottleneck
60 ms	90008.56824	no bottleneck
65 ms	86402.64135	no bottleneck

From the Table 5. Results of the communication test above, it can be seen that the observation time increases linearly with the request time pause. A time pause that is too small, has a short experiment time, this can occur because in experiments with low time pause, the device must process and respond to requests with high frequency. This causes a high workload on the device causes a build-up of requests and eventually causes bottlenecks.

3.2 Throughput

Throughput is a parameter that refers to the amount of data sent and received by the client and server in a certain period. Usually, throughput has units of bits per second (bps) or bytes per second (Bps). Data from throughput testing is shown in Table 6.

Table 6. Results of the throughput test

Pause Request	**Observation Time (s)**	**Amount of Data (Bytes)**	**Throughput (Bps)**
0 ms	198.860855	378891	1905
15 ms	237.297724	576904	2431
30 ms	23026.79154	57168739	2483
45 m	26152.39305	62370254	2385
55 ms	67526.35867	153788729	2277
60 ms	90008.56824	188211998	2091
65 ms	86402.64135	132317506	1531

Based on the data from Table 6. Results of the throughput test, it is known that the optimal throughput is achieved in experiments with a time pause of 15 ms, 30 ms, and 45 ms. With the largest value occurring in testing with a pause of 30 ms with a value of 2483 Bps. The addition of pause time causes the throughput to decrease, this is because the increasing pause time between requests, the number of packets communicated will be more limited. The decrease in throughput occurred in tests with 55 ms, 60 ms, and 65 ms pauses. In the 65 ms test, the throughput has the lowest value of the other tests, which is 1531 Bps. While in testing without using a pause time, the throughput value is low because the Omron CP1L-E PLC experiences a bottleneck due to sending requests too quickly.

If in the ideal condition calculation, the request and response data packet size is 379 bytes, the average test delay value is implemented, and the pause time between requests is added. Then the ideal throughput can be calculated, and the results are shown in Table 7.

Table 7. The ideal throughput

Pause time	**Average delay (s)**	**Throughput (Bps)** $= \frac{\mathbf{379\ byte}}{\mathbf{Rata-rata\ delay+waktu\ jeda}}$	**Testing throughput (Bps)**
0ms	0.070632305	5366	1905
15 ms	0.081814315	3915	2431
30 ms	0.075783858	3583	2483
45 ms	0.087935572	2851	2385
55 ms	0.080291914	2801	2277
60 ms	0.091254981	2506	2091
65 ms	0.120827875	2040	1531

Based on the comparison from Table 6 Results of the throughput test and Table 7 The ideal throughput, it can be concluded that in each test, the test throughput value is smaller than the ideal throughput. This can occur due to other factors that affect the performance of the communication. One of them is the state of the network which is represented by jitter.

3.3 Delay / Latency

In this test, the delay value calculated is the Round Trip Time (RTT) delay. This delay is calculated because Modbus TCP has a request-response work system. RTT measures the total time from the request sent until the response is received. The delay data from the previous test is described in Table 8.

Table 8. Results of the delay test

Pause Time	**Observation Time (s)**	**Average Delay (s)**
0 ms	198.860855	0.070632305
15 ms	237.297724	0.081814315
30 ms	23026.79154	0.075783858
45 ms	26152.39305	0.087935572
55 ms	67526.35867	0.080291914
60 ms	90008.56824	0.091254981
65 ms	86402.64135	0.120827875

Based on Table 8. Result of the delay test, it is known that the value of adding a pause to the experiment does not linearly affect the delay. This can be seen in the experiment with a pause of 30 ms which is worth 0.075783858 s, the delay value in the experiment is less than the two previous experiments which rose from 0.070632305 s to 0.081814315 s. This can occur due to different network conditions in each test. In the TIPHON standard, the delay is in excellent condition if it is less than 150 ms.

3.4 Jitter

Jitter is calculated based on the time variability of the RTT delay of different packets. Jitter can also be interpreted as the value of the communication time inconsistency of one communication cycle. Jitter data from previous tests can be seen in Table 9. Results of the jitter test.

Table 9. Results of the jitter test

Pause Time	**Observation Time (s)**	**Average Jitter (s)**
0 ms	198.860855	0.009517499

Pause Time	Observation Time (s)	Average Jitter (s)
15 ms	237.297724	0.011412981
30 ms	23026.79154	0.00935813
45 ms	26152.39305	0.013535751
55 ms	67526.35867	0.027380233
60 ms	90008.56824	0.01876482
65 ms	86402.64135	0.057242308

3.5 Packet Loss

Packet loss testing is carried out to determine the reliability of communication. The high number of Packet loss indicates that communication is not going well because the communicated data packets are lost in the communication process. Packet loss test results can be seen in Table 10.

Table 10. Results of the packet loss test

Pause time	Observation Time (s)	Request-Response Count	Lost Package	Packet Loss (%)
0 ms	198.860855	893	0	0
15 ms	237.297724	1523	0	0
30 ms	23026.79154	150841	0	0
45 ms	26152.39305	164557	14	0.00425
55 ms	67526.35867	405753	61	0.00752
60 ms	90008.56824	496592	41	0.00413
65 ms	86402.64135	362921	61	0.00840

Based on Table 10. Result of the packet lost test, it is known that no packet loss occurs during testing up to testing with a time pause of 30 ms. However, in experiments with a time pause of more than 30 ms, all of them have packet loss values. The largest packet loss occurs in tests with a time pause of 65 ms with a percentage of 0.0084%. The existence of packet loss can be caused by the unstable network used. Even so, all packet loss values in the experiment still have a very small value. Based on TIPHON standards, communication with a packet loss value of less than 3% is included in the excellent category with an index value of 4.

3.6 Quality of Service

After testing all the previous parameters, the next step is to summarize the results of all the previous tests. This is done to provide an overview of the performance of Modbus TCP communication over the internet in several scenarios tested, both tests using pauses and without pause time between requests. The results of the overall test are contained in Table 11.

Table 11. The results of the overall test

<table>
<tr><th rowspan="2">Parameters</th><th colspan="7">Pause Time</th><th rowspan="2">Index</th></tr>
<tr><th>0 ms</th><th>15 ms</th><th>30 ms</th><th>45 ms</th><th>55 ms</th><th>60 ms</th><th>65 ms</th></tr>
<tr><td>Delay (ms)</td><td>70,6</td><td>81,8</td><td>75,8</td><td>87,9</td><td>80,3</td><td>91,3</td><td>120,8</td><td>4</td></tr>
<tr><td>Jitter (ms)</td><td>9,5</td><td>11,4</td><td>9,4</td><td>13,5</td><td>27,4</td><td>18,8</td><td>57,2</td><td>3</td></tr>
<tr><td>Packet Loss (%)</td><td>0</td><td>0</td><td>0</td><td>0.0043</td><td>0.0075</td><td>0.0041</td><td>0.0084</td><td>4</td></tr>
<tr><td colspan="8">Average</td><td>3,667</td></tr>
</table>

From the Table 11. The results of the overall test above, it is known that the average index value is 3.667. This illustrates that the performance of Modbus TCP communication in both test conditions falls into the good category based on the QoS standard value from TIPHON because it is in the range of 3 - 3.79 which is in the Good category. The throughput value is not calculated due to its lack of relevance to the test which focuses on the Modbus TCP protocol.

4. CONCLUSION

Data communication between the Omron CP1L-E PLC and Arduino Portenta H7 can run using the Modbus TCP protocol via the Internet. Communication can run according to the design. Based on the experiments in this study, the most effective time pause is 60 ms. All parameters from the experiment are quite good, and no bottleneck occurs. In addition, the QoS parameters at this time delay fall into the good category based on the standards of TIPHON.

REFERENCES

[1] A. I. Marosan, G. Constantin, A. Barsan, M. Crenganis, and C. Girjob, "Creating an ethernet communication between a Simatic S7-1200 PLC and Arduino Mega for an omnidirectional mobile platform and industrial equipment," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 968, no. 1, 2020, doi: 10.1088/1757-899X/968/1/012022.

[2] G. Mageshkumar, N. Kasthuri, K. S. Tamilselvan, S. Suthagar, and A. Sharmila, "Design of industrial data monitoring device using IoT through modbus protocol," *Int. J. Sci. Technol. Res.*, vol. 9, no. 1, pp. 1392–1396, 2020.

[3] N. L. Gimenez, F. Freitag, J. K. Lee, and H. Vandierendonck, "Comparison of Two Microcontroller Boards for On-Device Model Training in a Keyword Spotting Task," *2022 11th Mediterr. Conf. Embed. Comput. MECO 2022*, 2022, doi: 10.1109/MECO55406.2022.9797171.

[4] D. L. Pino, F. Freitag, and M. Selimi, "Designing a Double LoRa Connectivity for the Arduino Portenta H7," *IEEE Work. Local Metrop. Area Networks*, vol. 2022-July, 2022, doi: 10.1109/LANMAN54755.2022.9820092.

[5] D. R. R. Ruwahida, I. Rachman, H. A. Widodo, R. Y. Adhitya, and Y. Irawan, "Sistem Komunikasi Mikrokontroler dan PLC Berbasis Komunikasi Serial Host Link dan Protokol C-Command RS232," *Infotekmesin*, vol. 14, no. 2, pp. 354–361, 2023, doi: 10.35970/infotekmesin.v14i2.1924.

[6] F. Katulic, D. Sumina, S. Gros, and I. Erceg, "Protecting Modbus/TCP-Based Industrial Automation and Control Systems Using Message Authentication Codes," *IEEE Access*, vol. 11, no. April, pp. 47007–47023, 2023, doi: 10.1109/ACCESS.2023.3275443.

[7] J. Fauzy Kusuma, M. Rifa'i, and I. Saukani, "Implementasi Protokol Komunikasi Modbus Untuk Mini Scada Pada Plant Pengisian Serbuk Temulawak," *Multidiciplinary Sci. J.*, vol. 2, no. 5, pp. 314–326, 2024.

[8] T. M. Sages, *Programmable Logic Controllers*. 2017. doi: 10.1115/CEC1991-3705.

[9] OMRON CORPORATION, "OPERATION MANUAL CP1L CPU Unit," p. 830, 2014, [Online]. Available: www.ia.omron.com

[10] Arduino, "Arduino® Portenta H7 Collective Datasheet," pp. 1–24, 2024, [Online]. Available: https://docs.arduino.cc/resources/datasheets/ABX00042-ABX00045-ABX00046-datasheet.pdf

[11] E. Gamess, B. Smith, and G. Francia, "Performance evaluation of modbus TCP in normal operation and under a distributed denial of service attack," *Int. J. Comput. Networks Commun.*, vol. 12, no. 2, pp. 1–21, 2020, doi: 10.5121/ijcnc.2020.12201.

[12] T. Tosin, "Perancangan dan Implementasi Komunikasi RS-485 Menggunakan Protokol Modbus RTU dan Modbus TCP Pada Sistem Pick-By-Light," *Komputika J. Sist. Komput.*, vol. 10, no. 1, pp. 85–91, 2021, doi: 10.34010/komputika.v10i1.3557.

[13] J. . I. M. A. P *et al.*, "Video Streaming QoS Analysis with TIPHON Standard NDNts," *2023 9th Int. Conf. Wirel. Telemat.*, pp. 1–4, 2023, doi: 10.1109/ICWT58823.2023.10335399.

[14] D. A. Rachman, Y. Muhyidin, and M. A. Sunandar, "Analysis Quality of Service of Internet Network Fiber To the Home Service Pt. Xyz Using Wireshark," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3s1, pp. 997–1006, 2023, doi: 10.23960/jitet.v11i3s1.3436.

[15] I. S. N. Nisa, Rahmat Miyarno Saputro, Tegar Fatwa Nugroho, and Alfirna Rizqi Lahitani, "Analisis Quality of Service (QoS) Menggunakan Standar Parameter Tiphon pada Jaringan Internet Berbasis Wi-Fi Kampus 1 Unjaya," *Teknomatika J. Inform. dan Komput.*, vol. 17, no. 1, pp. 1–9, 2024, doi: 10.30989/teknomatika.v17i1.1307.

[16] F. D. Aryanto, H. Wijanarko, D. S. Pamungkas, and I. Z. Putra, “Implementasi Smart Relay dan Protokol Modbus TCP pada Sistem Darurat di Perumahan,” *J. Appl. Sci. Electr. Eng. Comput. Technol.*, vol. 2, no. 2, pp. 1–6, 2021, doi: 10.30871/aseect.v2i2.3248.

[17] A. H. Putranto, Ichsan Edi , Maman Abdurohman, “Implementasi Dan Analisis Protokol Modbuus TCP Pada Smart Building Berbasis OpenMTC,” *Telkkom Univ.*, 2016.