Real-Time Wireless Sound Localisation and Frequency Analysis: A Low-Latency IoT Approach

Hritam Sarkar[1], Tamal Sarkar[2]

[1] Department of Computer Science and Technology, University of North Bengal, P.O. NBU, Siliguri- 734013
Email: *hritamsarkar[at]gmail.com*

[2]HECRRC, University of North Bengal, P.O. NBU, Siliguri-734013
Corresponding Author Email: *tsarkar[at]nbu.ac.in*

Abstract: ***This paper presents the design and implementation of a Cost-effective, directional audio monitoring system utilizing the ESP32 platform. To address the challenge of visualizing abstract acoustic phenomena, the system integrates a four-channel orthogonal microphone array with real-time Fast Fourier Transform (FFT) signal processing. The primary novelty of this work lies in the strategic synergy of the ESP32's high-speed dual-core architecture with the ESP-NOW connectionless protocol. This combination effectively overcomes the processing speed, memory, and wireless latency limitations found in alternative low-cost microcontrollers, such as standard Arduino or STM32 boards. By performing simultaneous 512-point FFT analysis on four independent inputs, the system achieves a frequency resolution of 15.625 Hz with a full-system update cycle of approximately 300ms. Experimental validation using tuning forks confirmed the system's accuracy in identifying dominant frequencies and source direction, attributing observed deviations to the FFT algorithm's inherent discrete "binning" rather than sensor error. Additionally, the system demonstrates sub-10 ms wireless transmission latency and identifies a positive correlation between sound frequency and reliable detection distance. This work establishes a robust, low-power foundation for intelligent acoustic sensing in IoT applications without the need for expensive digital signal processors.***

Keywords: Fast Fourier Transform, ESP32, ESP-NOW, IoT Audio Monitoring, Sound Localisation

1. Introduction

At its heart, physics is our way of making sense of how the universe behaves. We start by simply paying attention to the world around us using our own five senses. However, not everything in physics is "what you see is what you get." Scientists generally group these observations into two categories:(i) Concrete Phenomena: The "Real" World, (ii) Abstract Phenomena: The "Hidden" World.

Concrete Phenomena are the things we can touch, see, or hear directly. If we drop a ball and watch it bounce, or feel the heat coming off a campfire, we are experiencing a concrete phenomenon. Because we can witness objects moving and reacting in real time, these concepts are more intuitive and easier for us to wrap our heads around. Abstract Phenomena are the things that happen just out of reach of our natural senses.Think of the way magnetic fields pull at a compass needle, or how atoms vibrate in a solid. To "see" these, we need specialized tools and instruments. Because we can't observe these directly, they're much harder to learn. They require us to flex our high-order thinking skills, using visualization and profound logic to understand a world that our eyes can't actually see [1]. The ability to decompose complex acoustic signals into their constituent parts is fundamental to modern digital signal processing. The Fourier Transform [2] serves as the mathematical bridge between the time and frequency domains. In the time domain, a signal represents how its amplitude changes over a specific period; in the frequency domain, it reveals the frequencies present in that signal. By transforming a continuous function f(t) into its frequency representation F(ω), hidden patterns in the data become visible. Nowadays, noise cancellation using FFT (Fast Fourier Transform) works by converting a time-domain signal (like audio) into its frequency components, allowing us to identify and filter out unwanted noise frequencies (e.g., by setting low-amplitude frequencies to zero) in the frequency domain, and then converting the cleaned signal back to the time domain using the Inverse FFT (IFFT) to get a clearer signal. This process isolates noise (often high-frequency random noise or specific recurring frequencies) from the desired signal, effectively reducing or removing it. This transition is also essential for various applications, including audio equalization and image compression.

There are three transfer variants [3]:

- **Discrete Fourier Transform (DFT):** Used for digital signals consisting of discrete samples rather than continuous waves.
- **Fast Fourier Transform (FFT):** A highly efficient algorithm used by computers to calculate the DFT rapidly. It is the backbone of modern digital signal processing.
- **Fourier Series:** A specialized version used only for periodic functions (signals that repeat exactly over time).

2. Literature Review

The 1990 paper by Duhamel and Vetterli [4]provides a foundational tutorial on Fast Fourier Transforms (FFTs), tracing their evolution from historical roots to advanced 1990s implementations. It reviews core algorithms such as Cooley-Tukey, Winograd, and prime-factor methods, emphasizing arithmetic complexity and reductions for DSP applications. This work remains highly cited for its constructive breakdown of FFT variants suited to general-purpose computers, DSPs, and vector processors[4]. FFTs originated with Gauss's 1805 work on periodic functions, but gained prominence with the post-1965 Cooley-Tukey radix-2 algorithm, which slashed DFT complexity from $O(n^2)$ to $O(n \log n)$. Subsequent innovations included split-radix for fewer operations and Winograd's small-DFT focus, minimizing multiplications via algebraic identities. Duhamel and Vetterli

[4] catalogue these, highlighting Cook-Toom's real-data optimizations that replace complex multiplies.

3. Methodology

1) Fast Fourier Transform:

The Fast Fourier Transform (FFT) finds frequencies by converting a time-domain signal (like sound or voltage over time) into its frequency-domain components, revealing the individual sine/cosine waves that make it up. We apply the FFT algorithm to our sample data, such as audio, and the output shows spikes (peaks) at specific frequencies; the peak's position indicates the frequency, and its height indicates the amplitude, helping identify harmonics, noise, or fundamental tones in complex signals [5].

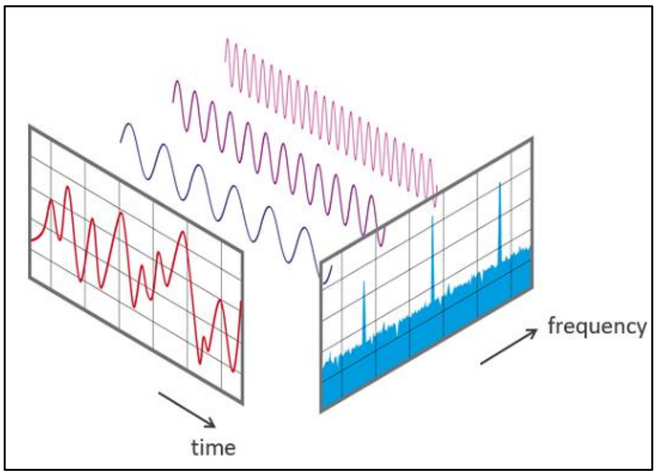


Figure 1: View of a signal in the time and frequency domains

The most common form is the Continuous Fourier Transform (CFT) [3], which transforms a continuous function f(t) into its frequency representation F(ω):

Forward Transform: $F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$ (Converts time into frequency)

Reverse Transform: $f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega$ (Reconstruct the Original Signal)

2) ESPNOW communication protocol:

A critical component of this monitoring system is the ESP-NOW protocol, a fast, low-power, connectionless Wi-Fi protocol developed by Espressif [6]. It enables direct device-to-device communication without a router or an internet connection.

ESP-NOW operates primarily at the data-link layer, effectively reducing the OSI model to a single layer. This simplification offers several benefits:

- **Reduced Latency**: By bypassing the network, transport, session, presentation, and application layers, data transmission is significantly faster.
- **Efficiency**: The protocol eliminates the need for complex packet headers or unpackers at each layer, reducing the delay caused by packet loss in congested networks.
- **Resource Management**: It consumes fewer CPU and flash resources than standard Wi-Fi protocols.
- **Security:** Utilises ECDH and AES algorithms to ensure secure data transmission.
- **Versatility:** Supports one-to-one, one-to-many, and many-to-many device configurations.
- **Payload:** Can carry up to a 250-byte payload per transmission.
- **Power Consumption**: Features a window synchronization mechanism that significantly reduces power consumption, making it ideal for battery-powered IoT sensors.

3) System Design and Implementation

The system's intelligence lies in the integration of the *ArduinoFFT* library [7]on the ESP32 platform. The process begins by capturing a fixed block of audio samples from sound sensor modules (analogue microphone) inputs and storing them in memory arrays (*vReal[]* and *vImag[]*).

Two primary parameters define the FFT analysis in this system:

- Sampling frequency (fs) = 8000Hz (8KHz), representing the average number of samples obtained in one second.
- Blocklength (BL) = 2^9=512 samples. In FFT, this value must always be an integer power of 2.

From these, several secondary parameters are derived:

- Bandwidth (also known as the Nyquist frequency) $(f_n) = fs/2 = 4000Hz$, establishing the maximum detectable frequency.
- Capture duration (D) = BL/fs = 512/8000Hz = 64ms
- Frequency resolution (Δf): The spacing between two measurement results = fs / BL = 8000Hz /512 = 15.625 Hz per FFT bin.

Digital Signal Processing Pipeline:
Each of the four microphones undergoes an identical five-stage analysis to ensure consistency in sound localization:

- **Total Energy Integration**: The system measures aggregate acoustic power across the entire spectrum to assess loudness.
- **Spectral Window Conditioning**: A Hamming taper is applied to the data to eliminate edge discontinuities and prevent spectral leakage artefacts.
- **Frequency Domain Transformation**: A 512-point FFT decomposes the time-domain signal into frequency spectra.
- **Dominant Frequency Extraction**: The system identifies the primary tonal component with 15.625 Hz precision.
- **Logarithmic Level Quantification**: RMS amplitudes are converted to relative decibels using the formula: $f(RMS) = 20.0 \times \log_{10}(RMS + 1e - 6)$

4) Components Required:

The system is composed of two distinct units: a transmitter/analyzer unit that captures and processes audio, and a receiver/display unit that visualizes the results.

a) ESP32 Development Board
b) NodeMCU ESP8266 [8]
c) Analog Sound Sensor (x4)
d) I2C 20X4 LCD display
e) 4LED and 220 Ω Resistor (x4)

4. Schematic Diagram and Experimental Setup:

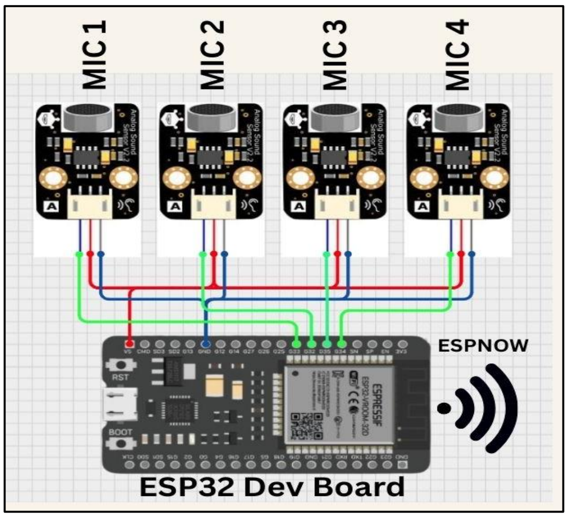


Figure 2: Frequency analyzer and transmitter

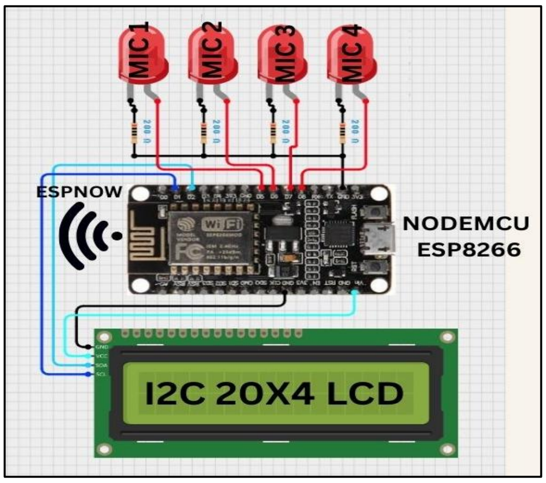


Figure 3: Receiver and remote visualization

The ESP32 dev board acts as a master for both Digital Signal Processing (after converting the analogue waveform to a digital signal) and a transmitter of calculated data, Figure 2. Figure 4 is the actual experimental setup of the ESP32 master, enclosed in a 270 mm X 225 mm X 90mm cardboard box for better reception of audio from the cardinal direction. Figures 3 and 4 show the receiver and remote visualization via 4 LEDs and a 20x4 I2C LCD display.

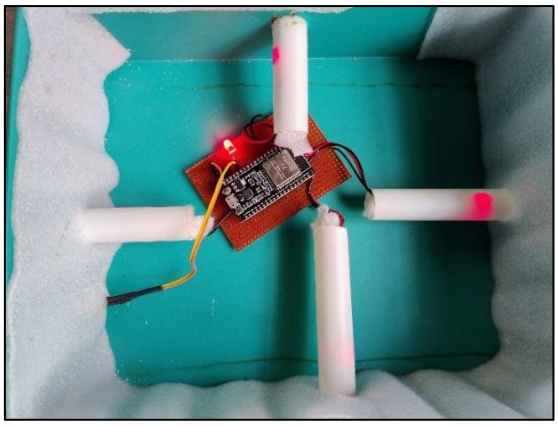
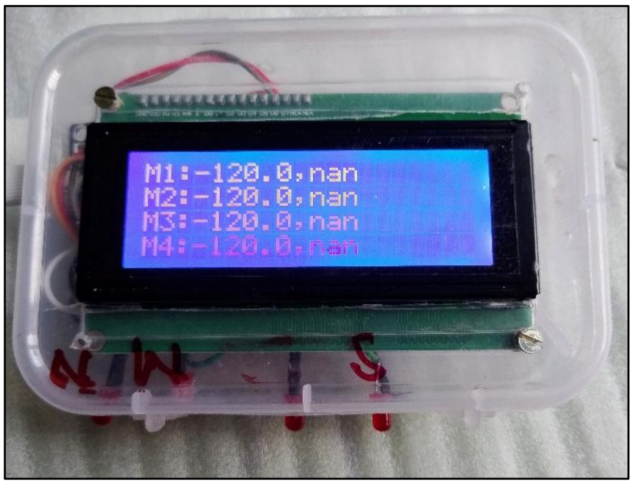


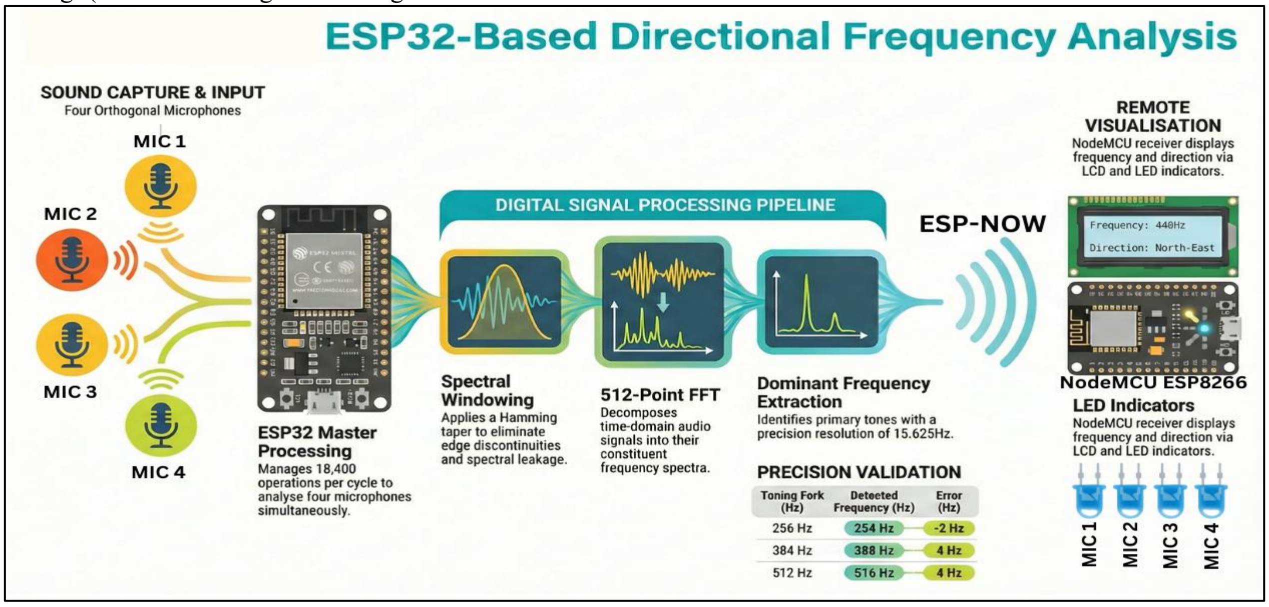
Figure 4: ESP32 master analyzer



```
M1:-120.0,nan
M2:-120.0,nan
M3:-120.0,nan
M4:-120.0,nan
```

Figure 5: Receiver and remote visualization

5. Working Principle



ESP32 Master Processing
Manages 18,400 operations per cycle to analyse four microphones simultaneously.

Spectral Windowing
Applies a Hamming taper to eliminate edge discontinuities and spectral leakage.

512-Point FFT
Decomposes time-domain audio signals into their constituent frequency spectra.

Dominant Frequency Extraction
Identifies primary tones with a precision resolution of 15.625Hz.

REMOTE VISUALISATION
NodeMCU receiver displays frequency and direction via LCD and LED indicators.

LED Indicators
NodeMCU receiver displays frequency and direction via LCD and LED indicators.

PRECISION VALIDATION

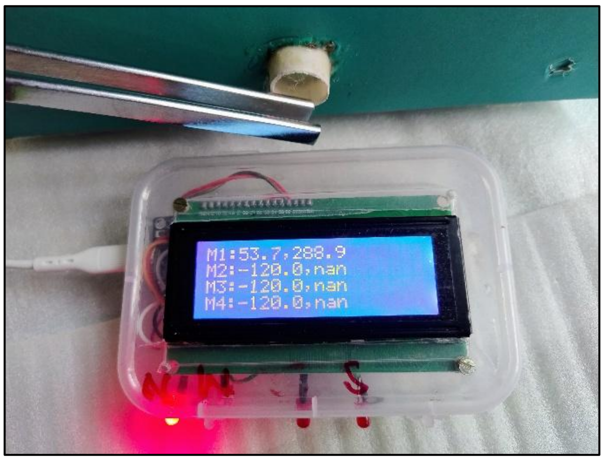
Toning Fork (Hz)	Detected Frequency (Hz)	Error (Hz)
256 Hz	254 Hz	-2 Hz
384 Hz	388 Hz	4 Hz
512 Hz	516 Hz	4 Hz

Figure 6: Working Principle

The hardware utilizes four sound sensor modules spatially arranged in an orthogonal configuration on the ESP32 master platform, shown in Figure 6. This physical layout allows the system to capture incoming sound wavefronts from all cardinal directions and measure the frequency and sound pressure level (SPL) simultaneously.

Once processed, the four-channel datasets are transmitted via ESP-NOW to the NodeMCU ESP8266[8] receiver. This transmission features sub-10ms latency and can maintain a range of up to 50 meters through typical building structures. The remote station displays the primary detected frequency and composite SPL, while four LED indicators provide immediate visual feedback regarding the direction of the sound source.

6. Experiment Results



```
M1:53.7,288.9
M2:-120.0,nan
M3:-120.0,nan
M4:-120.0,nan
```

Figure 7: Tuning Fork frequency response demonstration

Figure 8: Set of Tuning Fork

The experiment was conducted with different tuning fork frequencies (figure 8) to analyze the frequency response of each microphone sensor. Each tuning fork needs to be near the microphone because the tuning fork's sound is measured to determine its frequency (figure 7). Table 1 shows the frequency response for the different tuning forks.

Table 1: Sound Sensor Frequency Response to Different Tuning Fork

TUNING FORK	FREQUENCY DETECTION			
FREQUENCY	MIC 1	MIC 2	MIC 3	MIC 4
256 Hz	254 Hz	254 Hz	254 Hz	254 Hz
288 Hz	289 Hz	289 Hz	289 Hz	289 Hz
320 Hz	325 Hz	325 Hz	325 Hz	325 Hz
341 Hz	343 Hz	343 Hz	343 Hz	343 Hz
384 Hz	388 Hz	388 Hz	389 Hz	388 Hz
426 Hz	431 Hz	431 Hz	431 Hz	431 Hz
480 Hz	484 Hz	484 Hz	484 Hz	484 Hz
512 Hz	516 Hz	516 Hz	516 Hz	516 Hz

Amplifying the frequency significantly increases the sound sensor's detection range. For that, we used an online tone generator [9] on our mobile device, setting a fixed volume level at different frequencies. Table 2 shows a few sound frequency responses vs. the distance of the detected frequency from the sensor. The results clearly indicate a positive correlation: as the sound frequency increases, the distance at which it can be reliably detected by the sensor also increases. This finding provides valuable insight into the system's operational range for different acoustic sources.

Table 2: Frequency response vs detection of distance from the sound sensor

Frequency (Hz)	Node	Detected Frequency (Hz) at MIC 1	Frequency detection distance from the sensor (MIC 1) (cm)
440	A4	441	9
880	A5	884	50
1760	A6	1769	67
294	D4	296	5
587	D5	591	21
1174	D6	1180	56
349	F4	350	5
698	F5	703	34
1396	F6	1405	62

7. Discussion

First question: Why use an ESP32 development board? There are many development boards.

1) High-Speed Dual-Core Processing for Parallel FFT Analysis:
    a) 512-point complex FFT = ~4600 operation per microphone
    b) 4 microphone x 4600 = 18,400 operation per cycle
    c) 300msec update rate
2) 12-bit ADC for high-fidelity audio sampling,
3) ESP-NOW protocol Integration,
4) Memory Architecture for Real-Time FFT Buffers
    a) vReal[512] + vImag[512] = 1024 doubles (8KB) per mic
    b) 4 mics x 8KB = 32KB working buffers
    c) Struct payload: 49 bytes x 2(TX/RX) = 100 bytes
    d) Total RAM: ~33KB dynamics + 20KB code = 53KB
5) Cost-effective

No alternative microcontroller provides this exact combination at this price point. Arduino [10] family fails on speed/ wireless/ memory. STM32 requires an external radio. ESP32 is a perfect match for directional audio FFT monitoring.

Second, the frequency response for the Tuning Fork:

Table 3: Bin Index of Tuning Fork and Detected Frequency

Tuning Fork Frequency (Hz)	Bin Index (Expected) (freq / Δf)	Detected Frequency (Hz)	Error (Hz)	Bin Index (Calculated)
256 Hz	16.384	254	-2	16.256
288 Hz	18.432	289	1	18.496
320 Hz	20.48	325	5	20.8
341 Hz	21.824	343	2	21.952
384 Hz	24.576	388	4	24.832
426 Hz	27.264	431	5	27.584
480 Hz	30.72	484	4	30.976
512 Hz	32.768	516	4	33.024

The observed differences in Table 3 between the tuning fork frequencies and the system's measurements are not due to sensor error but are a direct consequence of the FFT's discrete nature. The *FFT.majorPeak()* function returns the centre frequency of the FFT bin containing the most signal energy; it does not interpolate to find the true peak frequency between bins. This "binning" effect is dictated by the system's frequency resolution of 15.625 Hz. The 426 Hz test case clearly illustrates this principle. The true frequency of 426 Hz falls between two adjacent FFT bins:

- Bin 27: Represents a centre frequency of 15.625 Hz × 27 = 421.875 Hz
- Bin 28: Represents a centre frequency of 15.625 Hz × 28 = 437.500 Hz

The resulting output of 431 Hz, as recorded in the experiment, indicates the algorithm determined the peak energy to be between the centres of Bin 27 and Bin 28, reporting a value close to their midpoint. This demonstrates a core performance characteristic tied directly to the fundamental parameters of the DSP pipeline.

8. Conclusion

This project successfully demonstrates the design and implementation of a directional audio monitoring system based on an ESP32. By integrating multi-channel audio capture, real-time FFT-based signal processing, and low-latency wireless transmission, the system provides a powerful and cost-effective solution for IoT audio analysis. The strategic selection of the ESP32 platform and the ESP-NOW protocol[6] proved essential to meeting the application's demanding computational and communication requirements.

The analysis of the system's performance underscores the key engineering trade-off between frequency resolution and update latency. The currently implemented parameters are optimized to provide a practical balance, delivering a 15.625 Hz resolution with a near real-time ~300 ms full-system scan cycle suitable for wireless monitoring. This work serves as a robust foundation for further development in intelligent acoustic sensing.

Acknowledgement
The authors would like to acknowledge the support of the Director of the High Energy and Cosmic Ray Research Centre for providing the necessary infrastructure for the project. T. Sarkar likes to acknowledge the support provided by the Inter-University Centre for Astronomy and Astrophysics (IUCAA), specifically under the IUCAA Associateship Programme and the National Research Centre (NRC) IUCAA.

Declarations

Author Contribution: Hritam Sarkar: Investigation, Resources, Programming, Validation, Writing-Original draft., Tamal Sarkar: Supervision, Project Administrator, Conceptualization, Validation, Investigation, Resources, Writing Review and Editing.

Code and Data Availability: The project's source code is publicly available for review and further development on GitHub: https://github.com/Hoptimus/Wireless_Frequency_Monitor.git

Conflict of Interest: The authors do not have any conflict of interest.

Ethical Approval: Not Applicable.

Consent for publication: We affirm that the manuscript is original, has not been published elsewhere, and is not currently under consideration by another journal.

References

[1] Jufrida, H. S. Falah, and N. H. Sehab, 'Development of An Audio Frequency Meter with Arduino and MAX4466 Sound Sensor', *J. Penelit. Pendidik. IPA*, vol. 9, no. 11, pp. 10280–10286, Nov. 2023, doi: 10.29303/jppipa.v9i11.6144.
[2] '3Blue1Brown - But what is the Fourier Transform? A visual introduction.' Accessed: Jan. 29, 2026. [Online]. Available: https://www.3blue1brown.com/lessons/3blue1brown.com
[3] P. R. Babu and R. Anandanatarajan, *Signals and Systems*, 4th edn. Chennai: Scitech Publications (India) Pvt. Ltd., 2017. [Online]. Available: https://www.scitechpublications.com/
[4] P. Duhamel and M. Vetterli, 'Fast fourier transforms: A tutorial review and a state of the art', *Signal Process.*, vol. 19, no. 4, pp. 259–299, Apr. 1990, doi: 10.1016/0165-1684(90)90158-U.
[5] 'FFT'. Accessed: Jan. 18, 2026. [Online]. Available: https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft
[6] 'ESP-NOW', ESP-NOW Wireless Communication Protocol. [Online]. Available: https://www.espressif.com/en/solutions/low-power-solutions/esp-now
[7] Arduino, 'arduinoFFT', arduinoFFt Document. [Online]. Available: https://docs.arduino.cc/libraries/arduinofft/#Releases
[8] H. Sarkar, R. Kumar Mandal, and T. Sarkar, 'Home-Based and Portable Smart Weather Station', *IISRR- Int. J. Res.*, vol. 10, no. IV, pp. 249–257, Dec. 2024.

[9] 'Online Tone Generator - generate pure tones of any frequency'. Accessed: Jan. 18, 2026. [Online]. Available: https://www.szynalski.com/tone-generator/

[10] H. Sarkar and T. Sarkar, 'Real-Time Monitoring of Radiant Flux and Surface Thermal Properties via an Integrated Arduino-Thermocouple Setup', *Int. J. Sci. Res. IJSR*, pp. 2269–2277, Dec. 2025, doi: 10.21275/SR251226144505.