EMITOR: Jurnal Teknik Elektro

Implementation of MQTT Protocol on ESP32-Based OEE Analysis Development Board

Amir Akbar Wicaksono, Yuli Kurnia Ningsih[*], Indra Surjati
Magister Teknik Elektro/Fakultas Teknologi Industri – Universitas Trisakti
Jakarta, Indonesia
[*]yuli_kn@trisakti.ac.id

Abstract – *The transition to Industry 4.0 requires major investments in devices and mechanisms that enable interconnectivity between people, machines, and processes. One important concept related to Industry 4.0 is the so-called Industrial Internet Of Things or IIoT. The application of IIoT in the industrial scope is the measurement of Overall Equipment Effectivness (OEE) through the IoT paradigma. Generally, OEE measurements are carried out manually by production operators on the machine being measured, and data processing is carried out by supervision manually as well to then analyze the OEE value of the machine being measured. In this research, an ESP32-based OEE Analysis Development Board with MQTT protocol is proposed to replace the manual OEE measurement process. The results of direct implementation on the production floor show that the ESP32-based OEE Analysis Development Board with MQTT protocol can be used as an alternative to OEE measurement with a maximum error value on OEE measurement of 16%.*

Keywords – *Industri 4.0; Industrial Internet Of Thing; Overall Equipment Efectivness; ESP32; MQTT.*

I. Introduction

THE Fourth Industrial Revolution, or Industry 4.0, represents the current evolution of production systems following the merger of industrial automation and information technology. Industry 4.0 technological innovations feature the integration of manufacturing systems [1], real-time management of product lifecycles, and the decentralization of Information Technology (IT) resources [2]. One of the critical concepts related to Industry 4.0 is the so-called Industrial Internet of Things (IIoT) [3], which refers to the use of interconnected machines and automation devices equipped with sensors in industrial environments [4]. The devices and equipment used in Industry 4.0 primarily consist of proprietary systems [5] owned by vendors [6] or services from external companies, which are typically expensive to implement and have various communication protocols [7] that are difficult to standardize [8].

In several previous studies, designs related to OEE have been implemented using low-cost devices. Gun Maulanan [9, 10] developed a prototype of a performance monitoring system for press machines based on the Internet of Things (IoT) to enhance machine performance with optimal production output. The system overview includes three Arduino Nano units as slaves directly connected to multiple sensors, each slave then connects to a Raspberry Pi as the master, which collects and displays data in real-time.

Halldórsson [11] designed a system using a Raspberry Pi as a data collector, receiving inputs from PLC logic, and forwarding the data to a cloud server for visualization using various software tools to measure OEE. Mastang [12] provided knowledge on the basic application of OEE measurement that can be easily implemented and efficiently used with a Raspberry Pi.

Furthermore, Herrero [2, 4] mitigated the efficiency utilization of Raspberry Pi as a low-cost device in direct production line implementation. When implemented, an error signal caused by electromagnetic interference was identified. The study introduced a non-physical method to handle electromagnetic interference, thus maximizing the accuracy of OEE measurement readings.

In another study, Kong [13] introduced a new device for calculating OEE using an ESP32 with the standard web protocol HTTP. However, the prototype design did not address the accuracy of the data produced, leaving the efficiency of using ESP32 for actual OEE

The manuscript was received on January 9, 2024, revised on June 29, 2024, and published online on July 26, 2024. Emitor is a Journal of Electrical Engineering at Universitas Muhammadiyah Surakarta with ISSN (Print) 1411 – 8890 and ISSN (Online) 2541 – 4518, holding Sinta 3 accreditation. It is accessible at https://journals2.ums.ac.id/index.php/emitor/index.

measurement in production line implementation uncertain.

The final reference for the design of the ESP32-based OEE Analysis Development Board with MQTT protocol implementation is the study by Surya [14], which compared the application of the MQTT protocol and the HTTP protocol. The results of this study indicated that the MQTT protocol has a faster data transfer capability compared to the HTTP protocol, capable of transferring data six times more efficiently than HTTP.

II. Research Methods

The primary goal of this research is to replace the manual data collection process on machines performed by operators and to eliminate the manual OEE calculation process, which is then replaced by a user interface. By designing this low-cost device, it is expected that industries will no longer rely on external services or vendors whose devices are difficult to standardize. The stages of design and implementation are as follows: Literature Study, System Architecture, Pre-Implementation Testing, Actual Implementation, and Problem Handling in Actual Implementation. Before the design process, a literature study was conducted from several journal sources related to this research. The details of the required references are as follows:

1. Overall Equipment Effectiveness (OEE) [15] is a metric useful for accurately estimating industrial productivity. This metric can be measured in real-time through the IoT paradigm, where smart devices can collect important data, helping stakeholders gather useful information to make informed decisions to enhance productivity while reducing costs.
2. Low Cost Device: is a solution implemented in the initial design of a system that has the same capabilities as devices typically used by large companies [2]. With low financial investment, it is expected to work optimally despite certain limitations. There are many low-cost devices available in the market, such as Particle.io, ESP8266, Arduino, and Raspberry Pi [16]. In this research, the low-cost device to be used is the ESP32 [17].
3. PostgreSQL: is a database management system that supports various SQL standards and offers many modern features [18].
4. Node-Red: is a development environment based on Node.js and JavaScript developed by IBM engineers, best suited for developing Internet of Things (IoT) systems [19].
5. Grafana: is the main tool used for data visualization [20].
6. MQTT Communication Protocol: is a communication protocol based on clients publishing/subscribing to topics from a broker. It is designed for implementation on devices with limitations, low bandwidth, and connected to unreliable networks [21].
7. EMQX Broker: is open-source software that functions to receive and transmit messages sent by clients [22].

i. System Architecture

The System Architecture used in the design of this tool can be seen in Figure 1 as follows:

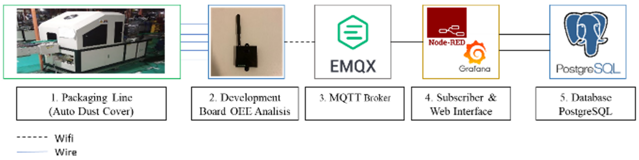

EMQX
Node-RED
Grafana
PostgreSQL
1. Packaging Line (Auto Dust Cover)
2. Development Board OEE Analisis
3. MQTT Broker
4. Subscriber & Web Interface
5. Database PostgreSQL
Wifi
Wire

Figure 1: System Architecture of ESP32-Based OEE Analysis Development Board

The explanation related to the overall proposed system is as follows:

1. Packaging Line: The actual line where the designed tool will be implemented.
2. Development Board OEE Analysis: Data Logger (Start, Stop, Qty OK & NG conditions).
3. MQTT Broker: using EMQX software.
4. Node-red & Grafana: Subscriber and User Interface.
5. PostgreSQL: Storing actual reading data in real-time.

ii. Pre-Implementation Testing

In the industrial scope, one of the implementations related to IIoT is the application of Overall Equipment Effectiveness (OEE) measurement. OEE is a metric useful for measuring the productivity of machines in industries. This measurement process is done manually by production operators collecting data, which is then processed to yield OEE measurement results. In the OEE measurement process, there are three important points: Performance, Availability, and Quality. OEE can be calculated using Equation 1 [4]:

$$\text{OEE} = \left(\frac{\text{Product CT} \times \text{OK Products}}{\text{Loading Time}} \right) \times 100\% \quad (1)$$

In IIoT, this calculation process can be replaced with the IoT paradigm. The data collection process performed by operators can be replaced by IoT devices that log data directly from the machine for start on/off status, stop on/off status, OK product count, and NG product count. Then, the OEE calculation can be performed directly by the user interface utilizing an algorithm similar to the OEE calculation formula.

Based on the designed system architecture, the Development Board OEE Analysis has 4 input pins activated as data loggers to collect data from the machine in real-time. Each input is directly connected to respective machine inputs to continuously collect data. Figure 2 explains the details regarding the OEE Analysis Development Board wiring and data logger targets on the machine where OEE measurements will be implemented using the IoT paradigm.

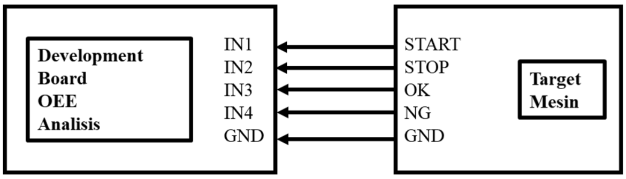


Figure 2: Wiring of the Development Board OEE Analysis with the target data logger on the machine

In the pre-implementation testing process, a laptop is used as a server running EMQX software as the broker, Node-red and Grafana as the subscriber and user interface, and PostgreSQL as the database collection on the server. Each software runs on the following links:

1. localhost:18083 for EMQX broker
2. localhost:1880/ui for UI on Node-red
3. localhost:3000/PK01 for UI on Grafana
4. localhost:5050 for PostgreSQL control

Figure 3 shows the EMQX and Node-Red interfaces, where the EMQX dashboard can be seen in Figure 3(a), and the user interface dashboard on Node-Red can be seen in Figure 3(b).

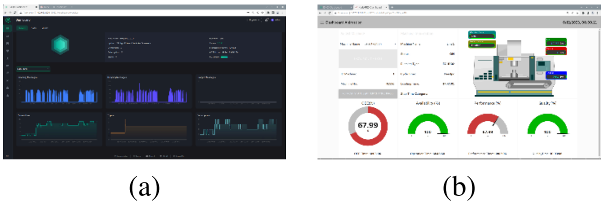


Figure 3: EMQX and Node-Red Interfaces (a) EMQX broker dashboard and (b) UI on Node-Red

Figure 4 shows the Grafana and PostgreSQL interfaces on pgAdmin, were the OEE measurement monitoring dashboard can be seen in Figure 4(a), and the PostgreSQL database control on pgAdmin can be seen in Figure 4(b). The detailed image of the overall system architecture during testing can be seen in Figure 5.

Next, testing is conducted on each input of the Development Board OEE Analysis by providing a voltage of 5-24V. The voltage is then processed by the algorithm built on the Development Board OEE Analysis

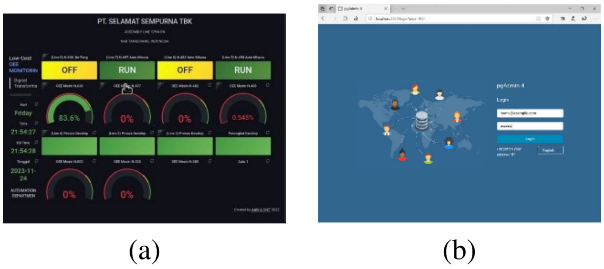


Figure 4: Grafana and PostgreSQL Interfaces (a) UI on Grafana (b) PostgreSQL on pgAdmin

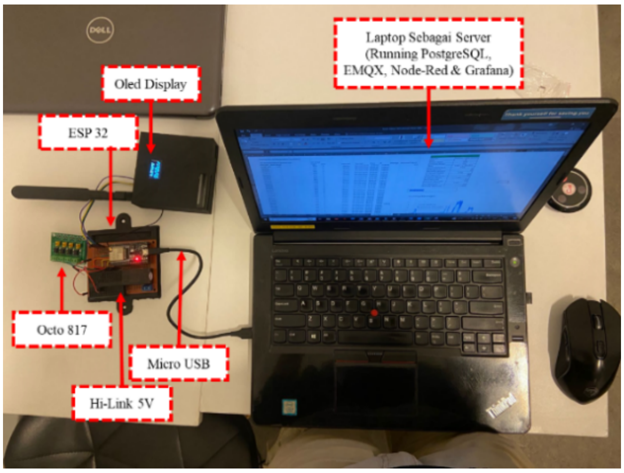

Laptop Sebagai Server (Running PostgreSQL, EMQX, Node-Red & Grafana)
Oled Display
ESP 32
Octo 817
Micro USB
Hi-Link 5V

Figure 5: Tool testing before implementation

to publish payloads according to the topic to the broker. The broker then forwards the payload from the published topic to the subscriber to be processed according to the algorithm built on the Node-red software. The detailed workflow of the publish and subscribe process in the MQTT protocol can be seen in Figure 6.

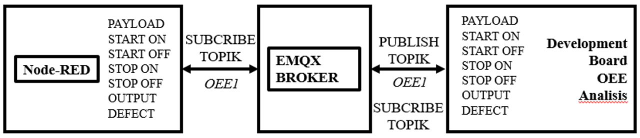


Figure 6: Publish and Subscribe in the MQTT protocol

Table 1 shows the results of testing input 1 (IN1) with payload START ON and START OFF.

Table 1: Test results for input 1 (IN1) by providing a voltage of 5-24V as a signal to publish data

Time Sampling		#Real Signal	#Store Signal	Error
Start	Stop			
08:04:06	09:10:11	2	2	0
09:12:18	10:05:06	2	2	0
10:31:18	11:32:49	2	2	0
13:56:12	14:48:18	2	2	0
15:10:18	16:30:08	2	2	0

Table 2 shows the results of testing input 2 (IN2) with payload STOP ON and STOP OFF.

Table 2: Test results for input 2 (IN2) by providing a voltage of 5-24V as a signal to publish data

Time Sampling		#Real Signal	#Store Signal	Error
Start	Stop			
08:34:06	08:46:11	2	2	0
09:15:18	09:20:06	2	2	0
10:41:18	11:02:49	2	2	0
14:26:12	14:28:18	2	2	0
15:40:18	15:50:08	2	2	0

Table 3 shows the results of testing input 3 (IN3) with payload OUTPUT. There is an error of -1% in the test tool with an input cycle time of 1.8 seconds and 2 seconds. This error is the comparison between the real signal input and the store signal when the signal is received by the PostgreSQL database. The error value in the store signal can affect the final OEE measurement result.

Table 3: Test results for input 3 (IN3) by providing a voltage of 5-24V as a signal to publish data

CT (Second)	Real Signal	Store Signal	Error
2	50	50	0%
1.8	200	200	0%
1.8	500	496	-1%
3	250	250	0%
2	400	398	-1%

Table 4 shows the results of testing input 4 (IN4) with payload DEFECT.

Table 4: Test results for input 4 (IN4) by providing a voltage of 5-24V as a signal to publish data

CT (Second)	Real Signal	Store Signal	Error
2	5	5	0%
1.8	25	25	0%
1.8	50	50	0%
3	100	100	0%
2	200	200	0%

The details of the parameters related to pre-implementation testing can be seen in Table 5.

Table 5: Parameter Testing Unit

Parameter	Testing Unit
Device	ESP32U + 8Dbi Antenna
QoS	0
Retain Status	FALSE
WiFi Name	ADRMOBILE
Internet Speed	20Mbps
Extender Router Range	28M
RSSI	(-65) - (-75) dBm

iii. Actual Implementation

After the initial design and testing, the next step is direct implementation on the production floor. The implementation process was carried out in the packaging line area at an automotive spare part manufacturing industry in the Tangerang district. Figure 7 shows the actual wiring of the Development Board OEE Analysis to the packaging line (Auto Dust Cover Machine).

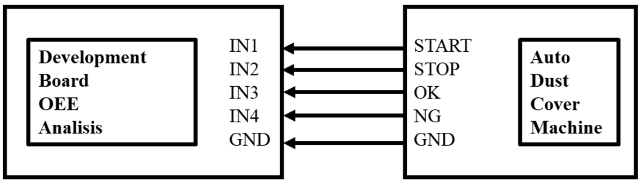
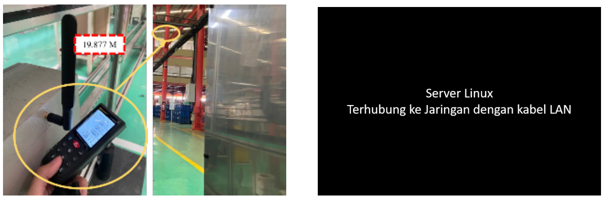


Figure 7: Tool testing during actual implementation

Figure 8 (a) shows the detailed process of placing the unit, where the distance between the Development Board OEE Analysis and the extender router is 19.877 meters. Figure 8 (b) provides information on the OS used and the communication details. The detailed parameters during actual installation on the line are as follows in Table 6.



Server Linux
Terhubung ke Jaringan dengan kabel LAN

(a) (b)

Figure 8: Actual Tool Implementation (a) Distance from Development Board OEE Analysis to Extender Router (b) Server OS information and data communication.

Table 7 shows the results of testing input 1 (IN1) during actual implementation. Data collection was carried out from November 20-25, 2023, over one week with five working days and a total of two shifts.

From Table 7, it can be observed that the accuracy of reading input 1 (IN1) shows varying results between the real signal and the stored signal, with the error being highly variable. This variation in the stored signal in MQTT communication is referred to as packet loss and delay. To address this issue, improvements were

Table 6: Installation Unit Parameters

Parameter	Installation Unit
Device	ESP32U + 8Dbi Antenna
QoS	0
Retain Status	FALSE
WiFi Name	ADRMOBILE
Internet Speed	20Mbps
Extender Router Range	19.877M
RSSI	(-55) - (-65) dBm

Table 7: Results of reading input 1 (IN1) during actual wiring on the auto dust cover machine

Time		Real Signal	Store Signal	Error
Start	Stop			
20/12/2023 07:00	21/12/2023 07:00	8	6	-25%
21/12/2023 07:00	22/12/2023 07:00	8	3	-63%
22/12/2023 07:00	23/12/2023 07:00	8	5	-38%
23/12/2023 07:00	24/12/2023 07:00	8	3	-63%
24/12/2023 07:00	25/12/2023 07:00	8	6	-25%

made by modifying the program through an algorithmic approach.

iv. Problem Handling in Actual Implementation

The algorithmic approach to address the issues encountered during implementation is as follows:

```
{\textbf{Algorithm 1:} Retry Mechanism}
// Reconnect WiFi if it's disconnected
   if (!wifiConnected) {Serial.println("WiFi
   disconnected. Reconnecting...");
  setup_wifi();
 }
   if (!client.connected()) {
      reconnect();
 }
  client.loop();

\textbf{Algorithm 2: Branch Setting}
const char* msg_topic1 = "OEE1";
const char* msg_topic2 = "OEE2";

\textbf{Algorithm 3: Retain Flag}
client.publish(msg_topic1, (const uint8_t*)"START ON",
strlen("START ON"), true);
```

III. Results and Discussion

After making improvements with several algorithmic approaches, the program was re-uploaded and re-implemented on the production floor. Data was collected for one week from December 4 to December 9, 2023. During one week, with 5 working days and a total of two shifts.

Table 8 shows the results of testing input 1 (IN1) after program modification with the algorithmic approach.

Table 8: Results of reading input 1 (IN1) with actual wiring on the auto dust cover machine

Time		Real Signal	Store Signal	Error
Start	Stop			
04/12/2023 07:00	05/12/2023 07:00	8	8	0,00%
05/12/2023 07:00	06/12/2023 07:00	8	8	0,00%
06/12/2023 07:00	07/12/2023 07:00	8	8	0,00%
07/12/2023 07:00	08/12/2023 07:00	8	8	0,00%
08/12/2023 07:00	09/12/2023 07:00	8	8	0,00%

Table 9 shows the results of testing input 2 (IN2) after program modification with the algorithmic approach. There is an error of 5.88% in the actual implementation. The error value in input 2 (IN2) is input for downtime data, which is one of the parameters in OEE measurement. Therefore, this can affect the final OEE measurement result.

Table 9: Results of reading input 2 (IN2) with actual wiring on the auto dust cover machine

Time		Real Signal	Store Signal	Error
Start	Stop			
04/12/2023 07:00	05/12/2023 07:00	26	26	0,00%
05/12/2023 07:00	06/12/2023 07:00	18	18	0,00%
06/12/2023 07:00	07/12/2023 07:00	16	16	0,00%
07/12/2023 07:00	08/12/2023 07:00	34	36	5.88%
08/12/2023 07:00	09/12/2023 07:00	30	30	0,00%

Table 10 shows the results of testing input 3 (IN3) after program modification with the algorithmic approach. The error varies. One cause of the error in input 3 (IN3) is the error appearing in input 2 (IN2). An error in input 2 (IN2) can prevent input 3 (IN3) from storing the signal of OK products to the PostgreSQL database. Besides this error, the values in Table 13 can also be caused by packet loss and delay in the MQTT protocol.

Table 10: Results of reading input 3 (IN3) with actual wiring on the auto dust cover machine

Time		Real Signal	Store Signal	Error
Start	Stop			
04/12/2023 07:00	05/12/2023 07:00	14812	14808	-0,03%
05/12/2023 07:00	06/12/2023 07:00	13254	11890	-10,29%
06/12/2023 07:00	07/12/2023 07:00	13725	13120	-4,41%
07/12/2023 07:00	08/12/2023 07:00	12615	9493	-24,75%
08/12/2023 07:00	09/12/2023 07:00	9945	9729	-2,17%

Table 11 shows the results of testing input 4 (IN4) after program modification with the algorithmic approach.

Table 12 shows the results of OEE measurement based on shift data sampling manually collected by production operators. This data is based on production results recorded on the Daily Production Control (KPH) form and serves as a reference for store signals on each

Table 11: Results of reading input 4 (IN4) with actual wiring on the auto dust cover machine

Time		Real Signal	Store Signal	Error
Start	Stop			
04/12/2023 07:00	05/12/2023 07:00	57	57	0,00%
05/12/2023 07:00	06/12/2023 07:00	60	60	0,00%
06/12/2023 07:00	07/12/2023 07:00	62	62	0,00%
07/12/2023 07:00	08/12/2023 07:00	29	29	0,00%
08/12/2023 07:00	09/12/2023 07:00	47	47	0,00%

input from IN1-IN4 on the Development Board OEE Analysis. The OEE calculation in Table 15 results from applying the OEE measurement formula.

Table 12: OEE measurement results based on manually collected sampling shift data by the operator

Day	Product CT (Second)	OK Products	Loading Time (S)	OEE
1	2.7	14751	50400	79%
2	2.7	11830	50400	63%
3	2.7	13058	50400	70%
4	2.7	9464	50400	51%
5	2.7	9682	50400	52%

Table 13 shows the results of OEE measurement based on shift data sampling using the IoT paradigm with the Development Board OEE Analysis.

Table 13: OEE measurement results based on shift data sampling using the IoT paradigm with the Development Board OEE Analysis

Day	Product CT	OK Products	Loading Time (S)	OEE
1	2.7	14751	50400	79%
2	2.7	11830	50400	63%
3	2.7	13058	50400	70%
4	2.7	9464	50400	51%
5	2.7	9682	50400	52%

Table 14 shows a comparison of measurement results with manually collected shift data and shift data available through the IoT paradigm using the Development Board OEE Analysis.

IV. Conclusion

Based on the implementation results of manual OEE measurement replaced by the IoT paradigm with the Development Board OEE Analysis design, it can be concluded that the MQTT protocol implementation can be applied for OEE measurement. In actual data collection during a one-week implementation, the maximum error in OEE measurement was -16%. Future research plans to address data accuracy issues using other algorithmic approaches to resolve the signal error problem.

Table 14: Comparison of measurement data between manual and IoT paradigm using the Development Board OEE Analysis

Manual	Development Board OEE Analysis	Error
79%	79%	0%
71%	63%	-8%
73%	70%	-3%
67%	51%	-16%
53%	52%	-1%

Acknowledgment

We would like to thank the lecturers of the Magister Teknik Elektro program at Universitas Trisakti and colleagues from the Automation Department who have supported the author in conducting this research and implementing the tool design.

References

[1] D. P. Resmi, “Kementerian perindustrian making indonesia 4.0 revolusi industry 4.0 indonesia,” 2018.

[2] A. C. Herrero, J. A. Sanguesa, F. J. Martinez, P. Garrido, and C. T. Calafate, “Mitigating electromagnetic noise when using low-cost devices in industry 4.0,” *IEEE Access*, vol. 9, pp. 63 267–63 282, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3074588

[3] S. Munirathinam, “Industry 4.0: Industrial internet of things (iiot),” in *Advances in Computers*. Academic Press Inc., 2020, vol. 117, no. 1, pp. 129–164. [Online]. Available: https://doi.org/10.1016/bs.adcom.2019.10.010

[4] A. C. Herrero, F. J. Martinez, P. Garrido, J. A. Sanguesa, and C. T. Calafate, “An interference-resilient iiot solution for measuring the effectiveness of industrial processes,” in *IECON Proceedings (Industrial Electronics Conference)*, 2020, pp. 2155–2160. [Online]. Available: https://doi.org/10.1109/IECON43393.2020.9254454

[5] M. J. Perry, *Evaluating and Choosing an IoT platform*. O’Reilly Media, 2016.

[6] A. S. Muhammed and D. Ucuz, “Comparison of the iot platform vendors, microsoft azure, amazon web services, and google cloud, from users’ perspectives,” in *8th International Symposium on Digital Forensics and Security, ISDFS 2020*, 2020. [Online]. Available: https://doi.org/10.1109/ISDFS49300.2020.9116254

[7] A. Chowdhury and S. A. Raut, “Benefits, challenges, and opportunities in adoption of industrial iot,” 2020.

[8] A. Martikkala, J. David, A. Lobov, M. Lanz, and I. F. Ituarte, “Trends for low-cost and open-source iot solutions development for industry 4.0,” in *Procedia Manufacturing*, 2021, pp. 298–305. [Online]. Available: https://doi.org/10.1016/j.promfg.2021.10.042

[9] G. G. Maulana, A. Budiyarto, and K. Aldi, “Production monitoring system using overall equipment effectiveness (oee) method to improve stamping machine performance,” *Desiminating Information on the Research of Mechanical*

Engineering-Jurnal Polimesin, vol. 20, no. 2, 2022. [Online]. Available: http://e-jurnal.pnl.ac.id/polimesin

[10] G. G. Maulana, S. Aminah, and B. A. Nugraha, "Implementation of a production monitoring system using iiot based on mobile application," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 5, pp. 1077–1087, 2023. [Online]. Available: https://doi.org/10.29207/resti.v7i5.5221

[11] A. Halldorsson, "Iiot data collection for oee measurements," 2016.

[12] Mastang and M. A. Pahmi, "Development of raspberry pi applied to real-time monitoring of overall equipment effectiveness (oee)," in *Journal of Physics: Conference Series*. Institute of Physics Publishing, 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1477/5/052013

[13] L. Kong, U. Tunku, and A. Rahman, "Oee implementation on safety helmet plastic injection process chua yun chin a project report submitted in partial fulfilment of the requirements for the award of bachelor of engineering (honours) mechatronics engineering," 2020.

[14] P. Surya, A. Suenbuel, and A. S. A. Doss, "Industrial internet of things - control of industry simulation through different protocols," *J Phys Conf Ser*, vol. 2115, no. 1, p. 12011, 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/2115/1/012011

[15] S. Nakajima, *Introduction to TPM: total productive maintenance*. Productivity Press, Inc., 1988.

[16] N. Tewari, N. Deepak, M. Joshi, and J. S. Bhatt, "Comparative study of iot development boards in 2021: Choosing right hardware for iot projects," in *Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 357–361. [Online]. Available: https://doi.org/10.1109/ICIEM51511.2021.9445290

[17] M. Babiuch, P. Foltýnek, and P. Smutný, "Using the esp32 microcontroller for data processing," in *2019 20th International Carpathian Control Conference (ICCC)*, 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/CarpathianCC.2019.8765944

[18] R. Poljak, P. Poščić, and D. Jakšić, "Comparative analysis of the selected relational database management systems," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 1496–1500. [Online]. Available: https://doi.org/10.23919/MIPRO.2017.7973658

[19] K. Ferencz and J. Domokos, "Using node-red platform in an industrial environment," *XXXV. Jubileumi Kandó Konferencia, Budapest*, pp. 52–63, 2019.

[20] T. Beermann *et al.*, "Implementation of atlas distributed computing monitoring dashboards using influxdb and grafana," *EPJ Web of Conferences*, p. 3031, 2020.

[21] M. W. Habibi, A. Bhawiyuga, and A. Basuki, "Rancang bangun iot cloud platform berbasis protokol komunikasi mqtt," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 2, pp. 479–485, 2018.

[22] F. A. Pazos, "Performance evaluation of mqtt broker servers deployed in the cloud," *Memorias de las JAIIO*, vol. 9, no. 3, 2023.