International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Design of an Indian Urban Noise Adaptive Acoustic Vehicle Alerting System Using ESP32-S3 and PC-Based Speed Simulation

Pravin Haridasrao Gulhane[1], Prof. Swati Deshmukh[2], Dr. Ashwini Bade[3], Dr. Prabhat Pallav[4]

1 PG Student, Department of Electronics and Telecommunication Engineering
2 PG Coordinator, Department of Electronics and Telecommunication Engineering
3 Assistant Professor, Department of Electronics and Telecommunication Engineering
4 Head of Department, Department of Electronics and Telecommunication Engineering
Siddhant College of Engineering, Sudhambare, Talegaon - Chakan Highway, Sudumbre, Maharashtra 410501, India

---

Abstract - Low-speed electric vehicles can be difficult to detect acoustically, especially in Indian urban traffic where horns, engines, roadside activity, and construction create rapidly varying spectral masking. This paper presents a low-cost adaptive Acoustic Vehicle Alerting System prototype in which an ESP32-S3 analyses ambient sound and selects a bounded warning profile according to the dominant noise bands. An INMP441 digital microphone supplies 16 kHz audio for windowed 1024-point fast Fourier transform processing, while a MAX98357A amplifier drives the warning loudspeaker. For laboratory development, vehicle speed, direction, and enable state are generated by a Python web application and transmitted through USB serial instead of a vehicle Controller Area Network. The proposed partition keeps all time-critical sensing, adaptation, and audio generation on the embedded controller while using the computer only as a repeatable test instrument. Feasibility analysis shows that the selected interfaces and processing sequence fit the ESP32-S3 platform and support controlled evaluation of latency, switching stability, relative sound level, and pedestrian detectability. The architecture is intended for academic prototyping and is not a regulatory compliance claim.

Key Words: acoustic vehicle alerting system, electric vehicle, ESP32-S3, fast Fourier transform, adaptive audio, USB serial, urban noise.

1. INTRODUCTION

Electric vehicles (EVs) remove much of the propulsion noise that pedestrians traditionally use to judge vehicle approach at low speed. International regulations therefore require an Acoustic Vehicle Alerting System (AVAS) or equivalent minimum sound behaviour for applicable quiet road transport vehicles [1], [2]. A fixed warning sound is straightforward to validate, but its perceptibility depends on the surrounding spectrum. In a quiet area it may be intrusive, whereas in dense traffic it can be masked by engines, horns, speech, and construction activity.

The design challenge is particularly relevant to Indian cities because the background changes over short distances and time intervals. An adaptive AVAS should not behave as an unrestricted volume controller. It should choose among pre-qualified profiles, impose output limits, and use temporal stability rules so that rapid acoustic fluctuations do not produce distracting sound changes. Prior field work has also indicated that background level influences detection performance and motivates adaptive functionality [3].

This paper proposes a college-level prototype based on an ESP32-S3. The contribution is a practical separation of functions: a PC-based Python web application simulates vehicle speed through USB serial, while the embedded system performs ambient-noise acquisition, frequency-domain analysis, profile selection, and warning-sound generation. This makes experiments repeatable without a vehicle CAN bus and preserves a clear migration path to later automotive integration.

2. RELATED WORK AND DESIGN MOTIVATION

UN Regulation No. 138 defines provisions for quiet road transport vehicles, while FMVSS No. 141 establishes minimum sound requirements and associated test procedures in the United States [1], [2]. These standards provide the safety context, but a student prototype must not claim compliance without calibrated acoustic measurements, prescribed test conditions, and production-grade hardware.

Berge and Haukland evaluated adaptive AVAS behaviour under different background-noise conditions and reported that increased warning level can be beneficial when the background is high [3]. Related work on adaptive auditory alerts similarly tracks the acoustic environment before presenting a warning, balancing audibility and annoyance [4]. These findings support a bounded adaptation strategy based on spectral masking rather than an unconstrained increase in amplitude.

The ESP32-S3 provides two I2S peripherals that can be configured for digital audio input and output [5]. Espressif also supplies an optimized DSP library containing FFT, window, filtering, and vector functions for ESP32-class devices [6]. A 1024-sample analysis frame at 16 kHz covers 64 ms and gives a frequency-bin spacing of 15.625 Hz, which is adequate for coarse low-, mid-, and high-band energy estimates.

3. SYSTEM ARCHITECTURE

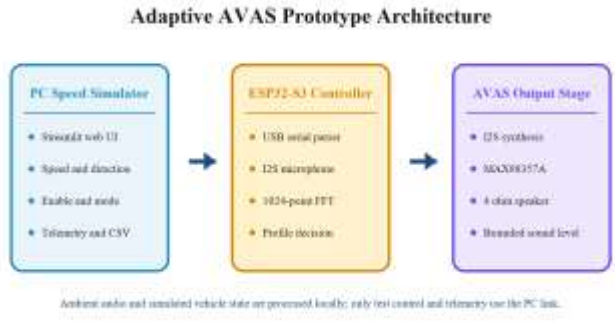


Fig -1: Proposed adaptive AVAS prototype architecture

The architecture in Fig. 1 contains three principal subsystems. The Python application supplies deterministic vehicle-state commands and displays telemetry. USB serial carries newline-delimited messages with speed, direction, AVAS enable state, and mode. The ESP32-S3 acquires microphone samples, calculates spectral features, selects a profile, and synthesizes audio. A digital Class-D amplifier and loudspeaker form the output stage.

3.1 PC-Based Vehicle-State Simulator

The simulator exposes a speed slider, direction selector, enable control, manual or automatic profile mode, and predefined ramps. Each command contains a sequence number and timestamp. A heartbeat permits the firmware to detect a disconnected host and return to a safe muted or default state. Telemetry packets report measured band energies, selected profile, output gain, frame time, and serial status for live plotting and CSV storage.

3.2 Embedded Acoustic Controller

An INMP441 I2S microphone provides digital audio, avoiding an external analogue preamplifier and ADC. Double buffering separates acquisition from processing. The controller applies DC removal, a Hann window, and a real FFT. Band energies are then smoothed with an exponential filter. The adaptive decision executes at a slower rate than sample acquisition, which reduces audible profile chatter and computational load.

3.3 Audio Output Stage

The warning waveform is generated numerically and transmitted over I2S to a MAX98357A mono Class-D amplifier [7]. For the bench prototype, a 4 ohm, 3 W loudspeaker is sufficient to demonstrate profile changes. This stage is not weatherproof or automotive qualified; output level is therefore treated as a relative laboratory variable.

4. ADAPTIVE SIGNAL PROCESSING AND CONTROL

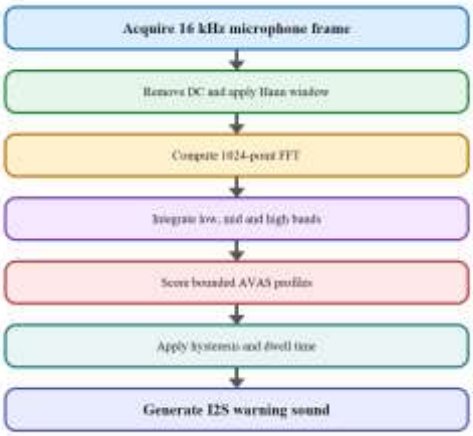


Fig -2: Ambient-noise analysis and profile-selection flow

For each frame x[n], the mean is removed and a Hann window w[n] is applied. The discrete spectrum X[k] is obtained with a 1024-point FFT. The energy in band b is calculated as the sum of squared magnitudes over the frequency-bin set Kb. Three bands are sufficient for a transparent prototype: low-frequency traffic rumble, mid-frequency speech and engine components, and higher-frequency horns or mechanical activity.

A smoothed estimate Eb[t] = alpha Eb[t-1] + (1-alpha) Eb,raw[t] is maintained for each band. The profile score combines the inverse overlap between an AVAS profile and the measured masking spectrum with a small penalty for excessive output level. Only a fixed library of candidate profiles is used. A new profile must exceed the current score by a margin and remain preferred for a dwell interval before switching.

Vehicle state constrains the decision. The prototype enables the warning only in the configured low-speed region and applies separate behaviour for forward and reverse motion. Loss of valid serial commands activates a timeout. These rules ensure that the PC does not become part of the real-time audio loop and that unstable network timing cannot directly modulate the emitted sound.

5. PROTOTYPE IMPLEMENTATION

Table -1: Prototype hardware and software

Block	Selected Item	Purpose
Controller	ESP32-S3 development board	I2S, FFT, control, telemetry
Microphone	INMP441	Digital ambient-noise capture
Amplifier	MAX98357A	I2S to Class-D speaker drive
Speaker	4 ohm, 3 W	Bench warning-sound output
Host	Laptop / PC	Speed simulation and logging
Software	ESP-IDF + ESP-DSP; Python + Streamlit	Firmware and web dashboard

The firmware is divided into acquisition, analysis, decision, synthesis, and communication tasks. Fixed-size queues pass frame descriptors and telemetry without dynamic allocation in the time-critical path. Parameters

such as band limits, smoothing coefficient, hysteresis margin, dwell time, and profile gain are stored in a configuration structure so that experiments can be reproduced.

The Python application scans available serial ports, validates the selected device, and transmits commands at a moderate fixed rate. User input is bounded before transmission. Incoming telemetry is decoded independently of the display refresh rate and appended to a CSV file containing host time, device time, speed, direction, energy features, profile, gain, and diagnostic flags.

6. EVALUATION METHODOLOGY AND FEASIBILITY

Evaluation is designed as a staged bench study rather than a compliance test. Recorded urban scenes representing quiet residential roads, moderate mixed traffic, horn-dominant congestion, and construction-dominant noise are replayed near the microphone. The PC executes repeated speed profiles while the controller log records analysis and selection behaviour. A fixed microphone-speaker geometry is maintained for comparisons.

Table -2: Proposed evaluation metrics and design targets

Metric	Method	Design Target
Frame processing	Firmware timestamp	< 64 ms frame period
Command latency	Host/device sequence timing	< 100 ms
Profile stability	Switch count in steady scene	No rapid oscillation
Masking response	Band-energy/profile trace	Expected profile selected
Audio continuity	Recorded output inspection	No clicks or dropouts
Detectability	Controlled listener trial	Improved over fixed baseline

The numerical values in Table 2 are prototype design targets, not measured results. A 1024-sample frame at 16 kHz provides a 64 ms acquisition interval. The ESP-DSP library includes optimized FFT implementations and examples using 1024 samples [6], supporting platform feasibility. Memory use is modest: even a floating-point complex buffer for 1024 samples occupies approximately 8 kB, with additional buffers and tables remaining well within typical ESP32-S3 development-board capacity.

Subjective testing should use informed participants, consistent playback level, randomized scene order, and a simple response such as detection time or rating. Any evaluation involving road users must remain in a controlled environment. Calibrated sound-pressure instrumentation is required before interpreting results against regulatory limits.

7. DISCUSSION

The proposed partition offers three advantages for an academic prototype. First, speed scenarios are repeatable and do not require access to a vehicle or CAN interface. Second, the embedded controller still contains the functions that matter for real-time behaviour, so USB does not conceal processing limitations. Third, the telemetry stream makes the adaptive decision explainable: each profile change can be traced to band energies, thresholds, and state constraints.

The principal limitations are equally important. A single microphone does not represent the complete spatial sound field around a moving vehicle. Playback recordings are approximations of real traffic. The small loudspeaker has a restricted and enclosure-dependent frequency response. Relative bench measurements cannot establish regulatory compliance, pedestrian safety benefit, or production reliability. The work should therefore be described as a feasibility and methods prototype.

Future development can replace USB serial speed with CAN or CAN-FD, introduce a weather-resistant exterior transducer, add microphone calibration, collect multi-location Indian traffic data, and compare multiple profile-scoring rules. Formal validation would require the exact test conditions, instrumentation, and approval procedures of the target market.

8. CONCLUSION

An Indian urban noise adaptive AVAS is feasible as a college prototype using an ESP32-S3, an I2S microphone, a digital audio amplifier, and a PC-based speed simulator. The proposed design performs real-time spectral analysis and bounded profile selection locally on the embedded controller, while USB serial provides a simple and repeatable substitute for vehicle CAN data during development. The architecture supports measurable experiments in latency, stability, spectral response, and relative detectability without overstating regulatory readiness. Its main engineering value is the disciplined combination of embedded DSP, controlled adaptation, and traceable test instrumentation.

ACKNOWLEDGEMENT

The author acknowledges the guidance and laboratory support provided by the Department of Electronics and Telecommunication Engineering, Siddhant College of Engineering, Pune.

REFERENCES

[1] United Nations Economic Commission for Europe, “UN Regulation No. 138: Quiet Road Transport Vehicles,” WP.29 Vehicle Regulations, Rev. 1 and subsequent amendments. Available: https://unece.org/transport/vehicle-regulations-wp29/standards/addenda-1958-agreement-regulations-121-140

[2] National Highway Traffic Safety Administration, “TP-141-01 Minimum Sound Requirements for Hybrid and Electric Vehicles,” U.S. Department of Transportation, 2020. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/tp-141-01_minimum_sound_requirements_for_hybrid_and_electric_vehicles.pdf

[3] T. S. Berge and F. Haukland, “Adaptive acoustic vehicle alerting sound, AVAS, for electric vehicles: Results from field testing,” SINTEF Digital, Report 2019:00062, 2019.

[4] E. Sabic, D. Henning, and J. MacDonald, "Adaptive Auditory Alerts for Smart In-Vehicle Interfaces," Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 63, no. 1, 2019, doi: 10.1177/1071181319631404.

[5] Espressif Systems, "Inter-IC Sound (I2S) - ESP32-S3," ESP-IDF Programming Guide. Available: https://docs.espressif.com/projects/esp-idf/en/v5.0.2/esp32s3/api-reference/peripherals/i2s.html

[6] Espressif Systems, "ESP-DSP Library and Examples," documentation. Available: https://docs.espressif.com/projects/esp-dsp/en/latest/esp32/index.html

[7] Analog Devices, "MAX98357A PCM Input Class D Audio Power Amplifier," product documentation. Available: https://www.analog.com/en/products/max98357a.html