CATCHING Z'S FINAL REPORT

By

Srikar Palani

Prineet Parhar

Suprathik Vinayakula

Final Report for ECE 445, Senior Design, Spring 2026

TA: Zhuchen Shao

6 May 2026

Project No. 88

Abstract

Catching Z's is an adaptive sleep-masking device that detects disruptive room sounds and responds with targeted masking noise. The final prototype uses an ESP32-S3, dual microphones, FFT-based frequency analysis, I2S speaker output, a custom PCB, and a 3D-printed enclosure with rotary encoder volume control. The system detects sound events using an adaptive peak-to-peak threshold, reduces false triggering with a reference microphone, and shapes white, pink, and brown noise through six frequency bands. Verification showed that the device sampled audio at 16 kHz, responded in under 50 ms, produced adjustable masking output within 40–70 dB at 0.5 m, and operated continuously for 8.254 hours without rebooting or leaking memory. Simulation results also showed a 19.5 dB reduction in an 820 Hz tonal disturbance, supporting the effectiveness of the adaptive masking algorithm.

Contents

1. Introduction

1.1 Problem and Purpose

Sleep fragmentation from sirens, traffic, barking dogs, HVAC cycling, and other short environmental noises can interrupt light sleep and reduce sleep quality. A fixed white-noise machine attempts to mask these sounds by running continuously, but it does not adapt to the acoustic environment and can become a source of unnecessary noise. The purpose of Catching Z's is to provide local, event-driven acoustic masking: the device listens for disturbances, generates masking sound only when the measured environment warrants it, and returns to silence after the disturbance has passed.

1.2 Final Functionality

The final prototype implements a continuously running embedded audio pipeline. A primary microphone is sampled through the ESP32-S3 analog-to-digital converter (ADC), while a second reference channel and an internal echo-reference buffer reduce the chance that the speaker output retriggers the detector. The analysis task estimates the current sound level, updates a baseline, computes spectral energy, and publishes a target gain and band profile. The audio task shapes white, pink, and brown noise sources through six band-pass filters and streams mono pulse-code modulation (PCM) samples to the amplifier over I2S. The physical prototype uses a custom PCB mounted inside a printed enclosure with front-facing speakers, side encoder for volume control, internal microphone and external microphone, foam microphone isolation, and a lid seal. A PEC11R rotary encoder provides 20 volume steps and press-to-mute control.

1.3 High Level Requirements

Table 1: High-level requirements

Detection Latency: The system should detect a sound event when the smoothed microphone peak-to-peak amplitude exceeds an adaptive threshold based on the moving average ambient baseline. The detection threshold is the larger of two values, either 80 ADC counts, or 1.22 times the moving average baseline plus 28 ADC counts.
Output Sound Pressure Level: The audio subsystem must be capable of producing a masking noise over an adjustable volume level of 40 dB to 75 dB measured at 0.5 meters (standard bedside distance).
Continuous Operation Stability: The system must operate continuously for a minimum duration of 8 hours (a standard sleep cycle) without any software or hardware crashes.

2 Design

The final Catching Z's system consists of four main subsystems: acoustic sensing, processing, audio output, and user interface. The acoustic sensing subsystem uses electret microphones to measure room sound and speaker feedback. The processing subsystem runs on an ESP32, which samples the microphones, detects sound events, performs frequency analysis, and controls the output noise. The audio output subsystem uses an I2S amplifier and speakers to play masking noise. The user interface uses a rotary encoder for volume control and LEDs for system status.

2.1 Block Diagram

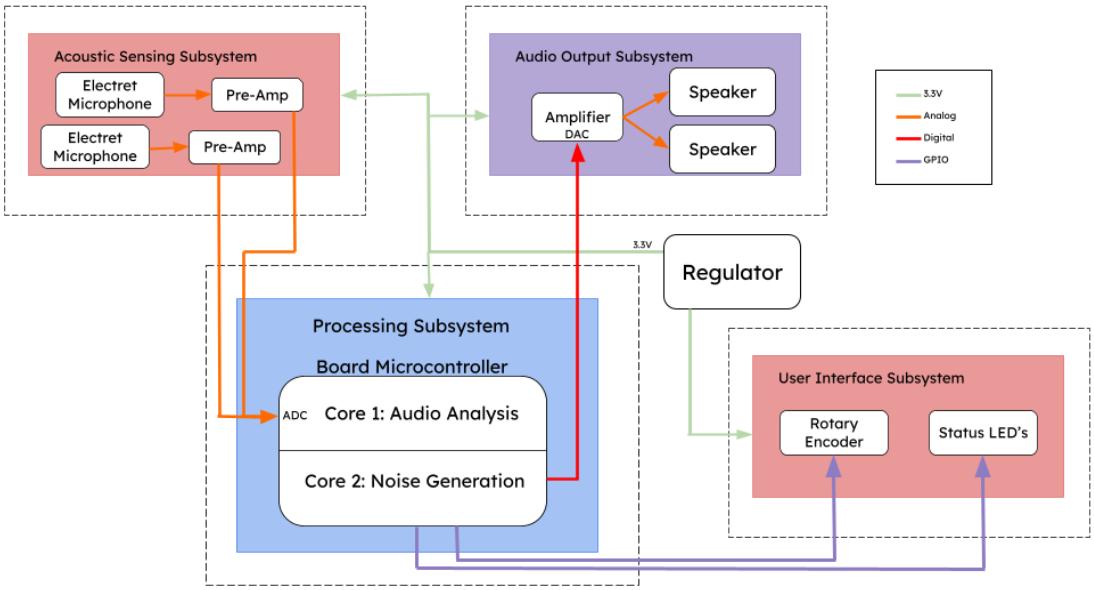

Figure 1: Block Diagram of the Catching Z's Subsystems interfacing with each other

The main design requirements were sound event detection using an adaptive threshold, adjustable masking output around bedside distance, and continuous operation for a full sleep cycle. The detection threshold was defined as:

$$Threshold = \max(80,\ 1.22 \cdot Baseline + 28)$$

A sound event is detected when the microphone's peak-to-peak amplitude exceeds this threshold. This adaptive approach was chosen instead of a fixed threshold so the device could work in both quiet and louder rooms.

2.2 Design Alternatives

Several design choices changed during development. First, we considered using a simple fixed sound threshold, but this was not reliable because the ambient noise level changes between rooms. The final design uses a moving average baseline so the threshold automatically adjusts to the environment.

Second, the first prototype generated general white noise whenever a spike was detected. This was simple, but it did not target the disturbance frequency. The final design uses FFT-based spectral analysis to identify the dominant frequency range and generate more appropriate masking noise.

Third, the enclosure design changed from a rectangular box to a pentagon shape. The rectangular enclosure helped with early testing, but the speaker output was too directional and feedback between the speaker and microphone was still a problem. The pentagon enclosure allowed two speakers to face different directions while keeping distance from the microphone. Foam was also added to reduce vibration and acoustic feedback.

Finally, the feedback solution changed from only software filtering to a dual-microphone approach. A second microphone near the speakers estimates the device's own output, which is subtracted from the main microphone reading:

$$cleaned = mic - gain \cdot refMic$$

This helped the system respond to room noise instead of retriggering from its own masking sound.

2.3 Acoustic Sensing and Detection

The sensing subsystem uses a main electret microphone to listen to the room and a reference microphone to monitor speaker output. The ESP32 samples the microphone signal at 16 kHz using 256-sample frames. For each frame, the firmware calculates the peak-to-peak amplitude:

$$P2P = x_{max} - x_{min}$$

The system remains in listening mode while the Peak to Peak is below the adaptive threshold. When the Peak to Peak exceeds the threshold, the device enters masking mode. The baseline is updated during normal listening so the device can adapt to the room's ambient noise level.

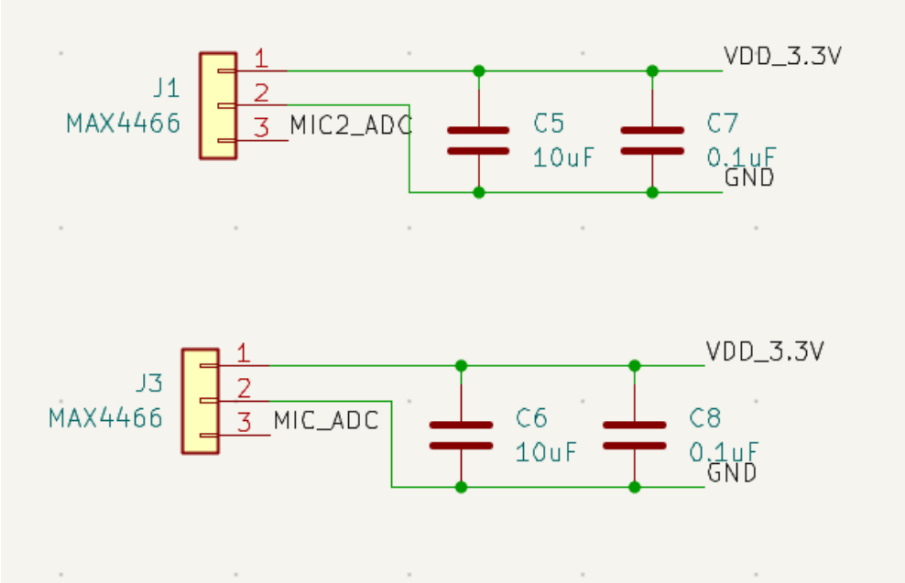

Figure 2: Circuit Schematic of components and connections involved for the Audio Sensing Subsystem

2.4 Processing Subsystem

The ESP32 handles both sound analysis and noise generation. The firmware uses one core for microphone sampling and analysis and the other core for audio output so that the masking noise stays smooth while the system continues listening.

The basic firmware flow is:

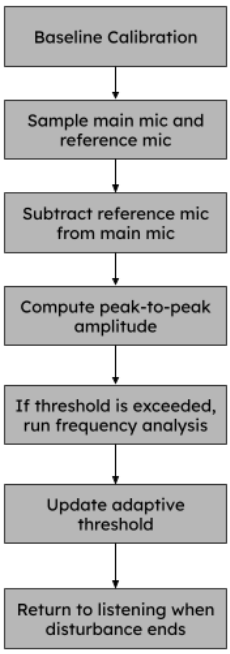

Figure 3: Flow Chart of Code Execution

To determine the disturbance frequency, the system applies a Hann window before performing an FFT:

$$w[i] = 0.5 - 0.5\cos\left(\frac{2\pi i}{N-1}\right)$$

The FFT result is used to estimate which frequency band contains the disturbance. The system then computes a radix-2 FFT. The firmware groups energy into six broad bands: 80-180 Hz, 180-360 Hz, 360-720 Hz, 720-1400 Hz, 1400-2800 Hz, and 2800-5000 Hz. Broad bands were chosen instead of a single narrow peak because bedroom disturbances vary widely and the short 256-sample frame gives a frequency-bin spacing of 62.5 Hz. The band targets are smoothed before publication so that the masking sound moves gradually rather than retuning abruptly. The output noise is then shaped toward that range instead of always playing the same masking sound.

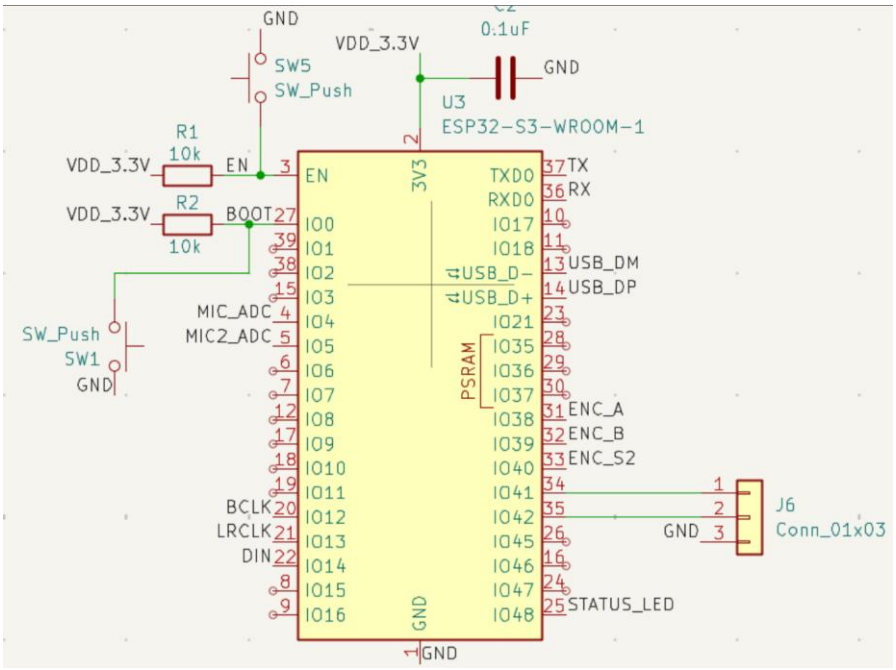

Figure 4: Circuit Schematic of components and connections involved for the Processing Subsystem

2.5 Audio Output

The audio output subsystem uses the ESP32 I2S interface, an audio amplifier, and two speakers. Two speakers were used to increase coverage and reduce the directional weakness seen in the first prototype. The device generates a mixture of white, pink, and brown noise depending on the detected frequency range.

To make the output smoother and less harsh, the system uses a second-order digital biquad filter:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2]$$

A buffer strategy was also added because analysis and audio generation were running at the same time. Duplicating generated values in the output buffer helped prevent choppy sound while keeping the masking noise adaptive.

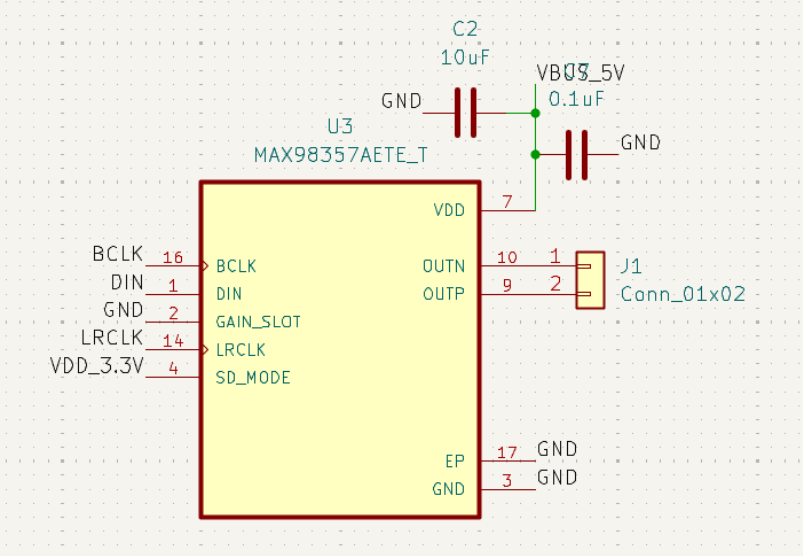

Figure 5: Circuit Schematic of components and connections involved for the Audio Output Subsystem

2.6 User Interface and Enclosure Design

The user interface includes a rotary encoder and LEDs. The rotary encoder adjusts the masking volume, while the LEDs show power and masking status. This allows the user to control the output level and quickly identify whether the device is listening or actively masking.

The final physical enclosure uses a pentagon shaped 3D-printed design that holds the PCB, microphones, speakers, and user controls in fixed positions. The pentagon shape was chosen so the two speakers could face outward in different directions, increasing the coverage area of the masking noise. The main microphone is positioned away from the speakers to reduce direct acoustic feedback, while the reference microphone is placed closer to the speaker output so it can measure the device's own masking sound for cancellation.

Foam was added inside the enclosure to dampen vibration and reduce acoustic coupling between the speakers and microphones. This helped prevent the system from falsely detecting its own output as a new sound event. The enclosure also includes openings for the speakers, microphone sensing, rotary encoder access, LEDs, and USB power so the device can operate as a compact bedside unit.

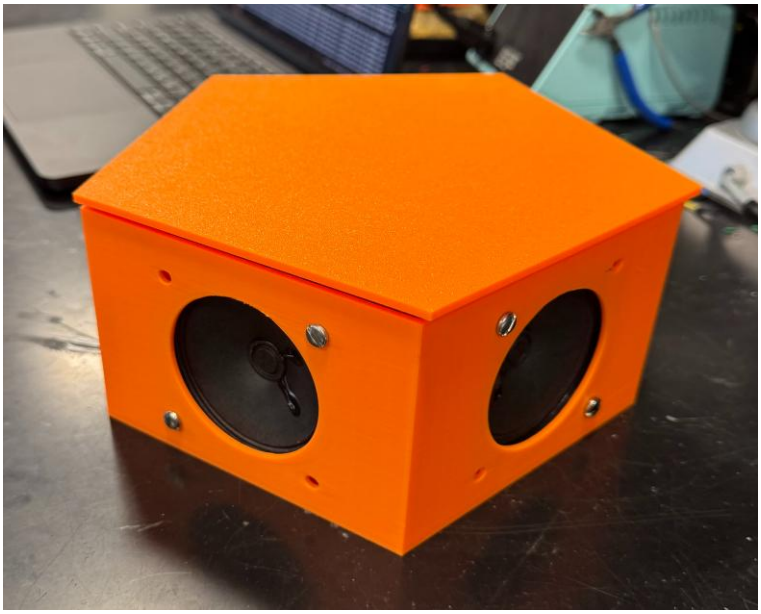

Figure 6: Final Pentagon Enclosure for Device

3. Design Verification

3.1 Overview

Testing was performed on the completed product and on each major subsystem. The goal of testing was to verify that the device could detect disruptive sound events, respond quickly, generate masking noise at an appropriate bedside volume, avoid false triggering, and run continuously overnight. The full Requirement and Verification Table is included in the appendix.

3.2 Acoustic Sensing

The acoustic sensing subsystem was tested by exposing the microphone to controlled noises and comparing the measured peak-to-peak ADC amplitude against the threshold. The system detects a sound event when:

$$P2P > \max(80,\ 1.22 \cdot Baseline\ +\ 28)$$

First, we let the system calibrate in the rom environment to establish a baseline. Then, the input volume was increased to produce input levels above the threshold. When the device began to play noise, we knew that the device had entered masking mode and was sufficiently above the threshold. This verified that the adaptive thresholding method worked as intended instead of relying on a fixed noise level.

Figure 7: No Disturbance Detected

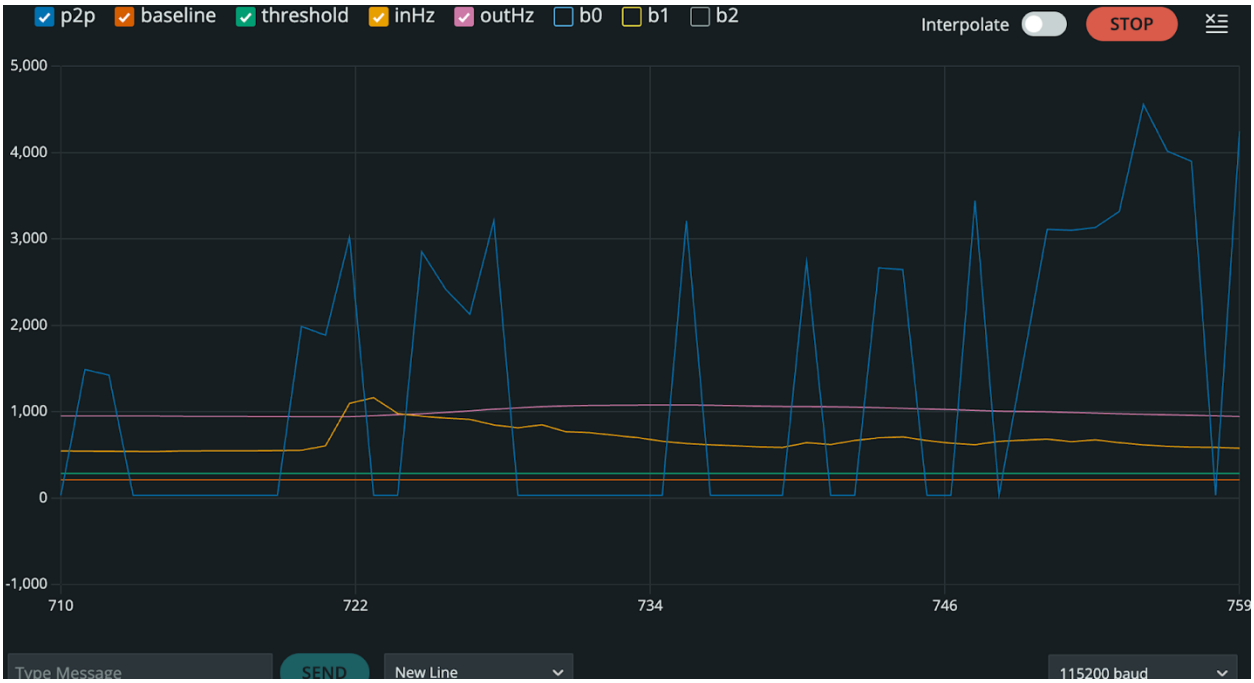

Figure 8: Disturbance Detected

The system was also tested in both quiet and louder environments. In quieter rooms around 30 dB and louder environments around 60 dB, the adaptive baseline allowed the device to continue detecting sudden disturbances without constantly being triggered from normal background noise. The second reference microphone improved this behavior by subtracting speaker feedback from the main microphone signal:

$$cleaned = mic - gain \cdot refMic$$

This helped prevent the device from responding to its own masking output.

3.3 Processing Subsystem

The processing subsystem was tested to verify that the samples were happening at the 16 kHz rate. The firmware analyzes frames of 256 samples, so the expected frame collection time is:

$$T_{frame} = \frac{256}{16000} = 0.016\ s\ = 16\ ms$$

During testing, the frame population time was measured and compared to the expected 16 ms value. This confirmed that the ADC sampling rate was consistent with the design requirement and that the FFT-based frequency analysis was receiving audio data at the intended rate.

The system response time was also measured using timer checks between the detection of a disturbance and the start of noise output. The requirement was a total response time below 50 ms, and

the completed system satisfied this target. This result shows that the device can respond quickly enough for the masking noise to begin shortly after a disruptive sound is detected.

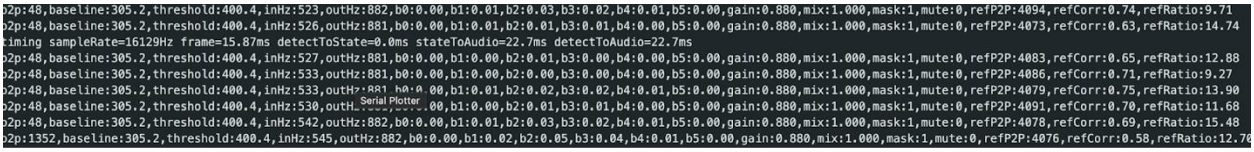

Figure 9: Serial Output with Timing and Sample Rate

3.4 Audio Output

The audio output subsystem was tested by measuring the masking noise 0.5 meters away from the system, which is the average distance that a user would set up their bedside table. The design requirement was for the output volume to fall within the usable range for sleep masking. The final verification range was 40–70 dB at 0.5 m in a quiet room, and the completed prototype was able to operate within this range.

The rotary encoder was also tested during output verification. Rotating the encoder changed the masking volume, and the output sound level was measured to confirm that the user interface actually adjusted the speaker output. This verified that the encoder was not only electrically functional, but also correctly integrated into the audio control software.

Figure 10: Maximum Sound Output Level

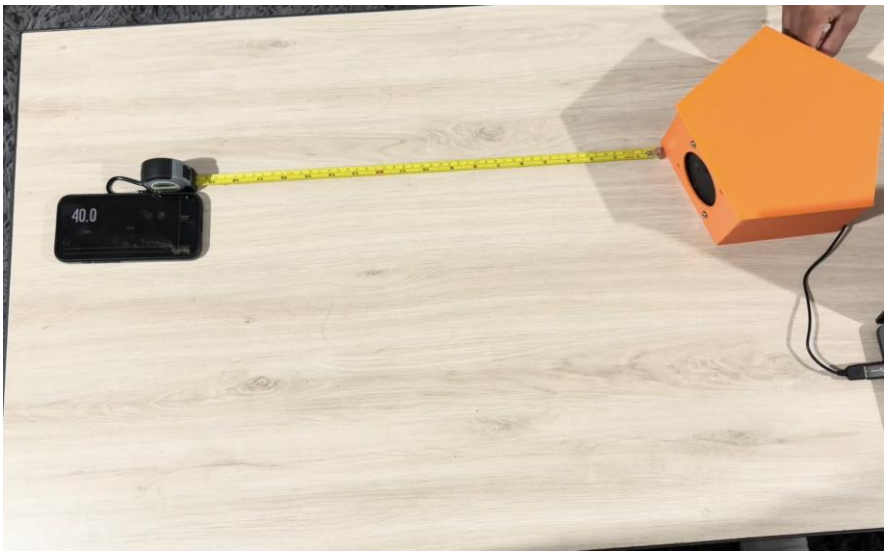

Figure 11: Minimum Sound Output

Audio quality was tested subjectively and through repeated operation of the output buffer. Early choppiness was reduced using the final buffering approach, where generated values were duplicated in the output buffer to keep playback smooth while the ESP32 continued running analysis tasks. The dual-speaker enclosure also improved the spread of masking sound compared with a single-direction output.

3.5 User Interface

The user interface was tested by confirming that the LEDs and rotary encoder were correctly configured. When the device was powered, the power LED turned on. When the system detected a sound event and entered masking mode, the status LED changed to indicate the active state. This provided a simple visual check that the firmware state matched the physical output behavior.

The rotary encoder was tested by rotating it through its range and observing changes in output volume. Since the masking device is intended for bedside use, this control is important because the user must be able to adjust the output without having to move the system entirely.

3.6 Continuous Operation Testing

To verify continuous operation stability, the completed device was run for an 8 hour test, matching the length of an average night's sleep. During the test, the noise output was left active and device logs were used to confirm that the system remained operational. The overnight logging script captured 496 one-minute telemetry rows. The device reported 8.254 h of wall clock logging and 8.254 h of firmware uptime. Uptime increased monotonically, with 0 drops, indicating that no reboot occurred during the run. Free heap stayed within 298.6-298.9 kB, which indicates that the firmware did not leak memory during the test window.

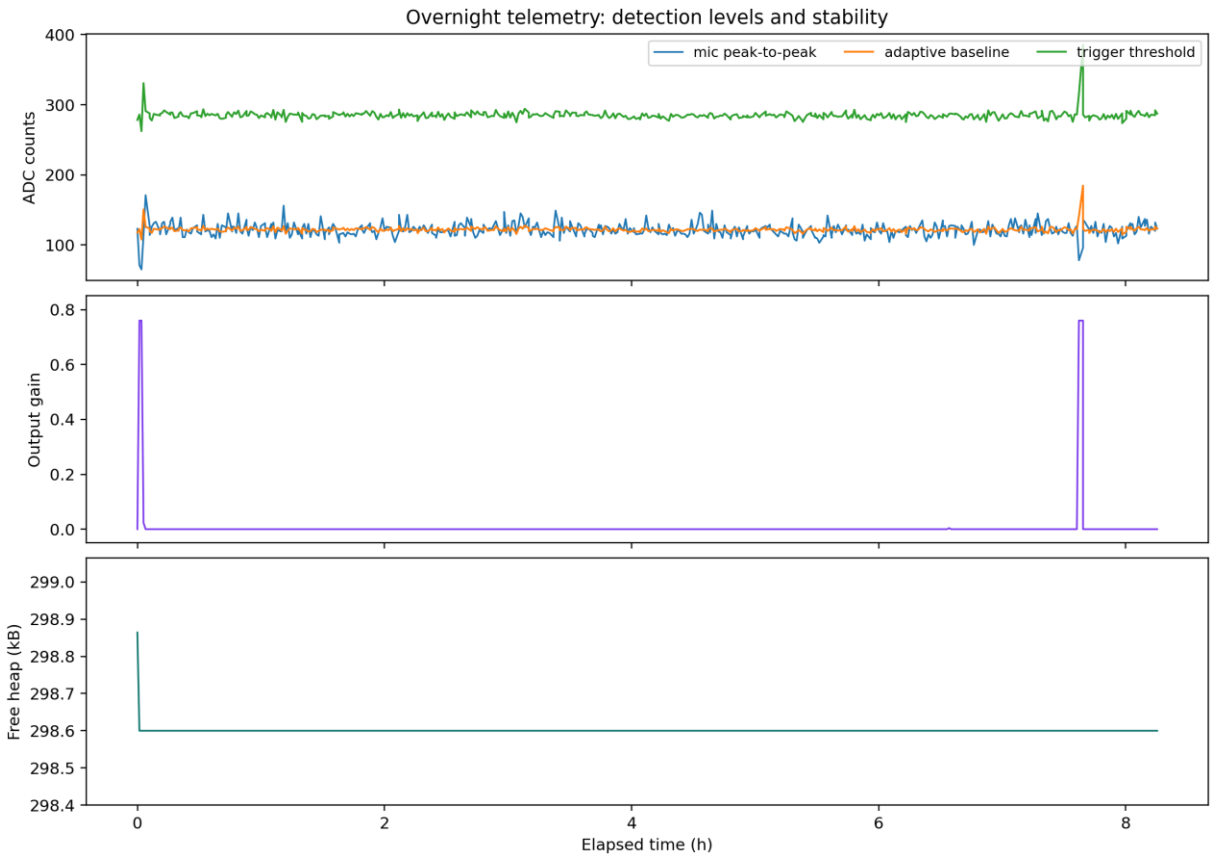

Figure 12: Overnight detection levels, output gain, and free heap.

3.7 Signal-Processing Simulation

The simulation study tests the algorithm in a controlled simulation. It tests short time spectral analysis, disturbance band estimation, shaped colored-noise generation, and before/after spectral comparison. The test contains a low frequency thump, an 820 Hz tonal disturbance, and a higher frequency burst. The 820 Hz tone prominence decreased from 35.3 dB before masking to 15.8 dB after masking, a 19.5 dB reduction.

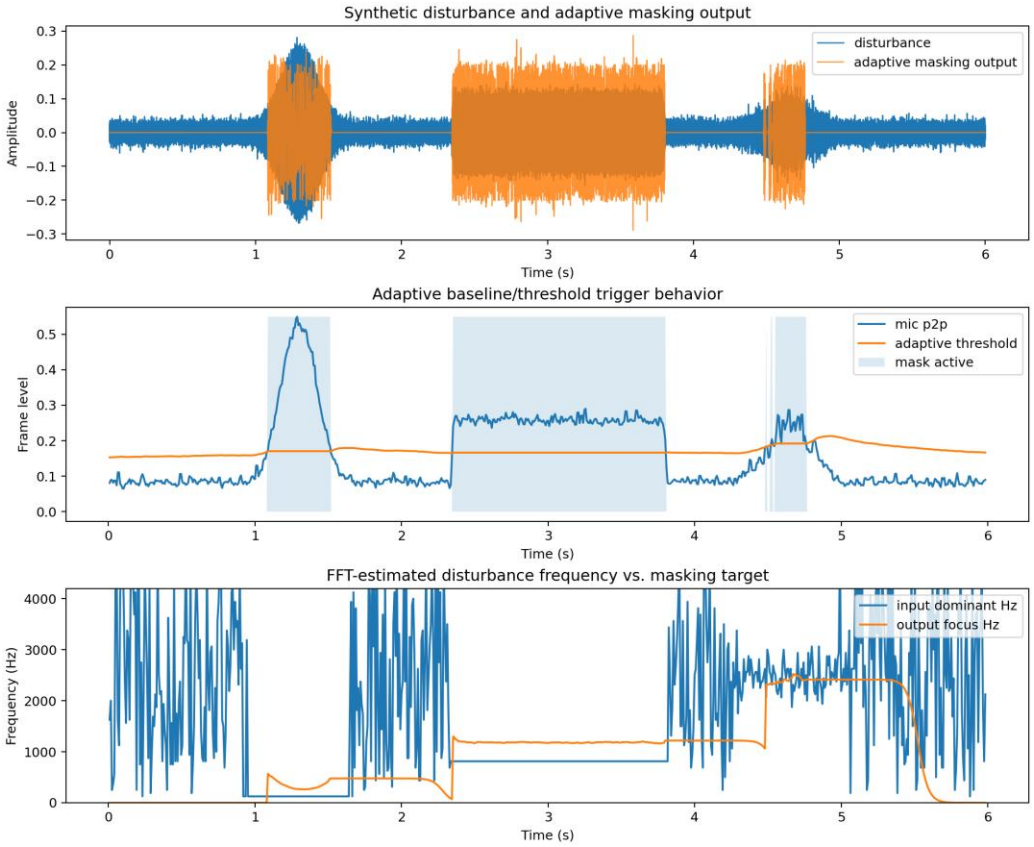

Figure 13. Simulated disturbance, adaptive masking output, trigger threshold, and estimated output focus.

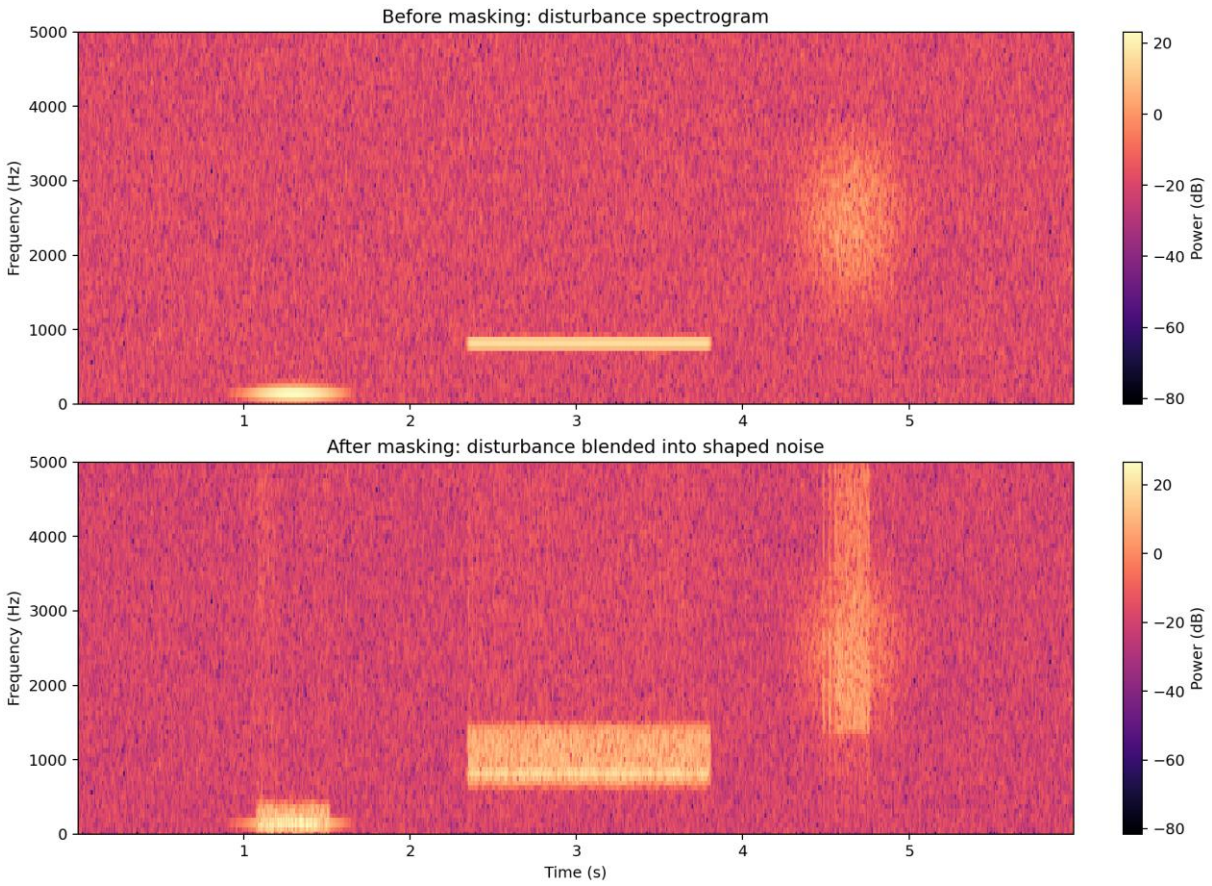

Figure 14. Spectrogram comparison before and after adaptive masking.

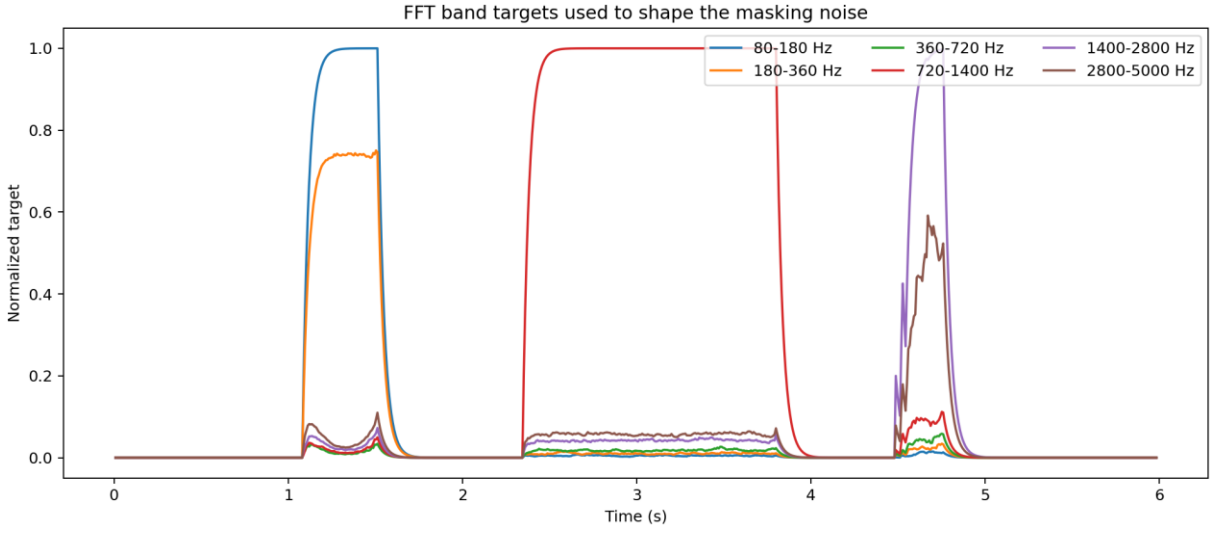

Figure 15. Six normalized FFT band targets used by the adaptive masking algorithm.

4. Costs

4.1 Parts

Parts Costs

Part	Manufacturer	Part #	Quantity	Cost ($)
ESP32-S3-WROOM-1	Espressif Systems	ESP32-S3-WROOM-1-N16	1	5.92
4Ω 3W Speaker	Adafruit	1528-2435-ND	2	3.90
MAX98357A Amp	Maxim Integrated	1528-1696-ND	1	5.95
MAX4466 Mic Amp	Maxim Integrated	1528-1013-ND	2	13.90
PEC11R Switch Encoder	Alps Alpine	4809-EC11E15244G1-ND	1	4.49
AMS1117-3.3	UMW	5272-AMS1117-3.3DKRND	2	0.54
Capacitor - 10µF / 20% / 10V (0603)	Cal-Chip	2571-GMC10X5R106M10NTTRND	4	0.40
Capacitor - 0.1µF 10% / 10V (0402)	YAGEO	C0402C104K8PACTU	3	0.30
Resistor - 330Ω (0603)	YAGEO	RT0603FRE07330RL	2	0.20
Resistor - 10kΩ (0603)	YAGEO	311-10KGRDKR-ND	2	0.20
Resistor - 5.1kΩ 5%(1/8W) (0805)	Stackpole Electronics	RMCF0805JT5K10	2	0.20
Resistor - 2.2kΩ 5%(1/8W) (0805)	Stackpole Electronics	RMCF0805JT2K20	1	0.10
LED - RED LTSTC150CKT (1206)	LITE-ON	LTST-C150CKT	2	0.28
Switch - Tactile	Littelfuse	PTS645SL43SMTR92 LFS	2	0.72
Connector - Micro USB-B	Amphenol	10118194-0001LF	1	0.43
PLA FILAMENT - 1.75MM	Hatchbox	N/A	1	28.00
Total				**65.33**

4.2 Labor

We can expect an hourly wage of $45 for a UIUC ECE grad. This gives us a salary of $45/hr×2.5×42 hours = $4725 per team member. Multiplying this amount by the three team members gives us $14175 in total labor cost.

4.3 Total

If we add the labor costs as well as the costs associated with buying all the parts, it comes to $14240.33. If we were able to mass produce this system, we would probably buy the individual parts wholesale and in bulk to reduce the cost it would take to manufacture one unit.

5. Conclusion

Catching Z's is an adaptive sleep-masking device that detects disruptive sounds and responds with targeted masking noise. The final prototype uses adaptive threshold detection and FFT-based frequency analysis to bucket incoming noise events and output a sound at the correct frequency to mask it.

The completed system met the major goals of the project. It was able to detect sound events using an adaptive baseline, sample audio at the required rate, respond within the required time limit, produce adjustable masking noise at bedside distance, and operate continuously for 8 hours.

In future iterations, the device could be improved by refining the frequency classification algorithm so more bins are created to better sort the incoming/outgoing frequency. Additionally, we could test with more real nighttime disturbances to better simulate the worst case night for our users. Finally, we can redesign the enclosure for manufacturability. If stronger performance is needed in louder rooms, the detection threshold range or speaker output could also be modified. Despite these possible improvements, the final prototype verifies that the main Catching Z's design is technically feasible and meets the project's key requirements.

5.1 Ethical considerations

Following the IEEE Code of Ethics, the design should prioritize user safety by keeping the masking sound within the tested bedside volume range and avoiding claims that the device is a medical treatment for sleep disorders.

Privacy is also important because the device listens to the room environment. To reduce this concern, the prototype processes microphone data locally on the ESP32 instead of storing or transmitting room audio. This makes the design less invasive and helps ensure that the sensing system is used only for real-time sound detection.

More broadly, the project could have positive societal value for people living in noisy dorms or rundown apartments where sleep disruptions are common. Future versions should also consider environmental impact. This can be done, for example, by designing the enclosure with more sustainable materials.

References

[1] B. Berglund, T. Lindvall, and D. H. Schwela, "Guidelines for Community Noise," World Health Organization, Geneva, Switzerland, 1999. [Online]. Available: https://apps.who.int/iris/handle/10665/66217

[2] ESP32-S3-WROOM-1 / ESP32-S3-WROOM-1U Datasheet, v1.1, Espressif Systems, Shanghai, China, 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

[3] MAX4465–MAX4469: Low-Cost, Micropower, SC70/SOT23-8, Microphone Preamplifiers with Complete Shutdown, Maxim Integrated, San Jose, CA, USA, 2001. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX4465-MAX4469.pdf

[4] MAX98357A/MAX98357B: Tiny, Low-Cost, PCM Class D Amplifier with Class AB Performance, Maxim Integrated, San Jose, CA, USA, 2015. [Online]. Available: https://www.analog.com/media/en/technicaldocumentation/data-sheets/MAX98357A-MAX98357B.pdf

[5] PEC11R Series - Contacting Encoder, Bourns, Inc., Riverside, CA, USA, 2015. [Online]. Available: https://www.bourns.com/docs/Product-Datasheets/PEC11R.pdf

[6] AMS1117: 1A Low Dropout Voltage Regulator, Advanced Monolithic Systems, Livermore, CA, USA, 2016.

Appendix A Requirements and Verification Table

Requirement	Verification	Verified
The system should detect a sound event when the smoothed microphone peak-to-peak amplitude exceeds an adaptive threshold that is the larger of two values, either 80 ADC counts, or 1.22 times the moving average baseline plus 28 ADC counts.	• Use a calibrated sound source to establish a steady baseline and record the baseline's peak-to-peak value • Increase the source volume to produce different input levels above and below the threshold and verify by polling ESP32	Yes
ADC should sample microphone output at 16 kHz.	• Every frame, it analyzes 256 audio samples from the mic. If at 16 kHz, each frame should take 16 ms. So, measure how long the frame takes to populate and then compares it to 16 ms.	Yes
Total system response time should be less than 50 ms.	• Use a timer library to measure the interval between reading and producing a noise on the respective pins.	Yes
The system should default to listening to the room and only change to masking when the threshold is met.	• Run the system in a room for an hour. • Make sure that there are no false positives that cause the state of the system to change.	Yes
The volume level at 0.5 meters should be between 40 and 70 dB in a quiet room when the device is functional.	• Set a microphone up and test the bounds to make sure it is between that.	Yes
The device must be able to operate for the entire 8 hours of night.	• Check for power via the USB-B port. • Then, turn the noise on for the whole 8 hours and check if it operates.	Yes

	• Have Device logs to ensure that it was operational the whole night	
The rotary encoder should adjust the masking volume.	• Rotate the encoder and measure how loud the noise is.	Yes
The LEDs should provide accurate feedback.	• Once the system is powered on, confirm that the power LED is on. • Confirm that the status LED color reflects that the system is in Masking Mode	Yes