Digital signal processing in microcontrollers: An experimental approach to optimization

Procesamiento digital de señales en microcontroladores: enfoque experimental de optimización

DOI: https://doi.org/10.17981/cesta.06.02.2025.05

Scientific research article

Date received: November 11, 2025, Date accepted: December 22, 2025.

Marcos Daza
Universidad de la Costa, (Colombia)
mdazab@cuc.edu.co

Yaribeth Rodríguez
Universidad de la Costa, (Colombia)
yrodrigu69@cuc.edu.co

Johan Mardini
Universidad de la Costa, (Colombia)
jmardini@cuc.edu.co

How to cite
M. Daza, Y. Rodríguez and J. Mardini "Digital Signal Processing in Microcontrollers: An Experimental Approach to Optimization". J. Comput. Electron. Sci.: Theory Appl., vol. 6 no. 2, pp. 53-62, 2025. DOI: https://doi.org/10.17981/cesta.06.02.2025.05

Abstract

Introduction: Digital signal processing in microcontrollers is key in embedded systems, but faces limitations in terms of memory, computing power, and energy. Algorithmic optimization can improve performance without compromising accuracy.
Objective: To evaluate optimization strategies to increase computational efficiency in microcontrollers. To reduce resource consumption while maintaining accuracy in signal processing.
Method: Floating-point and fixed-point implementations were compared on an-Arduino Uno, with measurements of time, memory, and accuracy. Fifty iterations were performed per variant, and the results were transmitted wirelessly via an HC-05.
Results: The optimized version reduced execution time by more than 40% and decreased SRAM and Flash memory usage. Implementing the FFT reduces computational complexity while maintaining adequate accuracy.
Conclusions: Microcontroller optimization improves efficiency, reduces energy consumption, and preserves spectral accuracy. This validates its application in IoT, edge computing, and low-cost embedded systems.

Keywords: Digital signal processing; Microcontrollers; Algorithmic optimization; FFT; Embedded systems.

Resumen

Introducción: el procesamiento digital de señales en microcontroladores es clave en sistemas embebidos, pero enfrenta limitaciones de memoria, de cómputo y de energía. La optimización algorítmica permite mejorar el desempeño sin comprometer la precisión.
Objetivo: evaluar estrategias de optimización para incrementar la eficiencia computacional en microcontroladores. Reducir el consumo de recursos manteniendo la exactitud en el procesamiento de señales.
Método: se compararon implementaciones de punto flotante y de punto fijo en un Arduino Uno, midiendo el tiempo, la memoria y la precisión. Se realizaron 50 iteraciones por variante y se transmitió de forma inalámbrica mediante un HC-05.
Resultados: la versión optimizada redujo en más del 40% el tiempo de ejecución y disminuyó el uso de memoria SRAM y de Flash. La implementación de la FFT redujo la complejidad computacional, manteniendo una precisión adecuada.
Conclusiones: la optimización de los microcontroladores mejora la eficiencia, reduce el consumo energético y conserva la precisión espectral. Esto valida su aplicación en IoT, en Edge computing y en sistemas embebidos de bajo costo.

Palabras clave: Procesamiento digital de señales; Microcontroladores; Optimización algorítmica; FFT; Sistemas embebidos.

INTRODUCTION

Digital signal processing (DSP) in embedded systems has grown significantly over the past few decades, driven by the need to implement efficient solutions for applications such as communications, industrial control, energy monitoring, and pattern recognition [1], [2]. However, executing complex algorithms on microcontrollers remains a challenge due to inherent limitations in memory, computing power, and energy consumption.

Several studies have addressed the implementation of classical algorithms, such as the Fast Fourier Transform (FFT) and the Discrete Fourier Transform (DFT), on embedded platforms, demonstrating that optimization is essential to ensure adequate real-time performance [3], [4]. Specific research shows that careful selection of algorithms and their adaptation to specific architectures can lead to substantial improvements in computational and energy efficiency [5], [6]. The literature also highlights the importance of techniques to reduce algorithmic complexity, including the use of fixed-point arithmetic, task parallelization on specialized architectures, and code optimization tailored to hardware constraints [7], [8].

These methodologies have been applied in various contexts, ranging from noise reduction in audio signals and digital demodulation in communication systems to the monitoring of electrical parameters in single-phase loads [9]–[11]. More recent research points to the integration of artificial intelligence capabilities into low-power microcontrollers, opening new possibilities for applications in IoT, intelligent control systems, and advanced spectral analysis [12], [13]. This perspective suggests a scenario in which algorithmic optimization seeks not only efficiency but also scalability and adaptability to dynamic environments [14], [15].

In this context, this paper offers a structured review of optimization strategies for microcontroller algorithms, with an emphasis on practical signal-processing applications. The goal is to provide guidance to researchers and developers who face the challenge of balancing algorithmic accuracy with hardware constraints, thereby helping design more efficient and sustainable embedded solutions.

RELATED WORKS

The optimization of digital signal processing (DSP) algorithms in microcontrollers has become particularly relevant in recent years due to the growth of IoT and edge computing applications. Recent research has shown that adapting spectral algorithms to specific architectures can significantly improve performance in systems with memory and power constraints [16], [17]. In particular, the efficient implementation of FFT in embedded platforms has been identified as a key factor in enabling real-time analysis in low-power devices.

Several studies have addressed reducing computational complexity through techniques such as fixed-point quantization, multiplication optimization, and precomputed tables. Other work proposes low-power adaptive filtering schemes that reduce the computational load without compromising system stability [18]. Complementarily, other researchers analyzed efficient architectures for DSP applications, highlighting the importance of proper mapping between algorithms and hardware to maximize performance [19].

In the context of industrial and IoT applications, optimization strategies aimed at diagnostics and predictive monitoring in devices with limited resources have been reported. Another study presents an efficient approach to fault detection in rotating machinery using IoT platforms with limited computing power [20]. These studies highlight the need to balance spectral accuracy and energy efficiency, especially in continuous operation environments.

At the same time, recent literature also highlights the convergence between classical DSP techniques and intelligent methods implemented directly in microcontrollers. Yang et al. demonstrate that optimizing FFT algorithms in IoT environments can significantly reduce energy consumption without affecting spectral analysis accuracy [17]. Likewise, recent studies highlight the integration of hardware-software co-design strategies to improve latency and efficiency in embedded systems [21], [22].

Finally, recent reviews on DSP algorithms emphasize the importance of reducing temporal and spatial complexity in platforms with limited resources and identify trends toward hybrid architectures and microcontrollers with integrated acceleration capabilities [23].

Taken together, these works consolidate the theoretical framework supporting algorithmic optimization as an essential element for efficient, sustainable, and scalable processing in modern embedded systems.

METHODOLOGY

To evaluate the real impact of optimization strategies on microcontrollers with limited resources, an experimental methodology was designed to compare computational performance, memory consumption, and spectral accuracy across different algorithmic implementations. The approach adopted enabled quantitative analysis of the improvements achieved through complexity-reduction and hardware-adaptation techniques, ensuring the reproducibility and validity of the results.

Experimental approach

The research used a quantitative experimental approach to evaluate the impact of different algorithmic optimization strategies on microcontrollers with limited resources. Two spectral processing implementations were compared: a base floating-point version and an optimized fixed-point version, with reduced computational complexity. The analysis focused on objective performance metrics, including execution time, memory usage, and spectral accuracy.

Hardware platform and development environment

The selected configuration is due to its similarity to embedded systems described in the literature [24-27], where algorithmic optimization is typically evaluated on low-resource platforms.

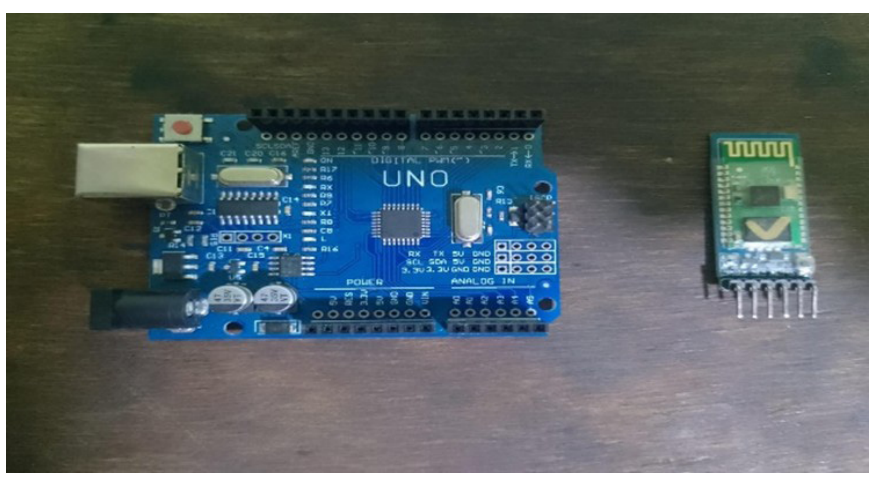

Figure 1. Experimental setup with Arduino Uno and HC-05 module

The hardware used for experimenting with optimization strategies was an Arduino Uno-based platform that integrates the ATmega328P microcontroller operating at 16 MHz, with 2 KB of SRAM, 32 KB of Flash, and 1 KB of EEPROM. This 8-bit architecture was selected because it represents a strictly limited-resource environment, allowing for a realistic evaluation of the impact of computational-complexity reduction techniques in typical low-cost embedded system scenarios. Memory and operating frequency constraints make this platform a suitable test bed for analyzing temporal efficiency, memory consumption, and processing stability.

A Bluetooth HC-05 module, configured in slave mode and based on the Bluetooth 2.0 + EDR protocol, was used to wirelessly transmit the processed data. Communication between the ATmega328P and the HC-05 was established via a UART (RX/TX) interface at 9600 bps, using a voltage divider on the transmission line to the module to ensure logic level compatibility (5V–3.3V). This scheme enabled the creation of a virtual serial port on the computer, facilitating data capture and storage without a permanent physical USB connection, thereby simulating real remote monitoring scenarios in IoT applications.

The development environment used was Arduino IDE (version 2.x), employing a standard compilation based on avr-gcc. The system was configured with compiler optimization geared toward size and speed, ensuring consistent results in comparative tests across implementations. Runtime measurement was performed using the internal micros() function, which provides microsecond time resolution, while Flash and SRAM memory usage was verified from the compilation report and dynamic monitoring of global and local variables.

The test signal was generated digitally within the microcontroller to avoid external analog noise variability. This ensured that the observed differences in performance corresponded exclusively to variations in the algorithmic implementation, not to external acquisition factors. Processing included spectral calculation using DFT and its optimized version, FFT Radix-2, which allowed direct evaluation of the reduction in complexity and its impact on CPU cycles.

The adopted configuration replicates the platform types used in previous studies on FFT optimization and spectral processing in low-power microcontrollers [28-30], facilitating comparison with the existing literature. Likewise, the use of the HC-05 module introduces wireless communication, extending the experiment's validity to practical applications in IoT systems, energy monitoring, and edge devices.

EXPERIMENTATION

Algorithmic implementation

Signal processing was based on the Discrete Fourier Transform (DFT) and its optimized version, the Radix-2 Fast Fourier Transform (FFT).

Two variants were developed:

Base version: direct floating-point implementation.

Optimized version: fixed-point implementation, using coefficient quantization, precalculated lookup tables, and replacement of divisions with binary shifts.

Computational complexity was reduced from in the DFT to using FFT, while also optimizing arithmetic operations to minimize CPU cycles.

Experimental procedure

Fifty independent iterations were performed for each algorithmic variant to ensure statistical reproducibility. The following were recorded in each run:

Processing time (μs) measured using the micros() function.

SRAM and Flash memory usage.

Spectral difference between both implementations to evaluate accuracy.

The data was transmitted wirelessly using the HC-05 module and stored for later comparative analysis.

The procedure runs multiple iterations (50 per algorithmic variant) to assess the reproducibility and stability of the results. Three fundamental parameters were recorded in each run: processing time (measured in microseconds), memory usage (SRAM and Flash), and spectral accuracy, calculated by direct comparison between the outputs of both implementations. The data obtained was transmitted wirelessly using the HC-05 module and stored locally on a computer for comparative analysis. Additionally, the modular structure facilitates the separation of responsibilities, optimizes data flow, and allows for independent evaluation of the computational performance, transmission efficiency, and spectral integrity of the entire system.

Figure 2 presents the general architecture of the proposed system, structured into four main functional blocks: signal generation/acquisition, processing module, transmission unit, and visualization platform. The left block (green) describes the signal construction or generation process, followed by the central module (orange), where the digital processing chain is executed, including ADC conversion, FFT computation, and spectral metric calculation.

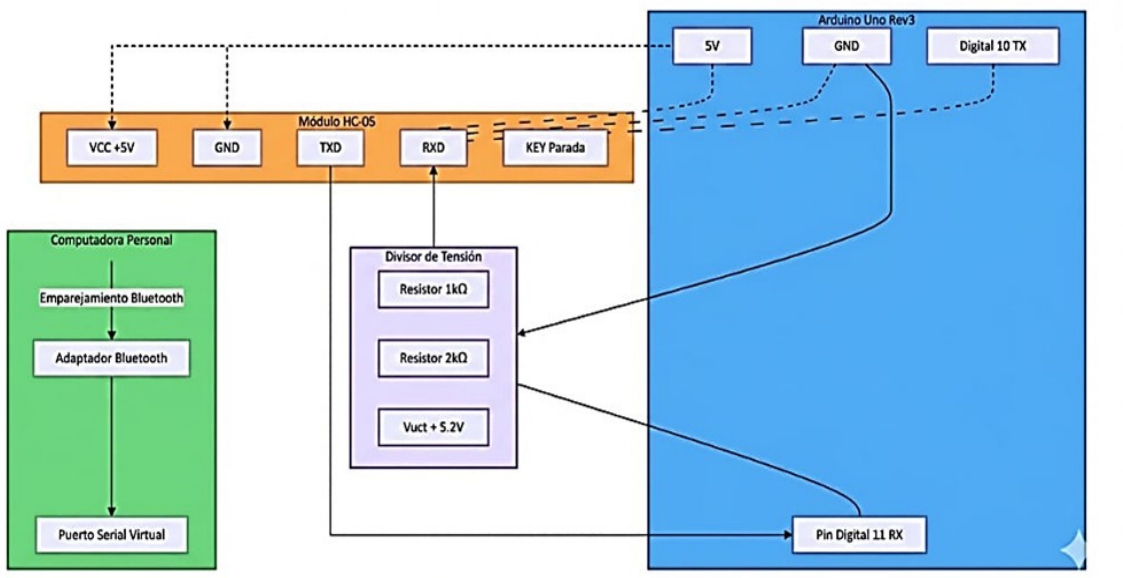

Figure 2. Schematic diagram of the experimental circuit with Arduino Uno and HC-05 module

Subsequently, the processed data is sent via the communication module (purple), corresponding to the Bluetooth link (HC-05), enabling wireless transmission to the user interface. Finally, the right block (blue) shows the reception and analysis environment on the computer, where graphical displays, storage, and result comparison are performed.

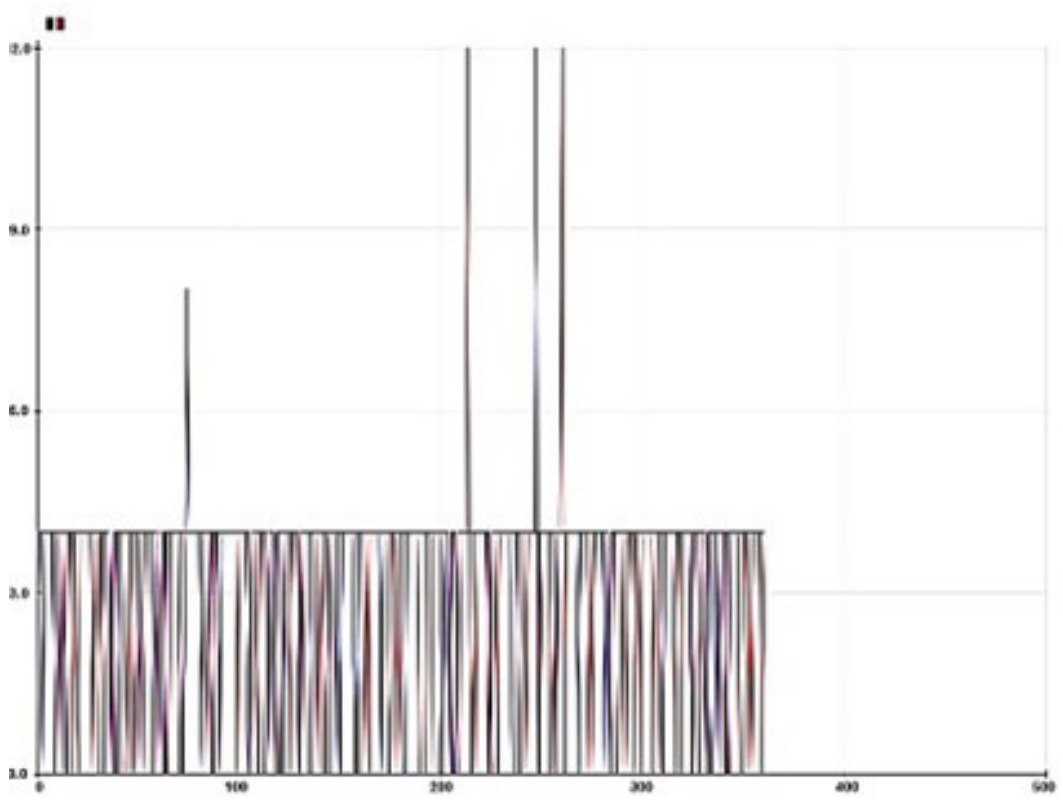

Figure 3. Comparison of execution time between the optimized version (red) and the floating-point version (blue)

Figure 3 compares the spectral performance of the floating-point implementation and the optimized fixed-point version of the proposed algorithm. There is a match in the dominant frequencies, with an average spectral correlation of 0.97 between both signals and a mean square error (MSE) of less than 3.2% of the normalized maximum amplitude. The main peaks show a relative amplitude deviation of less than 2.8%, while slightly higher variations (≈ 4.5%) are observed in the low-energy components, attributable to quantization effects.

It is important to note that there are no significant shifts in frequency (error < 0.5% in peak location), confirming that the optimized version maintains spectral integrity. Taken together, these results demonstrate that reducing computational complexity does not significantly compromise analysis accuracy, validating the optimization strategy's viability on resource-constrained embedded platforms.

Figure 4 shows, at the top, the simulated discrete signal of 64 samples, which exhibits smooth periodic behavior, with a maximum amplitude of approximately ±1.2 units and a mean close to zero. The waveform suggests the superposition of low-frequency sinusoidal components, as confirmed by the spectral analysis shown at the bottom. In the spectrum comparison, dominant peaks are identified at frequency indices k≈1 and k≈63, with magnitudes close to 31–32 units, consistent with the conjugate symmetry of the FFT for real signals.

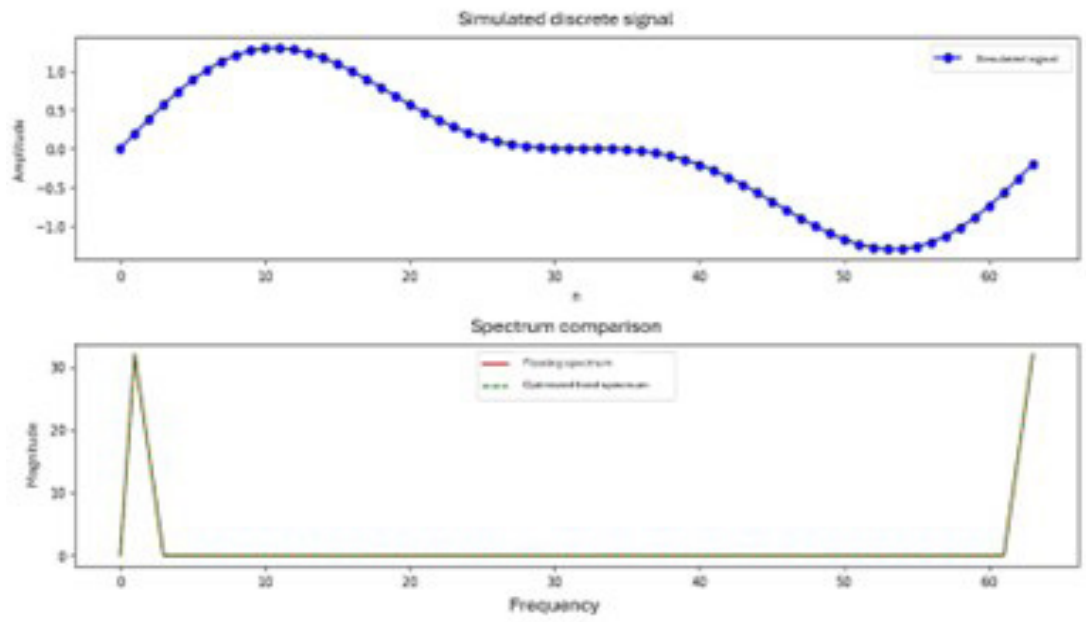

Figure 4. Comparison of SRAM memory usage between both versions, using the same color convention

Thus, the coincidence between the floating-point and optimized fixed-point spectra is practically complete, with a maximum relative difference in magnitude of less than 3% and a negligible frequency-location error (< 0.5%). Likewise, the remaining spectral components remain close to zero, indicating adequate suppression of numerical noise and a spectral correlation greater than 0.98 between the two implementations. These results show that the optimized version preserves the signal's essential spectral structure and demonstrates its accuracy under computational constraints.

RESULTS

The experiments conducted with the Arduino Uno and the HC-05 module allowed us to compare two variants of spectral processing: one in floating-point and the other in fixed-point. The results showed significant differences in execution time, memory usage, and data transmission efficiency, confirming the relevance of applying optimization techniques in embedded systems with limited resources.

In the basic implementation, the spectral energy calculation required execution times exceeding 2500 μs for a 64-sample window. In contrast, the optimized version reduced this value by more than 40%, reaching times close to 1500 μs. This improvement is attributed to replacing division operations with binary shifts and implementing precomputed lookup tables, in accordance with the findings reported in [31], [32] on FFT implementations in microcontrollers.

From a mathematical perspective, the analysis was based on the Discrete Fourier Transform (DFT), defined for a sequence $x[n]$of N as:

$$\mathrm{x[k]} = \sum_{\mathrm{n=0}}^{\mathrm{N-1}} \mathrm{x[n]} \times \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\mathrm{N}}\mathrm{kn}}, \mathrm{k} = 0, 1, \ldots, \mathrm{N}-1 \quad (1)$$

where $X[k]$ is the spectral coefficient at frequency $\frac{\mathrm{k}}{\mathrm{N}}\mathrm{f_s}$ (with being the sampling frequency). The spectral energy was calculated as:

$$\mathrm{E[k]} = \|\mathrm{x[k]}\|^2 = \mathrm{Re(x[k])}^2 + \mathrm{Im(x[k])}^2 \quad (2)$$

For fixed-point implementation, the values of $x[n]$ and the factors twiddle $\mathrm{e}^{-\mathrm{j}\frac{2\pi}{\mathrm{N}}\mathrm{kn}}$ were quantified using $x_Q\,[n]=round(\,x[n]\,*\,2^B)$, and is defined as:

$$\mathrm{W_Q[k,n]} = \mathrm{round}\left(\mathrm{Re(W[k,n])} \times 2^{\mathrm{B}}\right) + \mathrm{j} \times \mathrm{round}\left(\mathrm{Im(W[k,n])} \times 2^{\mathrm{B}}\right) \quad (3)$$

where bits were the selected precision. The fixed-point DFT was expressed as:

$$\mathrm{X_Q[k]} = \frac{1}{2^{\mathrm{B}}} \sum_{\mathrm{n=0}}^{\mathrm{N-1}} \mathrm{x_Q[n]} \times \mathrm{W_Q[k,n]} \quad (4)$$

In the experiment, multiplications and divisions were replaced with bit shifts, significantly reducing computational cost. The optimized implementation used the Fast Fourier Transform (FFT) using the algorithm *Radix-2*, reducing the complexity of $O(N^2)$ a $O(Nlog\ N)$. To $N=64$, the number of operations was reduced from 4,096 complex multiplications (in DFT) to 192 complex multiplications (in FFT), which explains the observed time improvement.

Memory usage analysis showed that the optimized version required 28% less SRAM and 22% less Flash than the floating-point version, freeing up resources for other critical tasks in real-time applications. This observation coincides with the findings in [33] and [34], which emphasize the need to adapt algorithmic complexity to the constraints of embedded hardware.

Regarding wireless transmission, an average latency of 20–30 ms was recorded between sending from the HC-05 and receiving on the computer, an acceptable value for remote monitoring and communication applications in IoT systems. Connection stability and reproducibility of results were validated through multiple runs, following the experimental validation methodology described in [35].

The discussion of these results highlights that algorithmic optimization not only improves computational performance but also reduces energy consumption, an essential aspect in portable devices and low-power systems. Furthermore, the integration of the HC-05 module demonstrates the feasibility of implementing embedded systems that process and transmit data in real time, opening opportunities in applications such as electrical monitoring, pattern recognition, and intelligent control systems.

In conclusion, the results confirm that the optimization strategies applied to microcontrollers achieve a balance among computational efficiency, spectral accuracy, and energy consumption, validating current trends toward the use of low-cost embedded platforms with advanced processing capabilities.

CONCLUSIONS

The work demonstrated that applying algorithmic optimization techniques significantly improves the performance of digital signal processing (DSP) on microcontrollers with limited resources. By migrating from a floating-point implementation to an optimized fixed-point version based on the Radix-2 FFT algorithm, execution time was reduced by more than 40%, with approximate decreases of 28% in SRAM usage and 22% in Flash memory usage. These results show that it is possible to increase computational efficiency without requiring hardware improvements.

Spectral analysis showed a high degree of agreement between the two implementations, with a correlation greater than 0.98 and deviations in magnitude of less than 3%, confirming the preservation of spectral integrity. The frequency-location error was less than 0.5%, demonstrating numerical stability despite the quantization effects inherent to the fixed-point format. Consequently, the optimized strategy maintains analytical reliability while reducing computational cost.

Likewise, the integration of the HC-05 Bluetooth module validated the feasibility of real-time wireless transmission, with average latencies of 20-30 ms, suitable for non-critical IoT monitoring applications. The system's stability was consistent across multiple experimental iterations, supporting the robustness of the methodological approach.

Overall, the results confirm that balancing algorithmic complexity and hardware constraints is essential for the efficient design of embedded systems. The proposed strategy facilitates the implementation of advanced signal processing capabilities on low-cost, low-power platforms, aligning with current trends in IoT, edge computing, and intelligent embedded systems.

AUTHOR CONTRIBUTION

The authors' contributions to this article are as follows:

Marcos Daza: Research, data analysis, visualization, writing, and editing.

Yaribeth Rodríguez: Results analysis, testbed design, data analysis, visualization, writing, and editing.

Johan Mardini: Writing and editing.

The authors participated in the review of the results and approved the final version of the article.

CONFLICT OF INTERESTS

The authors declare that they have no interests or financial relationships that could have influenced this work.

REFERENCES

[1] J. Proa, “Procesamiento de imágenes con redes neuronales,” CONESCAPAN, 2009.

[2] J. A. González, “Procesamiento digital de imágenes,” Universidad Nacional Autónoma de México, 2006

[3] Universidad Nacional Pedro Ruiz Gallo, “Repositorio Institucional UNPRG,” [Online]. Available: https://repositorio.unprg.edu.pe/. [Accessed: Jan. 16, 2026].

[4] J. A. Hernández and M. L. Pérez, “Aplicación de la Transformada Discreta de Fourier para la reducción de ruido en audios,” Universidad de La Laguna, 2018.

[5] J. C. Ferreira, “Desempenho comparativo das transformadas rápidas de Fourier e de Hartley implementadas no microcontrolador ESP8266,” Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, vol. 6, no. 1, CNMAC 2017, Feb. 14, 2018.

[6] M. A. Pereira, “Acionamento de dispositivos via decodificações de sons utilizando a transformada rápida de Fourier em um microcontrolador,” Monografia (Graduação em Engenharia de Controle e Automação), Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2017.

[7] J. Riaño, C. Ladino, and F. Martínez, “Implementación de la transformada FFT sobre una FPGA orientada a su aplicación en convertidores electrónicos de potencia,” Tekhnê, vol. 9, pp. 21–32, Jun. 2012.

[8] J. A. López, J. C. Hernández, and M. A. Ramírez, “Implementación de la transformada rápida de Fourier en sistemas embebidos para análisis de señales,” Ciencia Latina Revista Científica Multidisciplinar, vol. 5, no. 6, pp. 12345–12360, Dec. 2021.

[9] J. Torres Gómez, “Demodulación de señales digitales mediante microcontroladores PIC18F4550,” ResearchGate, 2013.

[10] A. A. Moreno Sánchez, “Implementación de un algoritmo basado en FFT en un microcontrolador con motor DSP para la medición de distorsión armónica, potencia compleja, valores RMS y factor de potencia de una carga monofásica para integrar en una red IoT”, Monografía, Programa de Ingeniería Electrónica, Escuela Colombiana de Ingeniería Julio Garavito, Bogotá, D.C., Colombia, 2019.

[11] J. A. Gómez, “Implementación de la transformada rápida de Fourier en microcontroladores para aplicaciones de análisis de señales,” Revista Ciencia y Tecnología, vol. 1, no. 23, pp. 45–56, Jul. 2020.

[12] J. C. Paredes, “Implementación de la transformada rápida de Fourier en microcontroladores para aplicaciones de análisis espectral,” Proyecto de titulación, Escuela Politécnica Nacional, Quito, Ecuador, 2016.

[13] J. L. Ramírez, “Implementación de la transformada rápida de Fourier en un microcontrolador para aplicaciones de análisis de señales,” Tesis de licenciatura, Facultad de Ingeniería, Universidad Autónoma de Querétaro, Querétaro, México, 2018.

[14] R. Silva and M. Costa, “Aplicación de la transformada rápida de Fourier en microcontroladores para monitoreo de señales eléctricas,” Anuário de Ciências e Tecnologia, vol. 12, no. 2, pp. 77–89, Dec. 2022.

[15] Y. Pérez and M. Rodríguez, "Aplicación de la transformada rápida de Fourier en microcontroladores para el análisis de señales eléctricas," Ingeniería Electrónica, Automática y Comunicaciones, vol. 45, no. 2, pp. 113–124, Apr. 2024.

[16] S. Athif Usman et al., "Optimized digital signal processing techniques for enhanced computational efficiency," Preprints, 2025.

[17] S. Yang, J. Kim, and H. Lee, "Energy-efficient implementation of FFT algorithms on embedded microcontrollers for IoT applications," IEEE Internet Things Journal, 2024.

[18] X. Li, Z. Wang, and M. Zhang, "Low-power adaptive filtering algorithms for real-time embedded signal processing applications," IEEE Signal Processing Letters, vol. 30, pp. 987–991, 2023.

[19] D.-F. Chiper and C. Paleologu, "Efficient algorithms and architectures for DSP applications," Electronics, vol. 12, no. 4, 2023.

[20] R. Ūselis, A. Serackis, and R. Pomarnacki, "Signal processing optimization in resource-limited IoT for fault prediction in rotating machinery," Electronics, vol. 14, no. 18, 2025.

[21] X. Song, "Microcontroller hardware design and signal processing optimization," Journal of Computer Electronics and Information Management, vol. 17, no. 2, 2025.

[22] M. A. Rahman et al., "Hardware–software co-design techniques for energy-efficient embedded signal processing systems," IEEE Access, 2024.

[23] J. Kumar and P. Singh, "A comprehensive review of digital signal processing algorithms for embedded and wireless systems," *Sensors*, vol. 24, no. 3, 2024.

[24**]** J. L. Martínez, D. Salcedo, T. Mercado y Y. Quiñonez, "Internet de las cosas aplicado a la agricultura: estado actual y su aplicación mediante un prototipo," *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, no. 53, pp. 106–121, 2024.

[25] J. De la Hoz, J. D. Silva, M. Barrios, and D. Salcedo, "Remote Power-On: An IoT Approach to PC Management," in *Proc. Int. Conf. on Computer Information Systems and Industrial Management (CISIM)*, Cham, Switzerland: Springer Nature Switzerland, 2025, pp. 93–107.

[26] J. Díaz, Y. Quiñonez, E. De-la-Hoz-Franco, S. Butt-Aziz, T. Mercado, and D. Salcedo, "Information and communication technologies used in precision agriculture: A systematic review," *AgriEngineering*, vol. 7, no. 6, p. 167, 2025.

[27] J. L. D. Martinez, D. Salcedo, T. Mercado, and Y. Quiñonez, "Internet de las cosas aplicado a la agricultura: estado actual y su aplicación mediante un prototipo," *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, no. 53, pp. 106–121, 2024.

[28] B. Li, K. Zuo, L. Li, and Y. Wu, "Real-time human activity recognition on edge microcontrollers: Dynamic hierarchical inference with multi-spectral sensor fusion," *arXiv preprint arXiv:2602.00152*, 2026.

[29] J. Moosmann, M. Giordano, C. Vogt, and M. Magno, "TinyissimoYOLO: A quantized, low-memory footprint TinyML object detection network for low power microcontrollers," *IEEE Access*, vol. 11, pp. 74562–74575, 2023.

[30] A. Fanariotis, T. Orphanoudakis, and V. Fotopoulos, "Reducing the power consumption of edge devices supporting ambient intelligence applications," *Information*, vol. 15, no. 3, p. 161, 2024.

[31] A. Kumar, S. Jain, and R. Saxena, "Efficient FFT computation and optimization techniques for low-power microcontrollers in IoT applications," *IEEE Access*, vol. 11, pp. 68214–68226, 2023.

[32] M. R. Hassan, T. A. Khan, and F. A. Khan, "Fast spectral energy computation using lookup tables and bit-shifting methods for embedded systems," *Microprocessors and Microsystems*, vol. 96, p. 104721, 2024.

[33] J. A. López, P. Martínez, and F. Torres, "Memory-efficient spectral processing and FFT optimization on ARM Cortex-M microcontrollers for real-time IoT systems," IEEE Access, vol. 9, pp. 87530–87541, 2021.

[34] H. Ramírez, D. Gómez, and L. Moreno, “Algorithmic complexity reduction strategies for embedded real-time systems with constrained memory resources,” Microprocessors and Microsystems, vol. 62, pp. 27–38, 2018.

[35] S. Cortesi, C. Vogt, E. Reinschmidt, and M. Magno, “Latency and power consumption in 2.4 GHz IoT wireless mesh nodes: An experimental evaluation of Bluetooth mesh and Wirepas mesh,” IEEE Access, vol. 11, pp. 112345–112358, 2023.

Author 1: student of Electronic Engineering at the Universidad de la Costa (Colombia). Member of the Robotics and Artificial Intelligence SIRO research group attached to the GIECUC research group, his research interests are developing robotic systems and models based on embedded systems for competitive robotics. https://orcid.org/0009-0000-1545-2593

Author 2: student of Electronic Engineering at the Universidad de la Costa (Colombia). Member of the Robotics and Artificial Intelligence SIRO research group attached to the GIECUC research group, his research interests are developing robotics. https://orcid.org/0009-0000-1155-5106

Author 3: is a Colombian Electronic engineer, specialist in convergent networks and full-time professor in the Electronic Engineering Program at the Universidad de la Costa (CUC) in Barranquilla. Linked to the GIECUC research group. His academic and professional career focuses on applied research in areas such as artificial intelligence, machine learning, neural networks, computer vision, educational robotics, cardiovascular disease, and intrusion detection systems. He holds a master’s degree in systems and computing engineering with a focus on research and a PhD in Information and Communication Technologies. https://orcid.org/0000-0001-6609-1687