**Journal of Artificial Intelligence and Software Engineering**
Vol. 5, No. 2, Juni 2025, pp. 403~409
E-ISSN: 2777-001X, DOI: 10.30811/jaise.v5i2.6512 

# An IoT-Enabled Low Latency Automatic Identification System Using Round-Robin Scheduling Algorithm

**Belen Septian[1*], Aidil Adrianda[2], M. Fauzan Ridho[3], Md. Misbahuddin[4]**
[1,3] Department of Electrical Engineering, Faculty of Science and Engineering, Universitas Bangka Belitung, Balunijuk, 33172, Indonesia
[2] Department of Mathematics, Faculty of Science and Engineering, Universitas Bangka Belitung, Balunijuk, 33172, Indonesia
[4] Department of Electronics and Communication Engineering, College of Engineering, Osmania University, Hyderabad, 500001, India

**Informasi Artikel**

| | | |
|---|---|---|
| Diterima | : | 21 Februari 2025 |
| Revisi | : | 3 Mei 2025 |
| Publikasi | : | 20 Juni 2025 |

***Kata Kunci:***

*Automatic Identification System*
*Internet of Things*
*Real-Time Operating System*
*Microcontroller*
*Round-Robin Scheduling*

**ABSTRAK**

Automatic Identification System (AIS) adalah teknologi maritim yang penting untuk meningkatkan keselamatan navigasi dan pelacakan kapal. Sistem ini terintegrasi dengan *Internet of Things* (IoT) untuk transmisi data secara *real-time*. Namun, karena banyaknya tugas yang dijalankan, latensi sistem meningkat. Untuk mengatasi masalah tersebut, penelitian ini mengusulkan pendekatan penjadwalan tugas yang dioptimalkan menggunakan algoritma Round-Robin dengan mekanisme pengacakan ulang. Metode yang diusulkan diterapkan pada mikrokontroler ESP32, memungkinkan pemrosesan pesan AIS secara *real-time* serta meminimalkan latensi dan konsumsi energi. Hasil eksperimen menunjukkan bahwa metode *hybrid* Round-Robin dan pengacakan mencapai waktu transmisi rata-rata terendah, yaitu 28,736 detik, mengungguli pendekatan penjadwalan Prioritas dan Round-Robin. Temuan penelitian ini berkontribusi dalam meningkatkan kemampuan pemrosesan secara *real-time* pada sistem tertanam untuk pelacakan kapal. Lebih lanjut, penelitian di masa depan sebaiknya berfokus pada penanganan fluktuasi pada kondisi transien dengan teknik penjadwalan adaptif.

**ABSTRACT**

The Automatic Identification System (AIS) is a vital maritime technology that enhances navigation safety and vessel tracking. This system is integrated with the Internet of Things (IoT) for real-time data transmission. However, due to multiple tasks being employed, the latency of the system increases. To address this problem, this study proposes an optimized task-scheduling approach using a Round-Robin algorithm with an additional task-reshuffling mechanism. The proposed method is implemented on an ESP32 microcontroller, enabling real-time processing of AIS messages. Experimental results demonstrate that the hybrid Round-Robin and Shuffling method achieves the lowest average transmission time of 28.736 seconds, outperforming traditional Priority and standard Round-Robin scheduling approaches. The findings of this study contribute to enhancing real-time processing capabilities in embedded vessel tracking systems. Following this, future research should focus on addressing transient fluctuations using adaptive scheduling techniques.



***Penulis Koresponden**

Email: belenseptian@ubb.ac.id

Cara sitasi IEEE::

B. Septian, A. Adrianda, M. F. Ridho, & M. Misbahuddin, “An IoT-Enabled Low Latency Automatic

Identification System Using Round-Robin Scheduling Algorithm" *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 2, pp. 403-409, Juni 2025. doi: 10.30811/jaise.v5i2.6512

## 1. INTRODUCTION

The Automatic Identification System (AIS) receiver represents a maritime technology that utilizes radio communication to monitor vessel activity across the world's oceans [1], [2]. AIS enables the detection and tracking of vessels by transmitting critical data, including vessel identity, position, speed, and direction. This information is relayed through Very High Frequency (VHF) radio waves within the 156-162 MHz range, facilitating real-time monitoring and enhancing situational awareness for maritime stakeholders [3]. The primary objectives of AIS are to improve navigation safety, mitigate the risk of collisions, and improve overall naval security [4].

The implementation of AIS is governed by international regulations established under the International Maritime Organization (IMO). These regulations mandate the use of ship tracking systems, including AIS, to ensure compliance with global standards aimed at safeguarding lives and vessels at sea. As a result, the installation of AIS receivers on ships has become a critical requirement, underscoring its importance in modern maritime operations [5]. AIS data is structured using the standard NMEA 0183 format, which ensures compatibility with various maritime communication systems. This format consists of a series of encoded messages, each containing essential navigational and vessel information. In total, there are 11 data fields within these messages that can be decoded, such as vessel identity, position, speed, and heading [6]. Once decoded, this data is integrated into an Internet of Things (IoT) system, enabling real-time access and analysis for users.

The complexity of IoT systems increases as more parameters are integrated [7]. There are a few studies that proposed a deep learning-based approach to predict the trajectory based on different parameters [8], [9]. Heiselberg et al. [10] proposed the estimation of ship velocity by utilizing the combination of satellite images and NMEA data. However, these studies did not take into account the latency caused by the execution of multiple tasks within the system.

The studies on efficient multitasking have been done by [11]. It brought forward the FreeRTOS for AIS and fire detection systems. However, the use of an 8-bit microcontroller is not well-suited for a real-time operating system (RTOS) environment due to its limited processing power, memory capacity, and multitasking capabilities. Considering the drawbacks of previous studies, this research puts forward a Round-Robin algorithm to manage multiple tasks simultaneously. This method is advantageous due to its simplicity, but it tends to waste a significant amount of resources [12]. To address this problem, this study optimizes the use of the Round-Robin algorithm with an additional task reshuffling method. The main contribution of this study can be summarized as follows:

- This study proposes an efficient task-scheduling algorithm that can be executed on a 32-bit microcontroller.
- It puts forward a Round-Robin method to schedule multiple tasks. In addition, tasks are reshuffled in accordance with optimum priority when the system cannot establish the connection.
- Taking the above-mentioned points into account, the proposed system contributes to minimizing the latency of the data transmission.

The rest of this paper is structured into 3 sections. Section 2 explores the system architecture, providing an in-depth look at its design and operation. Section 3 reviews the experimental results and validates the method, including an analysis of the data and its significance. Finally, the conclusion summarizes the key findings, highlights the study's contributions, and proposes future research directions.

## 2. METHODOLOGY

This section provides a detailed overview of the proposed Internet of Things (IoT) system implemented in this study. The system is composed of two primary components: the hardware architecture and the proposed Round-Robin algorithm. The hardware architecture defines the physical components, including sensors, microcontrollers, communication modules, and other essential devices that enable data collection, processing, and transmission.

### 2.1. System Architecture

Figure 1 illustrates a system architecture for processing and transmitting AIS (Automatic Identification System) messages from a ship or vessel to a server. The system consists of an AIS transceiver installed on the vessel, which communicates over VHF frequencies with an AIS receiver. The AIS transceiver is equipped with a GPS antenna for positioning data and an AIS receiver antenna for receiving

AIS messages. The AIS receiver onshore collects AIS messages in NMEA 0183 format and transmits them to an ESP32 microcontroller for further processing.

The ESP32 processes the received AIS messages using a Round-Robin algorithm and publishes the data to an MQTT broker. The broker then enables communication between the ESP32 and a server by allowing the server to subscribe to the published messages. The server, in turn, exposes the received data through an API for further use.

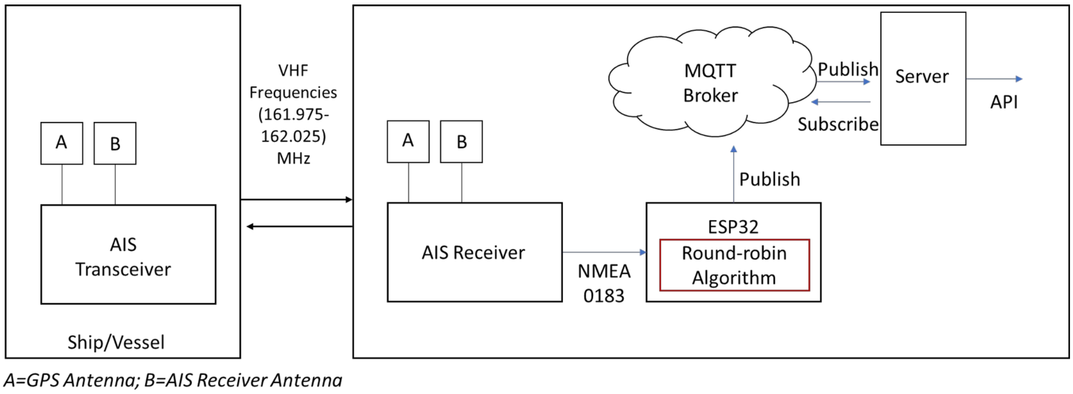


Figure 1. The architecture of the proposed system

### 2.2. Proposed Round-Robin Method

The proposed Round-Robin algorithm is shown in Figure 2. It is designed to optimize task execution based on transmission time. In this study, 3 tasks have been employed such as display, receive, and transmit. The process begins by calculating the transmission time of a given task. If the transmission time is less than 500 milliseconds, the algorithm checks whether the ready queue is empty. If the queue is empty, the system proceeds to execute the task for the quantum time. If tasks are present in the queue, the quantum time (QT) is determined as the average of all burst times. The task is then executed for the allocated quantum time unless its burst time is shorter than the quantum time.

If the transmission time exceeds 500 milliseconds, the algorithm prioritizes optimal scheduling. Tasks are shuffled to an optimal priority scheduling mechanism to minimize latency. The execution process continues cyclically, ensuring fair allocation of processing time while preventing any single task from monopolizing system resources.

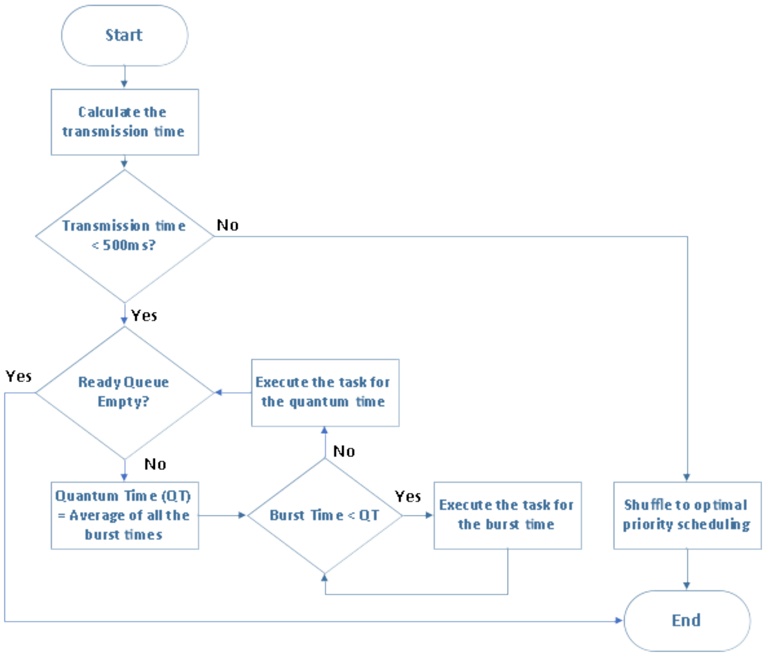


Figure 2. The flowchart of the proposed Round-Robin algorithm

## 3. RESULTS AND DISCUSSIONS

This section examines the experimental results of the proposed system, highlighting its effectiveness in carrying out the given tasks. Furthermore, a comparison of the methods concerning transmission time is performed. The evaluation considers that a stable internet connection with a maximum bandwidth of 20 Mbps is provided to the system.

### 3.1. Experimental Setup

Figure 3 presents the hardware setup of the proposed system, which integrates key components to facilitate efficient data processing and communication. The setup includes an ESP32 microcontroller, which serves as the core processing unit responsible for handling data and executing the Round-Robin algorithm. The specifications of the aforementioned are presented in Table 1. Additionally, an RS232 converter is incorporated to enable serial communication between the microcontroller and other connected devices. The system is powered by an isolated power supply module, which converts AC voltage to a stable DC output required for the operation of the ESP32 and other electronic components.

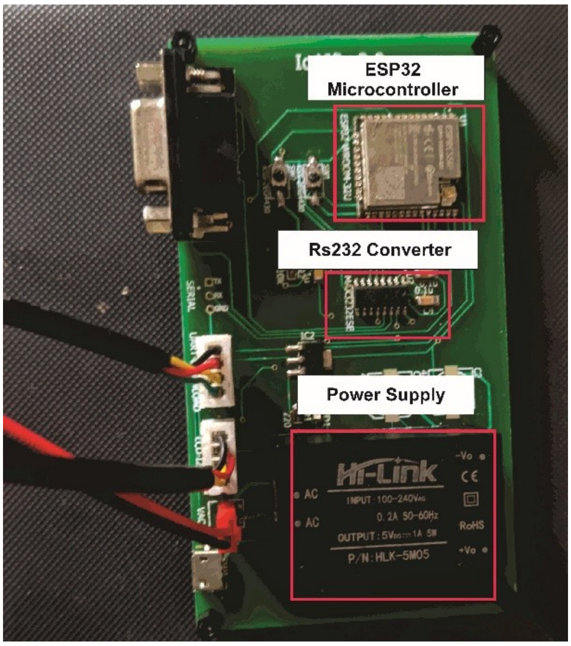


Figure 3. Hardware setup of the proposed system

Table 1. Specifications of ESP32 microcontroller

| Parameter | Value |
|---|---|
| MCU | Tensilica Xtensa single-/dual-core 32-bit LX6 |
| Operating Frequency | 240MHz |
| Flash Memory | 4MB |
| SRAM | 512KB |
| WiFi | 802.11 b/g/n |

Table 1 presents the specifications of the ESP32 microcontroller, highlighting its key hardware capabilities. The ESP32 is powered by a Tensilica Xtensa LX6 processor, which features a single or dual-core 32-bit architecture designed for high-performance embedded applications. It operates at a frequency of 240 MHz. The microcontroller includes 4MB of flash memory for program storage and 512KB of SRAM for temporary data handling. Additionally, the ESP32 supports WiFi connectivity (802.11 b/g/n), enabling wireless communication for IoT and network-based applications. These specifications make the ESP32 a versatile platform for RTOS applications, as demonstrated by various studies conducted by [13]-[16].

Table 2. Network Parameters

| Parameter | Value |
|---|---|
| Network Bandwidth | 10Mbps |
| Sentence Size | 82bytes |
| Protocol | TCP/IP |

Table 2 outlines the network parameters of the proposed system. The system operates over a stable network connection with a bandwidth of 10 Mbps. Data communication is established using the TCP/IP protocol, transmitting the sentence with a size of 82 bytes. Each character within the sentence is encoded using the extended ASCII format, occupying 8 bits per character.

### 3.2. Experimental Results

The execution of the NMEA 0183 decoder on the server is shown in Figure 4, where raw AIS data is processed and parsed into structured information. The decoded data includes essential attributes such as latitude, longitude, speed, heading, and various message flags. The extracted NMEA sentence represents a position report transmitted by a vessel. This data serves as the foundation for vessel tracking applications, converting raw AIS messages into geographical coordinates and navigation details.

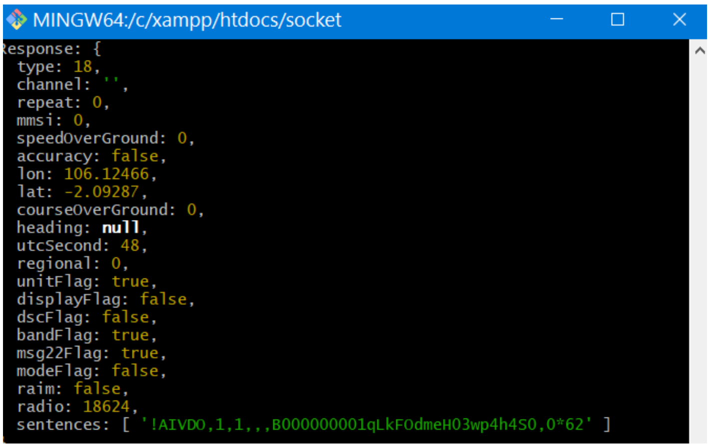

Figure 4. Execution of the NMEA 0183 decoder on the server

Figure 5 illustrates the next phase of the AIS data processing pipeline, where the decoded information is integrated into a web-based vessel tracking system. This system takes the parsed latitude and longitude values obtained from the NMEA 0183 decoder and visualizes vessel positions on a digital map. Each vessel is marked with a color-coded triangular icon, which may signify different types of ships or operational statuses. This visual approach enhances situational awareness for maritime authorities and port operators.

A key feature of this system is the ability to access detailed information about each vessel through a pop-up window. This feature allows users to track ship movements over time, assess navigation patterns, and detect potential anomalies in vessel behavior. Additionally, the inclusion of timestamps ensures that the data remains up-to-date, improving the reliability of the tracking system.

Figure 5. Vessel positions on the web server

### 3.3. Performance Analysis

Figure 6 presents a comparative analysis of transmission time among the proposed method and different scheduling algorithms such as Priority, Priority with Shuffling, and Round Robin. The x-axis represents the number of samples, while the y-axis denotes the transmission time in seconds. The plot reveals that Priority and Round Robin exhibit a steady increase in transmission time as the number of samples rises. However, the introduction of shuffling leads to variations, particularly in Priority with Shuffling, where an abrupt spike is observed around 15 samples before stabilizing.

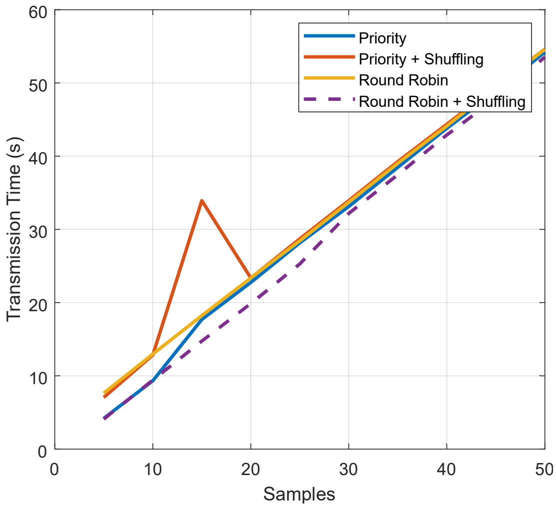


Figure 6. Transmission time comparison between the proposed method and other algorithms

The Priority method, represented by the blue line, follows a consistent trajectory, indicating a predictable transmission time increase. In contrast, Priority with Shuffling experiences noticeable instability due to abrupt shifting between the tasks. The Round Robin approach remains close to the Priority method but slightly below it, suggesting a marginally lower latency. Table 3 shows the average transmission time of different scheduling methods.

Table 3. The average transmission time of the proposed method and other algorithms was measured for 50 samples

| Method | Transmission Time (s) |
|---|---|
| Priority | 30.067 |
| Priority + Shuffling | 32.759 |
| Round-Robin | 31.148 |
| Round-Robin + Shuffling | 28.736 |

The results indicate that the combination of the Round-Robin and Shuffling method achieves the lowest transmission time at 28.736 seconds, highlighting its efficiency in minimizing delays. This suggests that incorporating shuffling with the Round-Robin scheduling algorithm leads to a more optimal transmission approach compared to other scheduling algorithms. The reduced transmission time demonstrates that this hybrid method can enhance the microcontroller's performance, especially when multiple tasks are involved.

However, it is important to note that this approach requires more than 10 samples before achieving stable transmission, meaning that initial fluctuations may impact performance during the early stages. This delay in stabilization could lead to increased power consumption, as additional resources are needed to manage variations before reaching a steady state. This may be addressed in future research with the aid of adaptive scheduling algorithms.

## 4. CONCLUSION

This study proposed an optimized task-scheduling algorithm that integrates the Round-Robin method with a task reshuffling mechanism to enhance the transmission efficiency of the AIS receiver. The experimental results demonstrated that the hybrid Round-Robin and Shuffling method achieved the lowest average transmission time of 28.736 seconds, outperforming traditional Priority and standard Round-Robin scheduling. This highlights the effectiveness of the proposed approach in reducing latency and improving system responsiveness. Additionally, the study confirmed that while shuffling improved the efficiency of Round-Robin scheduling, it negatively impacted Priority scheduling by introducing instability.

Despite its advantages, the proposed method required more than 10 samples to achieve consistent transmission, which in turn led to increased power consumption. This limitation suggests the need for further refinement, possibly through adaptive scheduling techniques or machine learning-based optimization. Future research should focus on mitigating transient fluctuations and exploring dynamic scheduling strategies to further enhance real-time performance and energy efficiency in embedded systems.

## ACKNOWLEDGEMENT

We extend our heartfelt gratitude to CV. Henafra Digitech for their invaluable assistance and support during the completion of this research.

## REFERENCES

[1] Y. Apriani, W. A. Oktaviani, and I. M. Sofian, ‘Vessel Tracking System Based LoRa SX1278’, *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 3, pp. 693–707, Jul. 2023, doi: 10.26555/jiteki.v9i3.26385.

[2] E. Lee, A. J. Mokashi, S. Y. Moon, and G. Kim, ‘The Maturity of Automatic Identification Systems (AIS) and Its Implications for Innovation’, *J Mar Sci Eng*, vol. 7, no. 9, p. 287, Aug. 2019, doi: 10.3390/jmse7090287.

[3] Q. Hu, J. Cao, G. Gao, L. Xu, and M. Song, ‘Study of an Evaluation Model for AIS Receiver Sensitivity Measurements’, *IEEE Trans Instrum Meas*, vol. 69, no. 4, pp. 1118–1126, Apr. 2020, doi: 10.1109/TIM.2019.2910341.

[4] A. Maulidi, M. Abdullah, and D. Handani, ‘Virtual private network (VPN) model for AIS real time monitoring’, *IOP Conf Ser Earth Environ Sci*, vol. 1081, no. 1, p. 012028, Sep. 2022, doi: 10.1088/1755-1315/1081/1/012028.

[5] A. Maulidi, T. Irmiyana, A. F. Ilman, Annafiyah, and Kuzzairi, ‘The development of internet-of-things (IoT)-based safety monitoring system in north sea madura’, *IOP Conf Ser Earth Environ Sci*, vol. 972, no. 1, p. 012013, Jan. 2022, doi: 10.1088/1755-1315/972/1/012013.

[6] D. Yang, L. Wu, S. Wang, H. Jia, and K. X. Li, ‘How big data enriches maritime research – a critical review of Automatic Identification System (AIS) data applications’, *Transp Rev*, vol. 39, no. 6, pp. 755–773, Nov. 2019, doi: 10.1080/01441647.2019.1649315.

[7] I. Behnke, L. Pirl, L. Thamsen, R. Danicki, A. Polze, and O. Kao, ‘Interrupting Real-Time IoT Tasks: How Bad Can It Be to Connect Your Critical Embedded System to the Internet?’, in *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*, IEEE, Nov. 2020, pp. 1–6. doi: 10.1109/IPCCC50635.2020.9391536.

[8] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, ‘A Multi-Task Deep Learning Architecture for Maritime Surveillance Using AIS Data Streams’, in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, Oct. 2018, pp. 331–340. doi: 10.1109/DSAA.2018.00044.

[9] W. Wang *et al.*, ‘A multi-task learning-based framework for global maritime trajectory and destination prediction with AIS data’, *Maritime Transport Research*, vol. 3, p. 100072, 2022, doi: 10.1016/j.martra.2022.100072.

[10] P. Heiselberg, K. Sørensen, and H. Heiselberg, ‘Ship velocity estimation in SAR images using multitask deep learning’, *Remote Sens Environ*, vol. 288, p. 113492, Apr. 2023, doi: 10.1016/j.rse.2023.113492.

[11] K. D. E. Kerrouche *et al.*, ‘Applications of Nanosatellites in Constellation: Overview and Feasibility Study for a Space Mission Based on Internet of Space Things Applications Used for AIS and Fire Detection’, *Sensors*, vol. 23, no. 13, p. 6232, Jul. 2023, doi: 10.3390/s23136232.

[12] J.-C. Bauduin and L. Roman, ‘Improving security in embedded systems secure multitasking in the ESP privilege separation framework’, École polytechnique de Louvain, Leuven, 2023.

[13] N. Cameron, ‘ESP32 microcontroller features’, in *Electronics Projects with the ESP8266 and ESP32*, Berkeley, CA: Apress, 2021, pp. 641–682. doi: 10.1007/978-1-4842-6336-5_22.

[14] R. Gautam, D. Pujara, M. Shah, and D. Shah, ‘Adapting a Real-Time Operating System to the RISC-V Based ESP32’, 2024, pp. 459–468. doi: 10.1007/978-981-97-1329-5_37.

[15] I. Plauska, A. Liutkevičius, and A. Janavičiūtė, ‘Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller’, *Electronics (Basel)*, vol. 12, no. 1, p. 143, Dec. 2022, doi: 10.3390/electronics12010143.

[16] J. Arm, O. Baštán, O. Mihálik, and Z. Bradáč, ‘Measuring the Performance of FreeRTOS on ESP32 Multi-Core’, *IFAC-PapersOnLine*, vol. 55, no. 4, pp. 292–297, 2022, doi: 10.1016/j.ifacol.2022.06.048.