

Blocking Delay Effects on Microcontroller Speed and Responsiveness in Industrial IoT Devices: A Systematic Review

Pauladie Susanto^{1*}, Weny Indah Kusumawati², Harianto³, Heri Pratikno⁴, Herru Prastyo⁵

^{1*,2,3,4}Department of Computer Engineering, Universitas Dinamika, Surabaya, East Java, Indonesia

⁵Department of Air Navigation Technology, Politeknik Penerbangan Makassar, South Sulawesi, Indonesia

e-mail: pauladie@dinamika.ac.id^{1*}, weny@dinamika.ac.id², hari@dinamika.ac.id³, heri@dinamika.ac.id⁴, heru.prastyo@poltekbangmakassar.ac.id⁵

Article Information

Article History:

Received : 11 January 2026
Revised : 20 March 2026
Accepted : 1 April 2026
Published : 16 April 2026

***Correspondence:**

pauladie@dinamika.ac.id

Keywords:

Blocking Delay, Microcontroller Speed, Microcontroller Responsiveness, IoT Devices, Industrial Automation

Copyright © 2026 by Author.
Published by Universitas Dinamika.



This is an open access article under the CC BY-SA license.



10.37802/joti.v8i1.1347

Journal of Technology and Informatics (JoTI)

P-ISSN 2721-4842

E-ISSN 2686-6102

[https://e-](https://e-journals.dinamika.ac.id/index.php/joti)

[journals.dinamika.ac.id/index.php/joti](https://e-journals.dinamika.ac.id/index.php/joti)

Abstract:

This review synthesizes research on the impact of blocking delay on microcontroller speed and responsiveness in IoT devices for industrial automation. It evaluates blocking delay effects on microcontroller performance. The review benchmarks scheduling and edge computing techniques, identifies mitigation strategies, compares case study outcomes, and analyzes architectural and software factors influencing blocking delay. A systematic analysis of experimental, simulation, and co-design studies was conducted. The analysis focused on real-time scheduling, interrupt handling, network-induced latency, and edge computing integration. Key findings reveal that advanced scheduling algorithms and interrupt nesting significantly reduce blocking delays and improve task responsiveness. Edge computing and hardware optimizations also minimize network-induced latency and enhance local processing capabilities. Multiple sources of blocking delay, including resource contention and network overload, are mitigated through adaptive scheduling and hardware-assisted mechanisms. Real-world case studies confirm substantial latency reductions and improved control performance in industrial IoT contexts. These findings underscore the interplay of software and hardware factors in shaping microcontroller responsiveness. The review highlights the necessity for scalable, integrated solutions that address dynamic industrial environments. It informs the design of more responsive and efficient microcontroller-based IoT systems for industrial automation.

INTRODUCTION

Research on the impact of blocking delay on microcontroller speed and responsiveness

in IoT devices for industrial automation has emerged as a critical area of inquiry due to the increasing integration of embedded systems in time-sensitive industrial environments. The evolution from traditional embedded control systems to networked IoT architectures has introduced complex timing challenges, particularly in real-time task execution and network communication latency [1], [2]. Industrial automation demands stringent real-time guarantees to ensure process reliability and efficiency, with latency reductions from cloud to edge computing paradigms demonstrating significant improvements in responsiveness [3], [4]. The proliferation of multicore microcontrollers and advanced scheduling techniques further underscores the importance of understanding blocking delays and their effects on system performance [5], [6].

The specific problem is the negative effect of blocking delays on microcontroller responsiveness and throughput in industrial IoT. These delays come from interrupt handling, resource contention, and scheduling policies [7], [5]. New mitigation strategies help, but understanding how blocking delays affect microcontroller speed is still incomplete [1], [8], [9]. Some studies suggest hardware solutions like interrupt nesting and direct register access to reduce latency [7], [10]. Others suggest software scheduling and resource allocation [11], [12]. This lack of consensus makes it hard to optimize IoT devices for strict real-time industrial tasks. As a result, systems may miss deadlines or lose control of performance [13], [14].

A conceptual framework for this review integrates three key concepts: blocking delay as the latency introduced by task preemption and resource contention; microcontroller speed as the effective processing rate under real-time constraints; and responsiveness as the system's ability to meet timing deadlines in IoT edge environments [15], [16], [17]. These concepts are interrelated, as blocking delays directly affect both speed and responsiveness, influencing overall system reliability. The framework guides the analysis of how architectural and scheduling factors impact real-time performance in industrial IoT microcontrollers.

The purpose of this review is to analyze the impact of blocking delay on microcontroller speed and responsiveness in IoT devices used for industrial automation. It aims to synthesize detailed analyses and case studies to bridge the identified knowledge gap, providing insights into effective mitigation strategies and design considerations. This review adds value by consolidating diverse approaches across hardware and software domains, informing future research and practical implementations in real-time industrial IoT systems.

METHOD

This systematic review employed a structured literature search strategy to investigate blocking delay impacts on microcontroller speed and responsiveness in industrial IoT devices. The methodology comprised four sequential phases, as shown in Figure 1: query transformation, database screening, citation chaining, and relevance assessment.

Literature Search and Screening

The primary research question was broken down into five thematic queries: (1) direct blocking delay impacts on microcontroller responsiveness; (2) latency effects and real-time execution strategies; (3) scheduling algorithms and the role of edge computing; (4) methods for minimizing latency; and (5) alternative frameworks for real-time task scheduling. These queries were run against a bibliographic database with over 270 million records. Only peer-reviewed articles and conference papers that directly addressed blocking delay, microcontroller performance, real-time scheduling, network-induced latency, or edge computing in industrial IoT were included. Studies unrelated to these topics or lacking methodological rigor were

excluded. This process yielded 329 candidate papers.

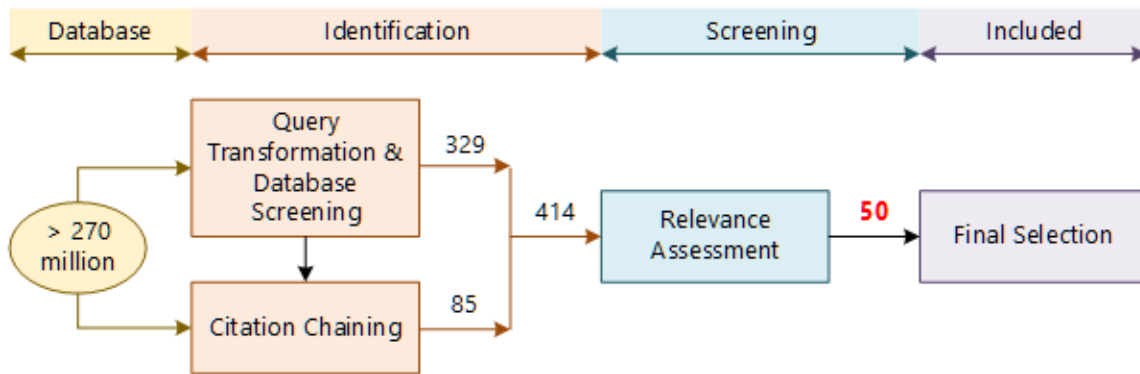


Figure 1. Flow diagram of the article selection process

Citation Chaining and Relevance Assessment

Backward citation chaining examined the reference lists of core papers to identify foundational studies. In contrast, forward chaining captured recent publications citing these works, adding 85 papers for a total pool of 414 candidates. To ensure comprehensive evaluation, a systematic relevance assessment was then evaluated each paper against review objectives, thematic contribution, methodological rigor, and empirical evidence. Through this process, 406 relevant papers were identified. Of these, 50 were classified as highly relevant based on their substantial focus on blocking delay mechanisms, mitigation strategies, case studies, or architectural considerations in microcontroller-based industrial IoT systems. Ultimately, these 50 papers constituted the core evidence base for analysis and synthesis.

RESULTS AND DISCUSSION

Descriptive Summary of the Studies

Table 1 presents a comprehensive synthesis of the reviewed literature. It is organized across five thematic dimensions: blocking delay magnitude, scheduling algorithm efficiency, network-induced latency impact, edge computing integration, and case study performance outcomes. Each dimension consolidates key findings from multiple studies. The table reveals convergent evidence regarding the sources, impacts, and mitigation strategies for blocking delays in microcontroller-based industrial IoT systems. It demonstrates the interdependence of hardware optimizations, scheduling techniques, and architectural approaches in achieving low-latency, responsive microcontroller performance for time-critical industrial automation applications.

Table 1. Review of The Studies

Thematic Area	Key Findings	Supporting Studies
Blocking Delay Magnitude	High delays from network overload (packet bursts, interrupt handling) and resource contention; 60-75% reduction via hardware optimization (DRA, parallel context saving); variable delays depend on network conditions and task scheduling.	[1], [18], [2], [10], [5], [14], [19], [20], [21], [7], [8], [17], [22], [23], [24], [9], [25], [26], [27]
Scheduling Algorithm Efficiency	Adaptive, rate-based, and deterministic scheduling improve deadline adherence and responsiveness. In addition, multi-task learning and job-class-level	[11], [28], [29], [30], [12], [31], [32], [15], [16], [33], [7], [34],

	scheduling enhance resource utilization. Moreover, offline and high-precision scheduling reduce runtime overhead. Finally, scheduling co-design integrates control and timing considerations.	[35], [6], [36], [37], [38], [39], [40], [41], [9]
Network-Induced Latency Impact	Network traffic and interrupt handling are major contributors to blocking delays. Furthermore, network queuing, congestion, and unreliable wireless links increase latency and jitter. However, hardware acceleration and optimized protocols can mitigate network delays. Additionally, edge/fog architectures reduce latency by enabling local processing.	[1], [18], [21], [33], [19], [20], [10], [42], [24], [3], [39], [4], [2], [43], [44], [45], [46], [5], [13]
Edge Computing Integration	Edge computing central to latency reduction (up to 92.5%); edge analytics, embedded AI, and distributed MCU systems improve local processing; hybrid edge-cloud architectures balance latency and resource efficiency; dynamic task scheduling and resource allocation adapt to varying workloads	[3], [43], [47], [44], [45], [46], [34], [48], [28], [37], [40], [4], [49], [10], [11], [33], [7], [30], [5], [23], [50], [17], [13]
Case Study Performance Outcomes	Leveraging these foundational technologies delivers dramatic latency reductions and sharper deadline adherence across industrial automation. Rigorous RISC-V, ESP32, and Raspberry Pi trials validate the gains, fueling heightened control in CNC systems, motion control, and smart manufacturing. Yet challenges in scaling and optimizing for complex, dynamic environments persist.	[1], [3], [48], [2], [28], [41], [7], [8], [17], [10], [11], [33], [43], [44], [34], [30], [47], [23], [50], [22], [35], [37], [49], [9], [6], [27]

Critical Analysis and Synthesis

Extending this analysis, the reviewed literature presents a comprehensive yet multifaceted exploration of blocking delay impacts on microcontroller performance in industrial IoT systems. Substantial advances have been demonstrated in hardware optimizations, scheduling algorithms, and architectural innovations. However, critical gaps and methodological limitations persist. Table 2 synthesizes the strengths and weaknesses identified across seven analytical dimensions. It provides a balanced assessment of current knowledge and highlights areas requiring further investigation.

Tabel 2. Critical Analysis of Research Strengths And Weaknesses

Aspect	Strengths	Weaknesses
Impact of Blocking Delay on Micro-controller Responsiveness	Experimental validations robustly demonstrate direct correlations between blocking delays and responsiveness degradation in time-critical contexts, with network packet overload studies providing quantitative evidence of deadline misses and performance impacts [1], [18].	Studies predominantly examine isolated scenarios or specific network conditions, limiting generalizability to heterogeneous industrial environments with unpredictable workloads [1], [21].
Scheduling and	Advanced scheduling paradigms substantially improve task	Scheduling solutions are predominantly evaluated on limited

Interrupt Handling Techniques	management and reduce blocking delays through predictable overhead mechanisms [11], [30], [28]. Interrupt nesting techniques such as ENEST for RISC-V architectures offer validated responsiveness enhancements supported by both theoretical analysis and practical implementations [7].	hardware platforms or simulated environments, potentially failing to capture full industrial IoT constraint spectra [11], [30]. Integration challenges with legacy systems, impacts on system complexity and maintainability, and potential overhead-induced latency offsets in resource-constrained devices remain inadequately addressed [13].
Edge Computing and Hardware Optimizations	Edge computing paradigms consistently enable significant latency reductions through localized processing, with case studies demonstrating microsecond-level responsiveness [3], [4]. Hardware innovations including Direct Register Access and parallel context saving mechanisms effectively minimize network I/O and interrupt handling delays critical for industrial automation [10], [8].	Resource contention management and deterministic performance guarantees in multicore and heterogeneous edge platforms remain challenging [5], [43]. Edge solution scalability, cloud interoperability, and specialized architecture requirements constrain widespread adoption across diverse industrial deployment contexts [10], [8].
Sources and Mitigation of Blocking Delay	Multiple blocking delay sources are systematically identified, with mitigation strategies spanning adaptive scheduling, resource reservation, and hardware-assisted management demonstrating latency and jitter reductions [1], [5], [8], [41].	Mitigation approaches predominantly address symptoms rather than root causes, exhibiting reactive rather than proactive characteristics [1], [5], [41].
Real-World Case Studies and Practical Implications	Detailed industrial case studies in CNC systems and smart manufacturing validate theoretical models with tangible metrics on system accuracy, deadline adherence, and operational efficiency [2], [39]. Multi-MCU and hybrid edge-cloud implementations demonstrate practical enhancement pathways [46], [49].	Case studies frequently emphasize specific applications or controlled environments, constraining applicability breadth across diverse industrial scenarios [2], [46]. Integration complexity of multiple mitigation techniques, cost-benefit analyses, and long-term reliability and maintenance considerations remain underexplored [49], [39].
Architectural and Software Factors	Research establishes critical roles of multicore architectures, pipeline configurations, and RTOS features in shaping blocking delay characteristics [22], [6], [38]. Software factors including task partitioning and dynamic reconfiguration interact synergistically with hardware capabi-	Co-design complexity challenges practical deployment in legacy or resource-limited systems [22], [23]. Performance-complexity and performance-power consumption trade-offs lack systematic quantification, while industrial IoT device heterogeneity complicates

	lities, informing effective co-design optimization approaches [23], [31].	architectural standardization [9], [6].	optimization
Benchmarking and Evaluation Methodologies	Diverse methodologies enable multifaceted evaluations and comparative analyses of mitigation efficacy across scheduling algorithms and hardware optimizations [44], [13], [24].	Benchmarking uniformity is insufficient for robust cross-study comparisons [13], [12]. Reliance on synthetic workloads and limited datasets may inadequately reflect operational complexities [44], [45], [24].	

Thematic Review of Literature

A review of the research shows eight main themes that help address the many challenges of reducing blocking delays in microcontroller-based industrial IoT systems. Table 3 shows how these themes are spread out and linked in the studies. Most papers focus on scheduling algorithms and edge computing, which appear in 21 and 18 papers, showing their importance to current strategies. Newer themes, like dynamic reconfiguration and control system integration, show how research is adapting to more complex industrial needs.

Table 3. Thematic Distribution And Characterization

Theme	Prevalence	Theme Description
Scheduling Algorithms and Real-Time Task Management	21/50 papers (42%)	Response time analysis, slack time scheduling, and task partitioning demonstrate improved deadline adherence and resource utilization in industrial IoT contexts [11], [13], [30], [29], [31]. Moreover, integration with real-time operating systems and multi-task learning frameworks further optimizes execution under resource constraints [6], [12], [17].
Edge Computing and Latency Reduction	18/50 papers (36%)	Studies document significant latency reductions and bandwidth savings via edge analytics, scheduling optimizations, and hybrid edge-cloud architectures in smart manufacturing and industrial automation [33], [3], [43], [34], [4], [37], [49]. Task offloading and resource allocation strategies in fog environments optimize real-time performance [39], [47], [40], [36].
Interrupt Handling and Hardware-Level Optimization	15/50 papers (30%)	Novel techniques, including ENEST and Direct Register Acces, achieve substantial interrupt latency and jitter reductions, enhancing microcontroller responsiveness [7], [10], [8]. Hardware-software co-design approaches incorporating parallel context stacks and embedded AI on RISC-V platforms advance low-latency processing capabilities [44], [9], [17].
Network-Induced Delays and Communication Constraints	14/50 papers (28%)	Research addresses detection and mitigation of network bursts, bounded transmission latency analyses, and deterministic communication scheduling to ensure timely processing [1], [2], [28], [21], [42], [20]. Time-Sensitive Networking (TSN) and Ultra-Reliable Low-Latency Communication (URLLC) frameworks are investigated for reliable, low-latency industrial communication [10], [19].
Case Studies and	14/50 papers	Case studies spanning motion control systems, embedded real-time controllers, multi-MCU gateways, and CNC systems

Performance Evaluations	(28%)	document improvements in latency, throughput, and control performance under diverse scheduling and hardware configurations [2], [48], [22], [35], [15], [16]. Simulation-based studies validate task scheduling and resource allocation models [39], [41].
Resource Contention and Multicore Challenges	8/50 papers (16%)	Studies document extreme slowdowns attributable to contention and propose runtime control mechanisms such as μ TPArc to monitor and mitigate interference, enhancing timeliness in industrial IoT applications [5], [22], [6].
Dynamic Reconfiguration and Execution Flexibility	5/50 papers (10%)	Virtual machine-based approaches enable low-latency serialization/deserialization and hot reload capabilities, supporting continuous operation without significant blocking delays [23], [26].
Control Systems Integration with Scheduling	5/50 papers (10%)	Frameworks optimizing sampling frequencies, delays, and jitter alongside task scheduling enhance real-time control reliability in industrial environments [32], [15], [16], [14].

Chronological Review of Literature

The temporal evolution of research on blocking delays in microcontroller-based industrial IoT systems reflects a growing depth of understanding and increasing technological sophistication. Table 4 presents a chronological synthesis, organized into five distinct periods. Each period is characterized by dominant research paradigms, technological drivers, and conceptual innovations. This periodization shows a trajectory from foundational scheduling and control theory to integrated, multi-layered solutions. These solutions encompass edge computing, hardware acceleration, and intelligent resource management. The chronological analysis highlights how research priorities have adapted to emerging industrial automation needs, new computational architectures, and evolving networking technologies over the past two and a half decades.

Table 4. Chronological Evolution of Research Directions And Technological Focus Areas

Period	Research Direction	Characterization and Key Developments
2000–2010	Foundational Scheduling and Control Delay Analysis	Control task jitter analysis and scheduling feasibility assessments for real-time embedded controllers provided foundational insights into timing predictability requirements [14], [32], [15]. Frameworks integrated control performance objectives with scheduling decisions. These emphasized latency and jitter minimization in closed-loop systems. Early hardware-software co-design studies addressed both embedded real-time control and microcontroller task scheduling optimization [16], [31], [35].
2011–2015	Real-Time Networked Embedded Systems and Industrial Protocols	Research emphasis shifted toward networked embedded systems, addressing performance modeling for real-time Ethernet-based industrial automation platforms, including CNC systems [2]. Computational delays, communication latency, and jitter in embedded controllers operating within industrial network infrastructures became central concerns [21], [42]. Multicore microcontroller adoption introduced hardware resource

		contention challenges affecting real-time performance predictability, necessitating new analytical frameworks for timing verification in parallel processing architectures [22].
2016–2020	Edge Computing Emergence and Scheduling Innovations	Edge computing paradigms emerged as transformative approaches to latency reduction through computational redistribution toward data sources, fundamentally reshaping industrial IoT architectures [47]. Scheduling algorithm innovations for distributed real-time applications proliferated, with novel intelligent adaptive methods managing complex task sets in embedded IoT systems to enhance processor utilization and reduce deadline violations [29], [27]. Energy-efficient architectures and microarchitectural pipeline optimizations gained prominence to support mobile and resource-constrained platforms, reflecting growing recognition of power-performance trade-offs in industrial deployments [25], [24].
2021–2022	Adaptive Scheduling and Resource Management in IoT-Edge Ecosystems	Adaptive scheduling frameworks specifically tailored for resource-constrained IoT devices have matured significantly, incorporating multi-task learning techniques and weakly hard real-time models to accommodate timing violations while maintaining schedulability guarantees [30], [12]. Network packet overload detection and mitigation strategies for time-critical IoT devices addressed communication-induced blocking delays systematically [1], [19]. Continuous data stream operator scheduling in fog-edge environments optimized resource allocation for analytics workloads, while orchestration frameworks for distributed real-time applications enhanced coordination across heterogeneous edge nodes [40], [37].
2023–2025	Advanced Hardware Optimizations and Integrated Edge Architectures	Contemporary research emphasizes hardware-software co-design with RISC-V microcontrollers, featuring parallelized interrupt handling and context switching innovations that minimize blocking delays and achieve best-in-class interrupt latencies [7], [10], [8], [17]. Edge computing paradigms have achieved architectural maturity through deterministic scheduling algorithms, multicore-capable real-time operating systems, and integrated task scheduling strategies balancing computational and communication demands [28], [11], [33], [43]. Industrial automation case studies demonstrate substantial responsiveness and reliability improvements via hybrid edge-cloud frameworks, dynamic reconfiguration mechanisms, and AI-enabled inference on microcontrollers under stringent latency constraints [3], [4], [44], [49], [23]. Attention to multicore resource contention mitigation and runtime control mechanisms reflects growing architectural complexity [5], [6], [9].

CONCLUSIONS AND SUGGESTIONS

This systematic review of fifty studies shows that blocking delays critically compromise microcontroller speed and responsiveness in industrial IoT devices. These delays stem from

network-induced latencies, interrupt handling inefficiencies, and resource contention. As a result, deadline violations and real-time performance degradation occur. Although substantial progress has clarified delay magnitudes and causal mechanisms, heterogeneity and dynamism in industrial environments remain key challenges for robust mitigation across platforms and workloads.

Three primary mitigation strategies emerge. Adaptive, rate-based, and deterministic scheduling algorithms with interrupt nesting help improve deadline adherence. However, deployment in resource-constrained systems remains complex. Edge computing reduces latency using localized processing, embedded AI, and hardware optimization, achieving microsecond-level responsiveness, though multicore resource contention persists. Real-world case studies confirm that integrated approaches measurably improve latency, deadline adherence, and control accuracy.

Improving microcontroller responsiveness requires combining refined scheduling, hardware-accelerated latency reduction, and strategic edge computing. Future research should develop scalable, adaptive frameworks that address blocking delays through cross-layer optimization, standardized evaluation, and proactive mitigation for next-generation industrial automation.

REFERENCES

- [1] R. Danicki, M. Haug, I. Behnke, L. Mädje, and L. Thamsen, "Detecting and Mitigating Network Packet Overloads on Real-Time Devices in IoT Systems," in *EdgeSys 2021 - Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2021*, in EdgeSys '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 25–30. doi: 10.1145/3434770.3459733.
- [2] J. F. Wan, D. Li, Y. Q. Tu, and C. H. Zhang, "Performance analysis model for real-time Ethernet-based computer numerical control system," *J. Cent. South Univ. Technol. (English Ed.)*, vol. 18, no. 5, pp. 1545–1553, 2011, doi: 10.1007/s11771-011-0871-7.
- [3] K. K. P. Brahmaji, "Edge Computing and Analytics for IoT Devices: Enhancing Real-Time Decision Making in Smart Environments," *Int. J. Multidiscip. Res.*, vol. 6, no. 5, 2024, doi: 10.36948/ijfmr.2024.v06i05.29826.
- [4] K. Röbert, H. Bornholdt, M. Fischer, and J. Edinger, "Latency-Aware Scheduling for Real-Time Application Support in Edge Computing," in *EdgeSys 2023 - Proceedings of the 6th International Workshop on Edge Systems, Analytics and Networking, Part of EuroSys 2023*, in EdgeSys '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 13–18. doi: 10.1145/3578354.3592866.
- [5] D. Oliveira, W. Chen, S. Pinto, and R. Mancuso, "Shared Resource Contention in MCUs: A Reality Check and the Quest for Timeliness," in *Leibniz International Proceedings in Informatics, LIPIcs*, R. Pellizzoni, Ed., in Leibniz International Proceedings in Informatics (LIPIcs), vol. 298. Dagstuhl, Germany: Schloss Dagstuhl -- Leibniz-Zentrum für Informatik, 2024, pp. 5:1--5:25. doi: 10.4230/LIPIcs.ECRTS.2024.5.
- [6] M. Kim and D. Park, "Implementation of an SMP-based RTOS for Optimized Multicore Utilization on TriCore Architecture," in *2025 International Technical Conference on Circuits/Systems, Computers, and Communications, ITC-CSCC 2025*, 2025, pp. 1–6. doi: 10.1109/ITC-CSCC66376.2025.11137733.
- [7] P. Lindgren, P. Dzialo, H. Lunnikivi, and J. Ericsson, "ENEST - Efficient Interrupt Nesting for RISC-V based CPUs," in *2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference, ONCON 2023*, 2023, pp. 1–7. doi: 10.1109/ONCON60463.2023.10431132.

- [8] A. Nurmi, A. Kalache, and T. D. Hamalainen, "Hardware Solutions for Eliminating Context Switching Latency in Processor-Based Hard Real-Time Systems," in *2024 IEEE Nordic Circuits and Systems Conference, NORCAS 2024 - Proceedings*, 2024, pp. 1–6. doi: 10.1109/NorCAS64408.2024.10752471.
- [9] A. Saxena, R. Khalid, K. A. Jabbar, and L. H. A. Fezaa, "Enhancing Efficiency and Performance of Microcontrollers Through Advanced Optimization Techniques," in *Proceedings - International Conference on Technological Advancements in Computational Sciences, ICTACS 2023*, 2023, pp. 1188–1192. doi: 10.1109/ICTACS59847.2023.10390521.
- [10] C. Li, T. Li, J. Li, W. Fu, and B. Wang, "DRA: Ultra-Low Latency Network I/O for TSN Embedded End-Systems," in *IEEE International Workshop on Quality of Service, IWQoS*, 2023, pp. 1–10. doi: 10.1109/IWQoS57198.2023.10188749.
- [11] T. Springer and P. Zhao, "Design and Implementation of a Real-Time Rate-Based Task Scheduler for Real-Time Operating Systems: A Case Study with VxWorks," in *CS & IT Conference Proceedings*, 2024, pp. 01–20. doi: 10.5121/csit.2024.142301.
- [12] M. H. Bin Kamilin, M. A. Bin Ahmadon, and S. Yamaguchi, "Multi-task learning-based task scheduling switcher for a resource-constrained iot system†," *Inf.*, vol. 12, no. 4, 2021, doi: 10.3390/info12040150.
- [13] S. A. Khajeh, M. Saberikamarposhti, and A. M. Rahmani, "Real-Time Scheduling in IoT Applications: A Systematic Review," *Sensors*, vol. 23, no. 1, 2023, doi: 10.3390/s23010232.
- [14] P. Balbastre, I. Ripoll, and A. Crespo, "Control tasks delay reduction under static and dynamic scheduling policies," in *Proceedings - 7th International Conference on Real-Time Computing Systems and Applications, RTCSA 2000*, 2000, pp. 522–526. doi: 10.1109/RTCSA.2000.896437.
- [15] Y. Wu, E. Bini, and G. Buttazzo, "A framework for designing embedded real-time controllers," in *Proceedings - 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2008*, 2008, pp. 303–311. doi: 10.1109/RTCSA.2008.22.
- [16] S. Samii, A. Cervin, P. Eles, and Z. Peng, "Integrated scheduling and synthesis of control applications on distributed embedded systems," in *Proceedings - Design, Automation and Test in Europe, DATE*, 2009, pp. 57–62. doi: 10.1109/date.2009.5090633.
- [17] Y. Wang, X. Zhao, T. Chong, and X. Liu, "Embedded Microprocessor Extension Design and Optimization for Real-Time Edge Computing," *Wirel. Commun. Mob. Comput.*, vol. 2022, no. 1, p. 5705184, 2022, doi: 10.1155/2022/5705184.
- [18] F. Bender, J. J. Brune, N. L. Keutel, I. Behnke, and L. Thamsen, "Pieres: A playground for network interrupt experiments on real-time embedded systems in the IoT," in *ICPE 2021 - Companion of the ACM/SPEC International Conference on Performance Engineering*, in ICPE '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 81–84. doi: 10.1145/3447545.3451189.
- [19] K. Brun-Laguna, P. Minet, and Y. Tanaka, "Optimized scheduling for time-critical industrial IoT," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9014218.
- [20] K. Li, P. Zhu, Y. Wang, J. Wang, and X. You, "Scheduling of Time-Triggered Traffic for Deterministic URLLC in Industrial Automation," *IEEE Internet Things J.*, vol. 11, no. 16, pp. 26552–26567, 2024, doi: 10.1109/JIOT.2024.3422400.
- [21] Y. H. Song, Suk-Lee, H. J. Kim, and K. C. Lee, "Computational delay and real-time performance of industrial network," in *ICCAS 2015 - 2015 15th International Conference on Control, Automation and Systems, Proceedings*, 2015, pp. 735–737. doi:

- 10.1109/ICCAS.2015.7364716.
- [22] H. Youness, M. Moness, and M. Khaled, "MPSoCs and multicore microcontrollers for embedded PID control: A detailed study," *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2122–2134, 2014, doi: 10.1109/TII.2014.2355036.
- [23] E. Marevac, E. Kadušić, N. Živić, N. Buzadija, and S. Lemeš, "Framework Design for the Dynamic Reconfiguration of IoT-Enabled Embedded Systems and 'On-the-Fly' Code Execution," *Futur. Internet*, vol. 17, no. 1, 2025, doi: 10.3390/fi17010023.
- [24] M. Geier, T. Burghart, M. Hackl, and S. Chakraborty, "In situ latency monitoring for heterogeneous real-time systems," in *Proceedings - 32nd International Conference on VLSI Design, VLSID 2019 - Held concurrently with 18th International Conference on Embedded Systems, ES 2019*, 2019, pp. 275–280. doi: 10.1109/VLSID.2019.00066.
- [25] C. A. Tanase, "Reducing Energy Consumption in Microcontroller-based Systems with Multipipeline Architecture," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 16–24, 2020, doi: 10.14569/IJACSA.2020.0111203.
- [26] Q. Yang, "Embedded System Based on Task Interval Data Interaction Model," *Int. J. High Speed Electron. Syst.*, vol. 34, no. 2, 2025, doi: 10.1142/S0129156424400895.
- [27] P. Vasu, H. Chouhan, and N. Naik, "Design and implementation of optimal soft-programmable logic controller on multicore processor," in *2017 International Conference on Microelectronic Devices, Circuits and Systems, ICMDCS 2017*, 2017, pp. 1–4. doi: 10.1109/ICMDCS.2017.8211691.
- [28] Y. Hu *et al.*, "Deterministic Scheduling and Network Structure Optimization for Time-Critical Computing Tasks in Industrial IoT," 2025. doi: 10.1109/ton.2025.3587916.
- [29] S. Malik, S. Ahmad, I. Ullah, D. H. Park, and D. H. Kim, "An adaptive emergency first intelligent scheduling algorithm for efficient task management and scheduling in hybrid of hard real-time and soft real-time embedded IoT systems," *Sustain.*, vol. 11, no. 8, 2019, doi: 10.3390/su11082192.
- [30] H. Choi, H. Kim, and Q. Zhu, "Toward Practical Weakly Hard Real-Time Systems: A Job-Class-Level Scheduling Approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6692–6708, 2021, doi: 10.1109/JIOT.2021.3058215.
- [31] B. Ness and G. Karsai, "High precision automatic scheduling of periodic task sets for microcontrollers," in *Proceedings of the 46th Annual Southeast Regional Conference on XX, ACM-SE 46*, in ACMSE '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 1–6. doi: 10.1145/1593105.1593107.
- [32] C. Lozoya, M. Velasco, and P. Martí, "The one-shot task model for robust real-time embedded control systems," *IEEE Trans. Ind. Informatics*, vol. 4, no. 3, pp. 164–174, 2008, doi: 10.1109/TII.2008.2002702.
- [33] P. Fara, G. Serra, and F. Aromolo, "Bounded transmission latency in real-time edge computing: a scheduling analysis," in *Proceedings - 2023 26th Euromicro Conference on Digital System Design, DSD 2023*, 2023, pp. 618–625. doi: 10.1109/DSD60849.2023.00090.
- [34] D. Ko, J. Jeon, and S. Kang, "A Research on Low Latency Motion Control System using Real-time Scheduling in Edge Server," in *International Conference on Information Networking*, 2023, pp. 537–540. doi: 10.1109/ICOIN56518.2023.10048964.
- [35] F. C. Braescu, L. Ferariu, and C. Lazar, "OSEK-based multiple controllers with schedule feasibility self-testing," in *SPEEDAM 2010 - International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 2010, pp. 1237–1242. doi: 10.1109/SPEEDAM.2010.5542060.

- [36] L. Yang, X. Wang, Z. Liu, Y. Liu, and L. Fan, "Real-time processing and optimization strategies for IoT data streams," *Appl. Math. Nonlinear Sci.*, vol. 9, no. 1, 2024, doi: 10.2478/amns-2024-2978.
- [37] Z. Zhang, W. Sun, and Y. Yu, "Research on Intelligent Scheduling Mechanism in Edge Network for Industrial Internet of Things," *Secur. Commun. Networks*, vol. 2022, no. 1, p. 5358873, 2022, doi: 10.1155/2022/5358873.
- [38] K. Takado, T. Yokoyama, and M. Yoo, "A Real-Time Operating System for Physical and Logical Time-Triggered Distributed Computing," in *ACM International Conference Proceeding Series*, in APIT '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 78–85. doi: 10.1145/3651623.3651635.
- [39] J. Wang and D. Huang, "Effective Task Scheduling for Intelligent Manufacturing With Fog Computing," in *IET Conference Proceedings*, 2024, pp. 118–126. doi: 10.1049/icp.2024.2720.
- [40] P. Ntumba, N. Georgantas, and V. Christophides, "Scheduling Continuous Operators for IoT edge Analytics with Time Constraints," in *Proceedings - 2022 IEEE International Conference on Smart Computing, SMARTCOMP 2022*, 2022, pp. 78–85. doi: 10.1109/SMARTCOMP55677.2022.00026.
- [41] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie, "Toward Computing Resource Reservation Scheduling in Industrial Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8210–8222, 2021, doi: 10.1109/JIOT.2020.3044057.
- [42] P. Lindgren, J. Eriksson, M. Lindner, A. Lindner, D. Pereira, and L. M. Pinho, "Response time for IEC 61499 over Ethernet," in *Proceeding - 2015 IEEE International Conference on Industrial Informatics, INDIN 2015*, IEEE, Jul. 2015, pp. 1206–1212. doi: 10.1109/INDIN.2015.7281907.
- [43] L. Orciari, D. Raggini, and A. Tilli, "Taming Edge Computing for Hard Real-Time Advanced Control of Mechatronic Systems," *IEEE Trans. Ind. Informatics*, vol. 20, no. 8, pp. 9898–9906, 2024, doi: 10.1109/TII.2024.3390608.
- [44] K. G. Sundararaj and G. C. Megharaj, "IoT Enabled Smart Sensing Systems Using RISC V Based Microcontrollers and Embedded AI," in *Smart Microcontrollers and FPGA Based Architectures for Advanced Computing and Signal Processing*, RADemics Research Institute, 2025, pp. 131–158. doi: 10.71443/9789349552425-05.
- [45] M. V. Manjula, "Optimizing Edge Computing for Real-Time Data Processing in IoT Networks: A Comparative Study of Lightweight Algorithms," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. 8, pp. 895–899, Aug. 2025, doi: 10.22214/ijraset.2025.73662.
- [46] C. H. Chen, M. Y. Lin, and C. C. Liu, "Edge Computing Gateway of the Industrial Internet of Things Using Multiple Collaborative Microcontrollers," *IEEE Netw.*, vol. 32, no. 1, pp. 24–32, Jan. 2018, doi: 10.1109/MNET.2018.1700146.
- [47] P. Muralidhara and V. Janardhan, "Edge and cloud integration: Optimizing Latency and Resource Allocation for IoT Applications," *Int. J. Eng. Comput. Sci.*, vol. 5, no. 7, pp. 17388–17406, Jul. 2016, doi: 10.18535/ijecs/v5i7.33.
- [48] S. B. Kamatchi, V. N. Agme, S. Premkumar, K. Prasad, D. G. V, and I. Gugan, "Enhancing Microcomputer Edge Computing for Autonomous IoT Motion Control," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 3, pp. 58–67, Apr. 2023, doi: 10.17762/ijritcc.v11i3.6202.
- [49] V. S. Anooja, M. Krishnakumar, S. B. Sangeetha, and W. Rajaian, "Enhancing Intelligent Connectivity Through Embedded IoT Systems for Real-Time Applications," *ICTACT J. Microelectron.*, vol. 11, no. 1, pp. 2058–2063, Apr. 2025, doi: 10.21917/ijme.2025.0348.

- [50] M. Walker, S. Wörtz, M. Neubauer, A. Lechler, and A. Verl, "Scheduling for the Orchestration of Distributed Real-Time Applications," in *2025 IEEE 8th International Conference on Industrial Cyber-Physical Systems (ICPS)*, IEEE, May 2025, pp. 01–08. doi: 10.1109/icps65515.2025.11087855.