IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 2, March 2025

Audio Spectrum Analyser using Arduino Nano for Automation Applications

Parmar Avinashkumar[1], Chakkiwala Mohamad Shahzeb[2], Adhvaryu Sujal M.[3], Ashwin Dabhi[4], Savan Kachhatiya[5], Dr. Vipul A Shah[6]
Student, Department of Instrumentation and Control Engineering[1-5]
Professor, Department of Instrumentation and Control Engineering[6]
FoT Dharmsinh Desai University, Nadiad, India

Abstract: *This paper explores the design and implementation of an audio spectrum analyser using the Arduino Nano microcontroller. While traditionally employed for visualizing audio frequencies, this project focuses on leveraging the spectral analysis capabilities for automation applications. We examine the hardware and software considerations in creating a low-cost, portable spectrum analyser, highlighting the challenges and solutions involved in processing audio signals with limited computational resources. Furthermore, we delve into potential use cases within the automation sector, discussing how frequency analysis can be integrated into systems for predictive maintenance, noise monitoring, process control, and environmental monitoring. The paper concludes by outlining future research directions and potential improvements to enhance the accuracy and applicability of the Arduino-based spectrum analyser in demanding automation scenarios*

Keywords: Arduino Nano, Audio Spectrum Analyser, FFT, Automation, Predictive Maintenance, Noise Monitoring, Frequency Analysis, Signal Processing

I. INTRODUCTION

The ability to analyse the frequency content of audio signals is crucial in various fields, from music production and sound engineering to industrial monitoring and process control. Traditionally, dedicated spectrum analysers offer sophisticated features and high precision, but their cost and complexity can be prohibitive for many applications. The Arduino Nano provides a low-cost, accessible platform for creating a basic yet functional spectrum analyser, making frequency analysis more accessible.

This paper investigates the implementation of an audio spectrum analyser using the Arduino Nano, specifically focusing on its potential applications within the automation sector. In automation, understanding sound patterns and frequency signatures can provide valuable insights into the health and performance of machinery, environmental conditions, and process dynamics. By analysing the frequency content of audio signals, we can detect anomalies, predict failures, monitor noise levels, and even control certain processes based on specific acoustic characteristics.

The challenge lies in optimizing the signal processing techniques for the limited processing power and memory of the Arduino Nano. This paper details the hardware and software implementation, discussing the trade-offs between accuracy, resolution, and processing speed. We also explore concrete examples of how this technology can be applied to solve real-world problems in automation.

II. BACKGROUND AND RELATED WORK

2.1 FUNDAMENTALS OF SPECTRUM ANALYSIS:

Spectrum analysis involves decomposing a complex signal into its constituent frequencies and their corresponding amplitudes. This is typically achieved using the Fourier Transform, which mathematically transforms a signal from the time domain to the frequency domain. The resulting spectrum displays the amplitude of each frequency component, providing a visual representation of the signal's frequency content.[3]

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 2, March 2025

2.2 THE DISCRETE FOURIER TRANSFORM (DFT) AND FAST FOURIER TRANSFORM (FFT):

For digital signal processing, the Discrete Fourier Transform (DFT) is used instead of the continuous Fourier Transform. The DFT requires significant computational resources. The Fast Fourier Transform (FFT) is an efficient algorithm for calculating the DFT, significantly reducing the computational complexity. Therefore, the FFT is the algorithm of choice for real-time spectrum analysis on microcontrollers like the Arduino Nano.

2.3 ARDUINO AND AUDIO PROCESSING:

The Arduino platform provides a readily available and cost-effective solution for prototyping and developing various electronic systems, including audio processing applications. The Arduino Nano, specifically, is a small and versatile microcontroller ideal for embedded applications where size and power consumption are critical. Several projects have explored using Arduino for basic audio processing tasks, such as audio recording, playback, and simple filtering.[1]

2.4 PRIOR ART:

Previous work on Arduino-based spectrum analysers typically focuses on visualization, often using LED matrices or graphical displays to represent the frequency spectrum. These projects often implement simplified versions of the FFT algorithm or utilize pre-existing libraries designed for audio analysis.[2] However, few studies explore the application of such a system in an automation context, where the extracted frequency data is used for control actions or predictive analytics.

2.5 SCHEMATIC DIAGRAM

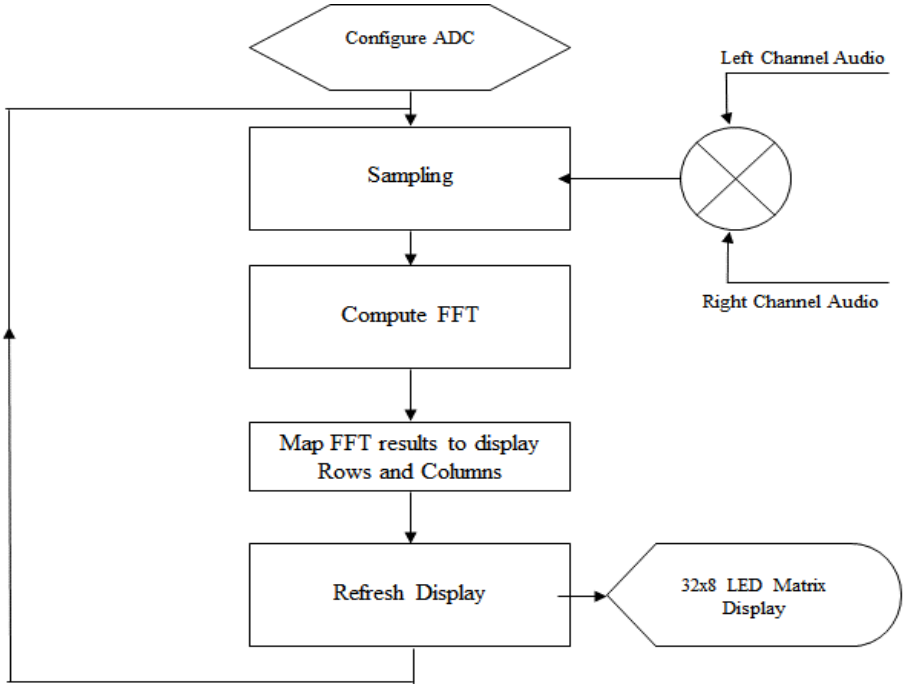

Fig. 1 A schematic Block Diagram of the Working of Spectrum Analyser

III. HARDWARE IMPLEMENTATION

The hardware components required for this project are:

- **Arduino Nano:** The microcontroller responsible for data acquisition, processing, and control.
- **Microphone Amplifier (e.g., MAX9814):** To amplify the weak audio signal from the microphone to a level suitable for the Arduino's ADC (Analog-to-Digital Converter).

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 2, March 2025

- **Microphone:** Captures the ambient sound.
- **Resistors and Capacitors:** For biasing and filtering the audio input signal.
- **LCD/OLED Display:** For visual feedback of the frequency spectrum.
- **Connectivity Module (e.g., ESP8266/ESP32):** For remote monitoring and data logging.

Hardware Setup and Connection:

- Connect the microphone to the Electret Microphone Amplifier.
- Connect the output of the amplifier to an analog input pin on the Arduino Nano (e.g., A0).
- Connect the required power and ground connections for the amplifier and other peripherals.
- Connect the LCD/OLED display to the appropriate digital pins on the Arduino.
- Connect the connectivity module to the appropriate serial communication pins on the Arduino.

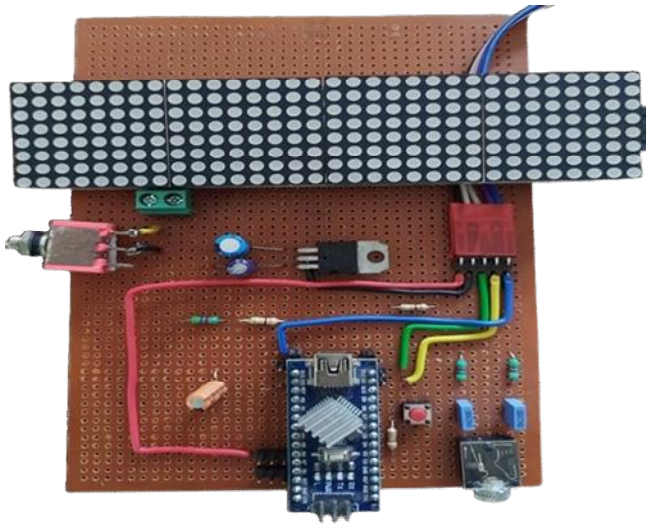

Fig. 2 Hardware Setup of the Spectrum Analyser

IV. SOFTWARE IMPLEMENTATION

4.1 DATA ACQUISITION

The Arduino reads the analog input from the microphone at a specific sampling rate. The sampling rate is a crucial parameter that determines the maximum frequency that can be accurately represented (Nyquist-Shannon sampling theorem).[1] A higher sampling rate allows for the detection of higher frequencies, but it also increases the processing load on the Arduino. A typical sampling rate for audio analysis is 8kHz or 16kHz.

4.2 FFT ALGORITHM IMPLEMENTATION

The core of the software is the FFT algorithm. Due to the limited memory and processing power of the Arduino Nano, a pre-optimized FFT library, such as the "Arduino FFT" library, is recommended. This library provides a relatively efficient implementation of the Cooley-Tukey FFT algorithm, suitable for real-time audio processing.[2]

4.3 WINDOWING FUNCTION

Before applying the FFT, a windowing function is applied to the sampled data. Windowing reduces spectral leakage, which occurs when the signal is not exactly periodic within the sampling window. Common windowing functions include Hamming, Hanning, and Blackman windows. The choice of windowing function depends on the specific application and the trade-off between spectral resolution and leakage reduction.[4]

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 2, March 2025

4.4 FREQUENCY BIN CALCULATION

The FFT algorithm outputs a complex-valued array representing the frequency spectrum. The magnitude of each frequency bin corresponds to the amplitude of that frequency.[5] The frequency corresponding to each bin can be calculated as:

frequency = (bin_index * sampling rate) / number_of_samples

4.5 DATA PROCESSING AND DISPLAY

The calculated frequency bins are then processed and displayed. For simple visualization, the magnitude of each bin can be mapped to the brightness of an LED or the height of a bar on an LCD/OLED display. For more complex applications in automation, the data can be filtered, thresholded, or used as input to a control algorithm.[7]

V. APPLICATIONS IN AUTOMATION

The Arduino-based audio spectrum analyser has a wide range of potential applications in the automation sector. Here are some examples:

5.1 PREDICTIVE MAINTENANCE:

- **Bearing Fault Detection:** Analyzing the sound emitted by rotating machinery can reveal early signs of bearing failure. A healthy bearing typically produces a relatively quiet and consistent sound signature. As a bearing deteriorates, it begins to generate specific frequencies related to the type and severity of the fault. Analyzing the frequency spectrum can detect these anomalies before catastrophic failure occurs.[4]
- **Pump Cavitation Detection:** Cavitation in pumps is a common problem that can lead to reduced performance and damage. Cavitation creates a distinct, high-frequency noise that can be easily identified in the audio spectrum. Monitoring the amplitude of these frequencies can provide early warning of cavitation onset.
- **Gearbox Monitoring:** Similar to bearing fault detection, analysing the sound of a gearbox can reveal issues such as gear wear, misalignment, or lubrication problems. Each gear meshing frequency appears at a specific fundamental frequency and its harmonics. Deviations from the normal frequency pattern indicate problems.[6]

5.2 NOISE MONITORING AND CONTROL:

- **Environmental Noise Monitoring:** Monitoring noise levels in industrial environments is essential for worker safety and compliance with regulations. The Arduino-based spectrum analyser can be used to identify and quantify the specific frequencies contributing to the overall noise level, allowing for targeted noise reduction measures.
- **Machine Noise Profiling:** By analysing the noise signature of different machines, it's possible to create a baseline profile. Any deviations from this baseline due to wear, malfunction, or improper operation could be flagged as potential issues.[8]
- **Active Noise Cancellation:** In specific applications, the spectrum analyser can be used as part of an active noise cancellation system. By analysing the incoming noise, the Arduino can generate an anti-noise signal that cancels out the unwanted frequencies. This is more complex and would likely require a faster processor for real-time analysis and signal generation.[5]

5.3 PROCESS CONTROL:

- **Acoustic Emission Monitoring:** Acoustic emission (AE) is the transient elastic waves generated by the rapid release of energy within a material. Monitoring AE can provide information about material properties, crack growth, and other processes. The Arduino-based spectrum analyser can be used to analyse the frequency content of AE signals, allowing for real-time monitoring of material behaviour.[5]
- **Combustion Monitoring:** The sound of combustion processes in engines or furnaces contains valuable information about the efficiency and stability of the combustion. Analyzing the frequencies emitted during combustion can help optimize combustion parameters for improved efficiency and reduced emissions.[7]

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 2, March 2025

- **Leak Detection:** Leaks of gas or fluids often generate a high-frequency whistling sound that can be detected by the spectrum analyser. This can be used for leak detection in pipelines, tanks, and other industrial equipment.[4]

5.4 ENVIRONMENTAL MONITORING:

- **Wildlife Monitoring:** Analyzing the soundscape of a specific area can help monitor wildlife activity. Identifying and classifying different animal vocalizations based on their frequency signatures.
- **Pollution Monitoring:** Certain types of pollution, such as noise pollution from traffic or industrial activity, can be monitored using an audio spectrum analyser.

VI. CHALLENGES AND LIMITATIONS

Implementing an audio spectrum analyser on the Arduino Nano presents several challenges:

- **Limited Processing Power:** The Arduino Nano has limited processing power, which restricts the complexity of the FFT algorithm that can be implemented in real-time.
- **Limited Memory:** The limited memory of the Arduino Nano restricts the number of samples that can be processed in each FFT analysis, which impacts the frequency resolution.
- **Analog-to-Digital Converter (ADC) Resolution:** The ADC resolution of the Arduino Nano is limited, which affects the accuracy of the amplitude measurements.
- **Noise and Interference:** The Arduino Nano can be susceptible to noise and interference, which can affect the accuracy of the frequency analysis.
- **Real-time Performance:** Achieving true real-time performance with complex FFT algorithms can be challenging due to the processing limitations.[3]

VII. SOLUTIONS AND OPTIMIZATIONS

To address the challenges and limitations mentioned above, several optimization techniques can be employed:

- **Optimized FFT Libraries:** Utilizing pre-optimized FFT libraries like arduinoFFT or fix_fft significantly improves performance.
- **Reduced Sampling Rate and Sample Size:** Adjusting the sampling rate and sample size to the minimum required for the specific application can reduce the processing load. Carefully consider the Nyquist rate for frequencies of interest.
- **Data Filtering:** Applying digital filters before the FFT can reduce noise and interference.
- **Hardware Acceleration:** For more demanding applications, consider using a dedicated FFT accelerator chip or a more powerful microcontroller (e.g., Arduino Due, ESP32) to offload the processing burden.
- **Pre-processing Techniques:** Implementing techniques like DC removal and normalization can improve the accuracy of the FFT results.
- **Averaging:** Averaging multiple FFT results can reduce noise and improve the stability of the spectrum.

VIII. FUTURE DIRECTIONS

Further research and development can enhance the capabilities of the Arduino-based audio spectrum analyser and expand its applications in automation:

- **Integration with Machine Learning:** Applying machine learning algorithms to the frequency spectrum data can enable more sophisticated fault diagnosis and predictive maintenance capabilities. For example, training a classifier on different machine sounds to identify specific fault conditions.
- **Wireless Connectivity and Cloud Integration:** Integrating the Arduino-based spectrum analyser with wireless connectivity (e.g., Wi-Fi, Bluetooth) and cloud platforms allows for remote monitoring, data logging, and analysis.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 2, March 2025

- **Development of Application-Specific Algorithms:** Developing specialized algorithms tailored to specific automation applications (e.g., bearing fault detection, leak detection) can improve the accuracy and reliability of the analysis.
- **Improved Hardware Platform:** Migrating to a more powerful microcontroller can improve the performance and accuracy of the spectrum analyser. Consider ARM-based microcontrollers like the STM32 series, which offer significantly more processing power and memory.
- **Calibration and Compensation:** Implementing calibration routines and compensation techniques can improve the accuracy and reliability of the spectrum analyser.
- **Development of User-Friendly Interfaces:** Creating user-friendly interfaces (e.g., web-based dashboards, mobile apps) can make it easier for users to interact with the spectrum analyser and interpret the results.

IX. CONCLUSION

This paper has demonstrated the feasibility of creating a low-cost, portable audio spectrum analyser using the Arduino Nano for potential use in automation applications. While the Arduino Nano has limitations in terms of processing power and memory, it can be optimized to perform basic frequency analysis for tasks like predictive maintenance, noise monitoring, and process control. By carefully selecting the hardware components, optimizing the software implementation, and exploring the potential applications, the Arduino-based spectrum analyser provides a valuable tool for enhancing automation systems. Further research and development, especially in machine learning integration and cloud connectivity, will continue to expand the role of such systems in the automation sector. The key lies in understanding the specific application requirements and carefully tailoring the hardware and software implementation to meet those needs, balancing cost, performance, and accuracy.

REFERENCES

[1]. Monk, S. (2016). Programming Arduino: Getting started with sketches (2nd ed.). McGraw-Hill Education.https://datasciencewiki.net/fast-fourier-transform/
[2]. Rao, K. R., Kim, D. N., & Hwang, J.-J. (2010). Fast Fourier transform - Algorithms and applications (1st ed.). Springer.
[3]. Diverse Daily. (n.d.). Can the fast Fourier transform be computed in O(n log n) time? Diverse Daily. Retrieved from https://diversedaily.com/can-the-fast-fourier-transform-be-computed-in-on-log-n-time/
[4]. Academia. (n.d.). A monolithic audio spectrum analyser. Academia. Retrieved from https://www.academia.edu/5744234/A_monolithic_audio_spectrum_analyser
[5]. Amazing Algorithms. (n.d.). Fast Fourier transform. Amazing Algorithms. Retrieved from https://amazingalgorithms.com/definitions/fast-fourier-transform/
[6]. Electronics Lab. (n.d.). The Fourier Analysis: The Fast Fourier Transform (FFT) Method. Retrieved from https://www.electronics-lab.com/article/the-fourier-analysis-the-fast-fourier-transform-fft-method/
[7]. Bracewell, R. N. (2000). The Fourier Transform and Its Applications (3rd ed.). McGraw-Hill Education.
[8]. Burrus, C. S. (Ed.). (1985). Fast Fourier Transforms. Prentice Hall