

UNIVERSITY OF WATERLOO

Faculty of Engineering
Department of Electrical and Computer Engineering

Quietr: Active Noise Control for Rooms

Final Report
Group 2022.06

Emma Blatt	[REDACTED]
Fey Asan	[REDACTED]
Hanna Muratovic	[REDACTED]
Matthew Nielsen	[REDACTED]
Yoav Arbiv	[REDACTED]

Consultant: Derek Wright
23 March 2022

Abstract

Insufficient sleep (poor quality and short duration) is associated with a range of adverse health effects, including cardiovascular disease, depression, irritability, and diminished well-being. In a 2002 study, more than one-third of Canadians reported poor sleep quality and trouble falling asleep. Since many people live in shared accommodations or densely populated areas, urban noise and loud sounds from roommates are common barriers to getting enough sleep. This project aims to design an active noise control system that can be easily installed in a room to lessen the impact of incoming ambient noise for one target during sleep. The project uses an acoustic transducer subsystem to measure noise entering a room. These sound waves are characterized by a microcontroller using a noise control algorithm and the appropriate out-of-phase signal is transmitted through a speaker to reduce noise. Existing noise control solutions require the user to wear headphones or earplugs or install insulating material during home construction for soundproofing. These options are expensive and often inconvenient. The main advantage of this project design over alternatives is that it aims to perform active noise control in a room rather than in a small enclosure (e.g. cars, planes) or with headphones.

Acknowledgements

Thank you to our advisor, Derek Wright, and to our lab instructor, Kim Pope, for providing guidance throughout the process of designing our capstone project.

Table of Contents

Abstract	1
Acknowledgements	1
1 High Level Description	5
1.1 Motivation	5
1.2 Project Objective	5
1.3 Block Diagram	6
1.3.1 Overall Working of the System	6
1.3.2 Transducer Subsystem	6
1.3.3 Acoustic Signal Processing Subsystem	7
1.3.4 Device Power Subsystem	7
1.3.5 User Control Subsystem	7
1.3.6 Adaptive Power Management Subsystem	7
2 Project Specifications	8
2.1 Functional Specifications	8
2.2 Non-functional Specifications	9
3 Detailed Design	9
3.1 Transducer Subsystem Design	9
3.1.1 Frequency Control Range	10
3.1.2 Microphone Selection	10
3.1.3 Speaker Selection	11
3.2 Acoustic Signal Processing Subsystem Design	12
3.2.1 Active Noise Cancellation	13
3.2.2 Device Calibration	15
3.3 Device Power Subsystem Design	16
3.3.1 Power Source Selection	18
3.3.2 Prototype Power Distribution Board	19
3.4 User Control Subsystem Design	20

3.4.1	Device Activation/Deactivation	21
3.4.2	Low Power Mode	21
3.4.3	Design Alternatives Considered	21
3.5	Adaptive Power Management Subsystem Design	22
4	Prototype Data	23
4.1	Transducer Data	23
4.2	Microphone Data	24
4.3	ANC Algorithm	25
4.4	Device Calibration	26
4.5	Device Power	27
4.6	Adaptive Power Management Algorithm	28
5	Discussion and Conclusions	28
5.1	Evaluation of Final Design	28
5.2	Use of Advanced Knowledge	29
5.3	Creativity, Novelty, Elegance	29
5.4	Quality of Risk Assessment	29
5.5	Student Workload	30
A	Appendix	33
A.1	Bill of Materials	33
A.2	Product Documentation	33
A.2.1	Product Information	33
A.2.2	User Manual	33
A.2.3	Safety Considerations	34
A.3	Device Footprint	34

List of Figures

1	Quietr system block diagram.	6
2	Block diagram of transducer subsystem.	9
3	Equal Loudness Contour measured in Sound Pressure Level (SPL) [1]	10

4	Speaker frequency response measured in Sound Pressure Level at 1 W and 0.5 m . . .	11
5	Block Diagram of Acoustic Signal Processing Subsystem	12
6	Schematic sheet for battery and battery protection IC.	20
7	Schematic sheet for voltage regulators.	20
8	High-level overview of the adaptive power management subsystem.	22
9	Physical Prototype	23
10	Prototype Battery, Power Distribution	23
11	Transducer data flow diagram	23
12	Transducer data flow diagram	24
13	Transducer data flow diagram	24
14	Simulation of LMS Algorithm to Cancel Noise Emitted by an Air Conditioner . . .	25
15	Simulated Magnitude Spectra of Residual Noise vs Input Disturbance	25
16	Convergence Profile of Untuned vs Tuned Model	26
17	Simulation of Feedback Path Estimation	26
18	Impulse Response of Feedback Path Estimate	27
19	Simulation of adaptive power management	28
20	Dimensions of selected enclosure	34
21	Dimensions of an A4 sheet	34

List of Tables

1	Functional Specifications	8
2	Non-Functional Specifications	9
3	Decision Matrix for Microphone Selection	11
4	Power Requirements for Key Components	18
5	Decision Matrix for User Interface Design/Components	22
6	Load testing of Device Power PCB	27
7	Student hours.	30
8	Bill of Materials	33

1 High Level Description

1.1 Motivation

Insufficient sleep (poor quality and short duration) is associated with a range of adverse health outcomes, including cardiovascular disease, depression, irritability, and reduced well-being. In a 2002 study, more than one-third of Canadians reported poor sleep quality and trouble falling asleep [2]. Since many people live in shared accommodations or urban areas, urban noise and loud sounds are common barriers to getting enough sleep.

1.2 Project Objective

This project aims to design an active noise control system that can be easily installed in a room to lessen the impact of incoming ambient noise (e.g. roommates, construction, traffic, loud appliances) for one target during sleep. The system uses noise-cancelling technology to measure and cancel sound waves. Existing solutions require the user to wear headphones or earplugs to reduce noise or install insulating material during home construction to soundproof the room. These options are expensive and generally unsatisfactory. The main advantage of this design over alternatives is that it aims to perform active noise control in a room rather than in a small enclosure (e.g, cars, planes) or with headphones.

1.3 Block Diagram

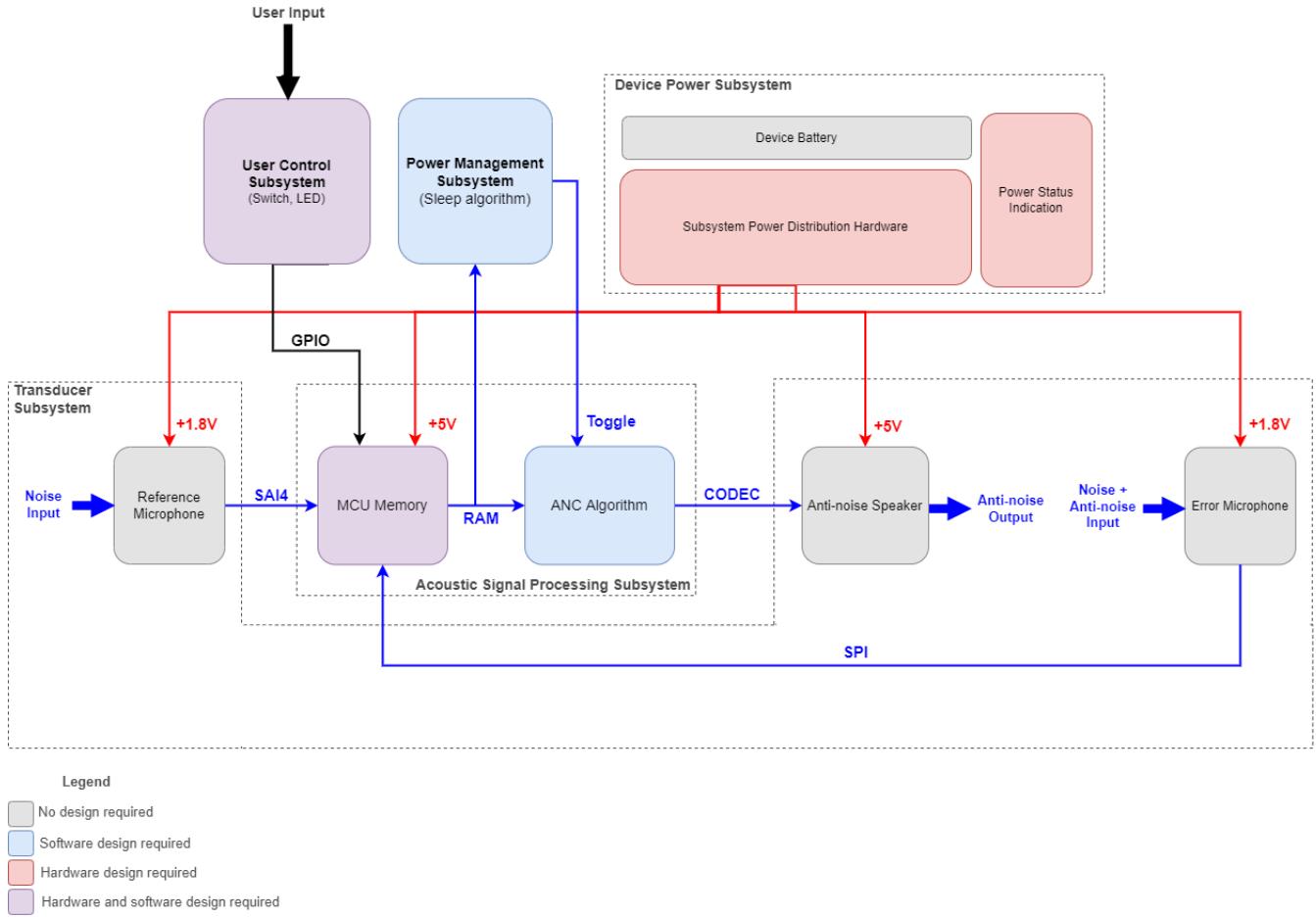


Figure 1: Quietr system block diagram.

1.3.1 Overall Working of the System

The device uses microphones to provide data to the signal processing system, which then calculates anti-noise and emits it through the speaker. The adaptive power algorithm analyzes that sampled data and determines if anti-noise should be emitted. Power is provided to the system using the device power subsystem and the user interface controls operation of the device.

1.3.2 Transducer Subsystem

This subsystem consists of an acoustic sensor which measures ambient noise within a room. This signal is then conditioned for processing in the microcontroller. After processing, the appropriate out-of-phase signal is transmitted to the electroacoustic transducer and converted to sound to destructively interfere with the noise. An error microphone is positioned near the user, which detects residual noise and creates a feedback loop to the microcontroller.

1.3.3 Acoustic Signal Processing Subsystem

This subsystem performs all the signal processing required to measure, characterize, and control ambient noise. The input signal is captured by an acoustic sensor within the Transducer subsystem and passed into a microcontroller. Processing of the signal includes determining phase delay, signal power, and signal frequency using the appropriate active noise controlling algorithms. Once the input signal has been characterized, the microcontroller outputs an out-of-phase signal which will be converted to sound using the electroacoustic transducer within the Transducer subsystem. The error microphone positioned near the user is used to fine-tune the generated anti-noise.

1.3.4 Device Power Subsystem

This subsystem is used to distribute power throughout the device. Power is required for all hardware components, namely the microcontroller, signal conditioning circuits, and the sensors/transducers. Device power can be sourced from a battery or a wall plug, and the power status is displayed using a visual indicator. Additionally, the user is able to control the power consumption of the device by setting the operation to “sleep” or “low power” mode. In this situation, an adaptive power management algorithm is executed to limit device activation if ambient noise is consistently below a pre-programmed threshold.

1.3.5 User Control Subsystem

This subsystem allows the user to manually adjust the operation mode and other features of the device. The user can interact with the device through a set of buttons, a touchscreen, a combination of the two, etc. The input signal is processed by a microcontroller and the requested process is executed. Potential modes of operation include ON/OFF, calibration, sleep (low-power), and volume control.

1.3.6 Adaptive Power Management Subsystem

This subsystem regulates the use of the speaker transducer. As the system operates, it tracks the last minute of samples. If that window falls below an average of 30 dB, then the system stops emitting anti-noise. During that time, samples are still collected and stored. If the window average rises above 30 dB again, the system transitions back into emitting anti-noise. This will help regulate power use, as the speaker is a large source of power consumption in the system. If the room is quiet, then power can be saved by disabling speaker use. The limit of 30 dB is chosen as it is roughly the sound of whispering in a room [3].

2 Project Specifications

2.1 Functional Specifications

ID	Subsystem	Specification	Essential?	Description
FS1	Transducer	Noise detection	✓	The device must have the ability to sense noise to reduce it. The sensor should have a minimum sensitivity of -25 dB [4, 5].
FS2	Acoustic Signal Processing, Transducer, User Control	Device calibration	✓	The device must have the ability to characterize the transducers to account for the frequency response of each sensor and speaker, as well as the acoustics of the room. The device will sweep every frequency in the cancellation range and monitor how the input amplitude compares to the output amplitude in order to create a coefficient mapping for use in noise-cancelling operation.
FS3	Acoustic Signal Processing	Convergence time	✓	The Active Noise Control Algorithm should be able to converge on an appropriate set of weights that minimizes error in no less than 10s.
FS4	Adaptive Power Management	Modes of operation	✗	A sleep mode feature to reduce power consumption. If the last minute of consecutive samples are below an average of 30 dB (roughly the volume of nearby whispering) [3], enter a mode that stops active noise reduction. When the window of samples rises above the threshold, the device returns to active noise reduction.
FS5	Acoustic Signal Processing, Transducer	Active noise control/reduction	✓	The algorithm designed must reduce noise by 5 dB at the target location for one user for noise in the frequency range of 300 Hz - 1 kHz [1].
FS6	Device Power	Power requirements	✓	The device must be able to operate longer than 8 hours (an average sleep cycle). For example, if a 1 W speaker is utilized, a battery with capacity of 8 Wh is needed [6].
FS7	User Control	Software interface to control the system	✓	A user-controlled interface to power the device on/off, and initiate calibration routines. The interface can be designed for use on-device, via a USB web portal, an app with BT, etc.
FS8	Acoustic Signal Processing, Transducer	Size of room for noise control	✓	The device can reduce noise within a 3 m radius, based on the average height of a bedroom of 2.6 m [7]. The error microphone will be attached to a 3 m cable.

Table 1: Functional Specifications

2.2 Non-functional Specifications

ID	Specification	Essential?	Description
NFS1	User-facing software components should be easy to operate.	✓	The user interface should be intuitive and must have a minimalist UI/UX design to calibrate the device and control power/volume.
NFS2	Documentation of the product's use and safety considerations.	✗	A user manual that includes product information, installation, set-up, calibration, and safe use of the device.
NFS3	Device must be easy to physically install and uninstall.	✓	The device can be installed and uninstalled from its operational location without professional technical support.
NFS4	Device must be small and unobtrusive.	✓	Device(s) footprint must not be larger than an A4 sheet (297 x 210 mm) [8] and have a visually simple design.
NFS5	Device must be affordable.	✗	System price must be substantially less than \$4000 (price of existing competitor products) [9].
NFS6	Device power must be reliable.	✓	Input voltage and current for the device must be monitored to ensure operation is within component limits.

Table 2: Non-Functional Specifications

3 Detailed Design

3.1 Transducer Subsystem Design

Active noise control requires a tight feedback loop to effectively control persistent noise at the location of the target. The transducer system uses a closed-loop negative feedback control system to reduce noise near the target. Based on an initial assumption that the noise originates within one wavelength of the reference microphone (i.e. "at-the-source"), the microphone will sample and process the noise to generate a weighted, inverted, and time-delayed version of the noise signal. Then, the speaker will output the anti-noise to cancel the original noise at the position of the target. The remaining noise picked up by the error microphone is then subtracted from the input noise, under a quasi-continuous assumption that the noise source will remain relatively constant for a short period of time.

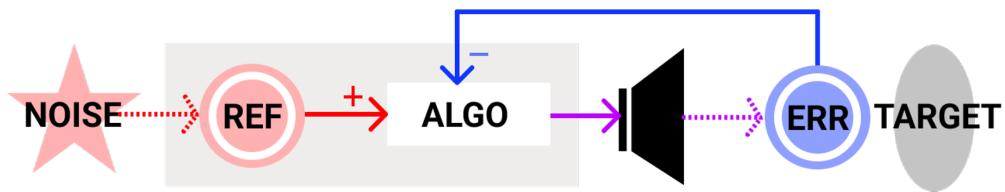


Figure 2: Block diagram of transducer subsystem.

3.1.1 Frequency Control Range

Most commercial noise control products use active noise cancellation below 1 kHz and passive noise cancellation above 1 kHz. This is because the anti-noise lags behind the noise due to processing and propagation delays, and this lag increases at higher frequencies due to shorter wavelengths. An equal loudness contour dictates the relative sound pressure levels of audio tones at different frequencies. If two tones with the same gain are played, a 1 kHz tone will sound louder than a 100 Hz tone, as depicted by Figure 3.

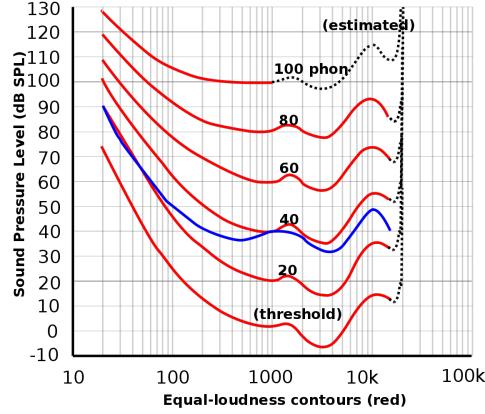


Figure 3: Equal Loudness Contour measured in Sound Pressure Level (SPL) [1]

It is easier to perform active noise control with low frequency signals because their larger wavelength translates to a greater error margin. An error in phase shift will be less noticeable on a lower frequency wave than a higher frequency wave. Therefore, the frequency range of interest was limited to 300 Hz to 1 kHz.

3.1.2 Microphone Selection

The reference and error microphones will be interfaced with the microcontroller to track noise. The two main types of microphone technologies are MEMS and Electret Condensers (ECM). MEMS microphones are made with a micro-electro-mechanical system component. Sound pressure waves cause movement of the MEMS diaphragm, and the change in capacitance is converted to an electrical signal. Alternatively, the electret condenser is a material with a fixed surface charge that is placed near a conductive plate and creates a capacitor. Sound pressure will cause the capacitance to change, and the voltage across the capacitor will change proportional to the fixed charge. The MEMS microphone is more suitable for this application because the technology is newer, making it faster and less expensive than ECM microphones [10].

The sensor data can be transmitted in the analog or digital domain. Even though both domains use the same technology, they are interfaced differently and therefore have different performance advantages [11]. Table 3 depicts the design options and trade-offs. The most important figures of merit are power consumption, cost, noise immunity, and complexity. Since the device needs to be able to operate for approximately 8 hours, it is important to track and limit power consumption from all transducers. Since the error microphone is placed up to 3 m away from the processing

unit, noise immunity and SNR are important factors to consider. Cost and complexity are relevant to the scope of this project because there is a time limit to the design and testing of the device.

Type of Microphone Evaluation Criteria	Weight	Analog		Digital	
		External ADC	PDM	I2S	
Power Consumption	0.25	0	-	-	
Noise Immunity	0.25	0	+	+	
Cost	0.20	0	-	-	
Complexity	0.10	0	-	+	
Flexibility	0.05	0	+	0	
Circuit Area	0.15	0	-	+	
Weighted Total	1	0	-0.1	0.05	

Table 3: Decision Matrix for Microphone Selection

The reference microphone is located on the Discovery Kit, and it is interfaced to a Serial Audio Interface (SAI). This is an STM32-specific interface that allows for PDM interfacing of up to eight digital microphones. Once the microphone data is read in as PDM, which is a bit stream where the relative density of the pulses corresponds to the analog signal's amplitude [12]. This data can be filtered to obtain the amplitude information. The data is converted by filtering and decimating data values to obtain PCM, pulse code modulation, which encodes the wave's amplitude values as pulses.

3.1.3 Speaker Selection

The main figures of merit for loudspeakers are power consumption and the sound pressure level frequency response. The power required will be dictated by the maximum level of noise that the device can cancel. The frequency response shows the sensitivity to sound pressure versus frequency. Figure 4 illustrates that for lower frequencies to achieve the same level of output as higher frequencies, the speaker output power will need to be higher to compensate for lower sensitivity. To achieve a high-power output that will reduce noise for the target, the input signal needs to be amplified. There is an integrated 2 W Class-D amplifier in the on-board audio codec that is used for anti-noise amplification to output sound to the $8\ \Omega$ speaker.

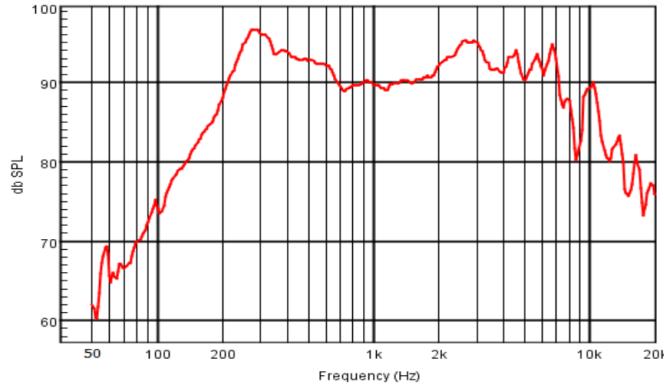


Figure 4: Speaker frequency response measured in Sound Pressure Level at 1 W and 0.5 m.
[13]

3.2 Acoustic Signal Processing Subsystem Design

Since noise is transformed as it travels through the acoustic domain, the signal received at the reference mic will differ from the signal received at the error mic. To account for this, a mechanism for predicting this transformation must be formed. In signal processing terms, these transformations are the result of an unknown plant acting on the signal. This subsystem is responsible for modelling these unknown plants in a process known as system identification. A block diagram for the acoustic signal processing subsystem is presented in Figure 5.

Figure 5 identifies three plants that could transform the signal as it travels through the acoustic domain. The first plant, $P(z)$, is the path from the reference mic to the error mic. This is the primary path that our algorithm is responsible for approximating. The second plant, $S(z)$, is the path from the cancelling loudspeaker to the observer (error mic). This is known as the secondary path which is used by the FxLMS algorithm (an alternative that will be discussed). The third plant, $F(z)$, is the feedback path from the cancelling loudspeaker to the reference mic. The modelling of this path is important to perform feedback neutralization.

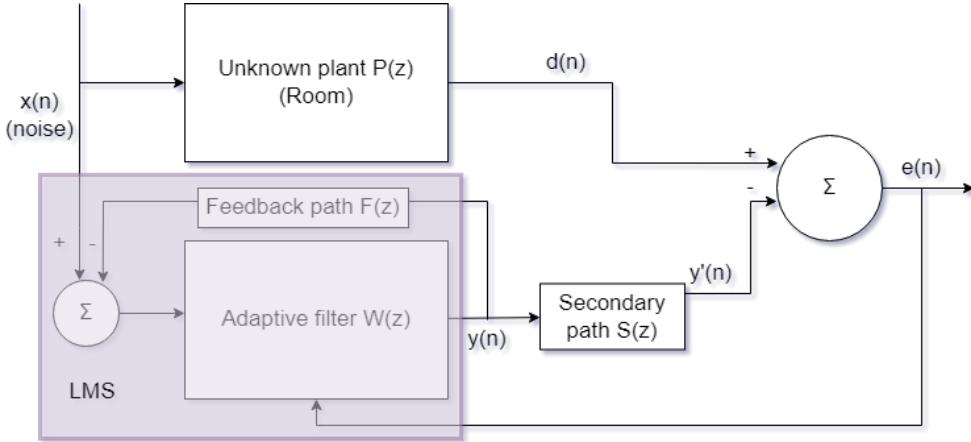


Figure 5: Block Diagram of Acoustic Signal Processing Subsystem

The plants in the above block diagram are modelled using adaptive FIR (Finite Impulse Response) filters of varying orders. This decision was made over IIR (Infinite Impulse Response) filters because FIR filters are unconditionally stable - the device shouldn't introduce any unpleasant noise. The taps of these adaptive FIR filters are calibrated to reduce the error in their approximation. This is done using the LMS (Least Mean Square) algorithm in a process known as stochastic gradient descent. The taps of the primary path are continually adapted during the device's operation through a process called online modelling. The taps of the feedback path are determined prior to device use in a process known as offline modelling.

The error of a sample is given by the following equation, note that w and x are the weight and state vectors of the filter:

$$e(n) = d(n) - [\vec{w}(n) \cdot \vec{x}(n)]$$

The LMS algorithm seeks to minimize the instantaneous error of a mean square cost function:

$$\text{cost}(n) = E[e^2(n)]$$

$$\xi(n) = e^2(n)$$

Performing gradient descent on this cost function yields the following weight update formula:

$$\begin{aligned}\vec{w}(n+1) &= \vec{w}(n) \frac{\mu}{2} \nabla \xi(n) \\ \vec{w}(n+1) &= \vec{w}(n) + \mu \vec{x}(n) e(n)\end{aligned}$$

3.2.1 Active Noise Cancellation

The ANC (Active Noise Cancellation) subsystem is tasked with predicting the optimal anti-noise to emit based on the current input. The optimal anti-noise is one which perfectly cancels the primary noise at the receiver. As previously mentioned, this is accomplished through the use of an adaptive FIR filter. The overall algorithm is known as LMS ANC. The ANC algorithm described in this section satisfies FS5.

DMA (Direct Memory Access) controllers are used to automatically transmit/receive audio samples through the respective SAI (Serial Audio Interface) peripherals. When a certain window size of samples is received, a DMA callback updates a flag which indicates to the main loop that the next batch of samples is ready for processing.

The estimated feedback component of the previous output is computed in the background through the use of the on-board FMAC unit (Filter Math Accelerator). This specialized hardware unit is designed to perform digital filter operations very quickly. This is realized through the use of cascading MAC (Multiply Accumulate) units. The order of our estimated feedback plant $F(z)$ is kept within the constraints of the FMAC unit.

Processing of the next batch of outputs can begin once the next window of input samples is ready and the computation of previous feedback component is ready. The feedback component is subtracted from the reference mic readings to perform feedback neutralization. The next batch of outputs is then computed by running the reference and error mic readings through the adaptive FIR filter that models the primary path.

An alternative design considered was the filtered-X LMS (Least Means-Squared) adaptive noise control filter, which is a common implementation for active noise cancellation. This algorithm introduces an additional filter stage to compensate for the effect of the secondary path $S(z)$. The algorithm introduces additional memory and computational overhead which, in turn, increases latency. Ultimately, the decision was made to reduce complexity to minimize technical difficulty, which is a risk identified in the specifications document for this project. The main operation of the LMS ANC algorithm is realized through the following C++ code:

```
State state = State::Waiting;
AdaptiveFIR<800> filter;      // primary path model

while (true) { // infinite main loop
    // Wait for new samples and fmac to finish computing feedback estimate
    while (state == State::Waiting || fmac_busy() ) {}
```

```

// with ping-pong buffers we need to identify which half we're currently using
size_t offset = (state == State::ProcessFirstHalf) ? 0 : 1024;

// Invalidate cache entries for the input buffers so we see the new data
SCB_CleanDCache_by_Addr(&ref_buff[offset], WINDOW_SIZE);
SCB_CleanDCache_by_Addr(&err_buff[offset], WINDOW_SIZE);

for (size_t i = 0; i < WINDOW_SIZE; ++i) {
    ref_buff[i + offset] -= fmac_output[i]; // feedback neutralization
}

// Now we can compute the next antinoise using our adaptive filter
filter.process_samples(&ref_buff[offset], &err_buff[offset], &out_buff[offset]);

// Invalidate cache to start new DMA transfer
SCB_CleanDCache_by_Addr(&out_buff[offset], WINDOW_SIZE);

// Start processing next feedback component using FMAC
fmac_start(&out_buff[offset], &fmac_output, WINDOW_SIZE);

state = State::Waiting;
}

```

The adaptive FIR filter is realized through the following C++ code:

```

extern "C" {
    float32_t rng(float32_t min, float32_t max); // HAL level rng
}

// @short This struct represents an adaptive FIR (finite impulse response) filter.
// @note The runtime of this algorithm is O(NK)
//       (where N is the number of taps and K is the number of samples)
template <size_t N>
struct AdaptiveFIR {
    Array<float32_t, N> weights;
    Queue<float32_t, N> state;

    AdaptiveFIR() {
        for (size_t i = 0; i < N; ++i) {
            weights[i] = rng(-1.0 / N, 1.0 / N); // small random number
            state[i]; // 0
        }
    }

    void process_samples(const float32_t ref[NUM_SAMPLES],
                        const float32_t err[NUM_SAMPLES],

```

```

                float32_t out[NUM_SAMPLES]) {
    for (size_t i = 0; i < NUM_SAMPLES; ++i) {
        state.push(ref[i]);
        out[i] = std::inner_product(weights.begin(), weights.end(), state.begin());
        adapt(err[i], 0.001); // adapt weights using step size of 0.001
    }
}

void adapt(float32_t err, float32_t mu) {
    float32_t delta = err * mu;
    std::transform(weights.begin(), weights.end(), state.begin(),
                  [&](float32_t weight, float32_t state) {
        return weight + (delta * state);
    });
}
};

```

3.2.2 Device Calibration

As previously mentioned, the device achieves feedback neutralization using a model of the feedback path $F(z)$. This model is realized through a FIR filter whose weights are trained outside of normal operation in a process known as offline modelling.

The training process begins by initializing an adaptive FIR filter of a given order. Next, a known waveform is then played through the cancelling loudspeakers and subsequently read from the reference microphone. The data transfer to/from the audio peripherals is similar to what was outlined in the previous section (SAI transfer using DMA). Next, the model is used to predict what the reference mic readings will be. As before, the error in this prediction is used to adapt the filter's weights. The resulting weights will remain constant during operation until the next calibration routine is run. Because of this, the weights can be copied to a specialized hardware unit (FMAC) to speed up filter computation. The device calibration routine is realized through the following C++ code:

```

AdaptiveFIR<200> model;
file_pointer = file_start; // Pointer to calibration waveform in SRAM

while (file_pointer < file_end) {
    // Wait for a section of the DMA transfer to be complete
    while (state == State::Waiting) {}

    size_t offset = (state == State::ProcessFirstHalf) ? 0 : 1024;
    size_t offset_prev = (state == State::ProcessFirstHalf) ? 1024 : 0;

    // Predict the current input from the previous output and error
    model.process_samples(&out_buff[offset_prev], &err_buff[offset_prev],
                         &prediction_buff[offset]);
}

```

```

// Clear cache entry so we get new reference mic readings
SCB_CleanDCache_by_Addr(&ref_buff[offset], WINDOW_SIZE);

for (size_t i = 0; i < WINDOW_SIZE; ++i) { // Calculate the current error
    err_buff[offset + i] = ref_buff[offset + i] - prediction_buff[offset + i];
}

for (size_t i = 0; i < WINDOW_SIZE; ++i) { // Copy the next audio samples to out_buff
    out_buff[offset + i] = (file_pointer < file_end) ? *file_pointer++ : AUDIO_ZERO;
}

// Invalidate cache entry to kick off new DMA transfer
SCB_CleanDCache_by_Addr(&out_buff[offset], WINDOW_SIZE);

state = State::Waiting;
}

```

The result of calibration is a set of weights (taps) for the FIR filter that models the feedback path from the cancelling loudspeaker to the reference microphone. This calibration routine satisfies FS2.

3.3 Device Power Subsystem Design

The design of this subsystem is based on the power requirements of the key components within this device, namely the microcontroller, signal conditioning circuits, and sensors/transducers. These requirements are used to select a suitable power source and the voltage regulators for the device.

Microcontroller

The QuietR device uses the STM32H735 microcontroller unit to perform the signal processing necessary to control noise. The main power supply, V_{DD}, is set to 3.3 V and powers a majority of the I/O ports, all internal clock sources, and the independent watchdog. An internal, regulated voltage named V_{CORE} supplies the three domains within the STM32: CPU1 Cortex-M7, CPU2 Cortex-M4, and D3. V_{CORE} can be supplied by connecting V_{DD} to either the SMPS step-down converter or the LDO found within the microcontroller unit. The regulators can also be entirely bypassed and V_{CORE} can be supplied using a separate, external source [14]. The lowest power consumption is observed when the step-down converter is used to power V_{CORE} [14], thus this is the selected supply configuration for the active noise cancellation device.

The STM32 also offers a variety of memory options including instruction tightly-coupled memory (ITCM) and data tightly-coupled memory (DTCM), flash, and cache. Since TCM offers the lowest latency due to its proximity to the processor, the device utilizes this form of memory for executing the noise control algorithm. The type of memory used to store and process data has a significant effect on the overall power consumption of the microcontroller unit [14].

With these configurations in mind and assuming the microcontroller is operating in Run mode at the highest performance (voltage scaling VOS0) at a clock rate of 550 MHz with all peripherals

enabled, the provided datasheet can be used to determine the typical current consumption, I_{DD} [14]. The typical current during these operating conditions is 125 mA.

The typical power consumption of the STM32H735 microcontroller is calculated:

$$\begin{aligned} P_{MCU} &= V_{DD}I_{DD} \\ &= 3.3(0.125) = 0.4125 \text{ W} \end{aligned}$$

Microphone(s)

The selected component for both the reference and error microphones is the DMM-4026-B-I2S-R, manufactured by PUI Audio, Inc. According to the datasheet, the microphone is rated at 1.8 V and 1 mA during Normal Mode operation [15].

The typical power consumption of both microphones is calculated:

$$\begin{aligned} P_{mic} &= 2V_{DD}I_{DD} \\ &= 2(1.8)(0.001) = 0.0036 \text{ W} \end{aligned}$$

Speaker

The selected component for the anti-noise speaker is the AS07708PS-2-R, manufactured by PUI Audio, Inc. This 8Ω speaker is rated for an input power of 4 W; however this may not be the typical power that is used by the QuietR device. The typical power required can be calculated using the sensitivity rating of the speaker. The sensitivity rating is a measure of the characteristic sound pressure level in decibels per watt per meter. According to Figure 4 in Section 3.1.3, the AS07708PS-2-R speaker has a sensitivity rating of 90 dB_{SPL}/1 W/0.5 m at 1 kHz [13]. This means that at 1 watt of power, the speaker generates a sound pressure level of 90 dB at a distance of 0.5 meters [16].

The relationship between input power (W) and sound pressure level (dB_{SPL}) is defined by the following equation, where p_n is the desired sound pressure level, p_0 is the characteristic sound pressure level, and P_{ref} is the reference input power [16]:

$$p_n = p_0 + 10 \log_{10}\left(\frac{P_{speaker}}{P_{ref}}\right) \quad (1)$$

Additionally, increasing the distance from the speaker will decrease the effective sound pressure level according to the following equation, where d_{ref} is the reference distance [16]:

$$p_n = p_0 - 20 \log_{10}\left(\frac{d_{speaker}}{d_{ref}}\right) \quad (2)$$

In order to calculate the typical input power for the QuietR device, some assumptions must be made regarding the operating conditions. First, it is assumed that the device is attempting to cancel a constant 1 kHz noise signal. This frequency is selected because within the frequency range of the QuietR device, the speaker has the lowest sensitivity at 1 kHz according to Figure 4. Next, it is assumed that the sound pressure level 3 meters away from the noise source is 70 dB. The distance and decibel values are selected based on the average floor-to-ceiling height for residential homes and the typical sound pressure level of a high-traffic road [7, 17].

Using these assumptions, a reference sound pressure level of 90 dB_{SPL}/1 W/0.5 m, and Equations (1) and (2), the typical input power can be calculated:

$$\begin{aligned} p_n &= p_0 + 10 \log_{10}\left(\frac{P_{\text{speaker}}}{P_{\text{ref}}}\right) - 20 \log_{10}\left(\frac{d_{\text{speaker}}}{d_{\text{ref}}}\right) \\ 70 &= 90 + 10 \log_{10}(P_{\text{speaker}}) - 20 \log_{10}\left(\frac{3}{0.5}\right) \\ P_{\text{speaker}} &= 10^{\frac{20 \log_{10}(6)-20}{10}} = 0.36 \text{ W} \end{aligned}$$

The typical current consumption of the speaker can also be calculated, using the 8Ω value found in the datasheet:

$$\begin{aligned} I_{\text{speaker}} &= \sqrt{\frac{P_{\text{speaker}}}{R}} \\ &= \sqrt{\frac{0.36}{8}} = 0.212 \text{ A} \end{aligned}$$

Class D Amplifier

A Class D audio amplifier is used to control the anti-noise speaker. The theoretical efficiency of this class of amplifiers is 100% – that is, all of the power supplied to the amplifier is delivered to the load. However, due to the switching losses generated by the transistors and the non-ideality of components within the low-pass filter, the true efficiency of a Class D amplifier ranges from 80% to 90% [18].

Assuming a worst-case amplifier efficiency of 80%, the speaker power at the amplifier input stage can be calculated:

$$P_{\text{amp}} = \frac{P_{\text{speaker}}}{\eta} = \frac{0.36}{0.8} = 0.45 \text{ W}$$

3.3.1 Power Source Selection

Table 4 outlines the calculation results of the previous section. The total typical current consumption of the device is approximately 339 mA.

Components	Voltage (V)	Current (mA)	Power (W)
Microcontroller	3.3	125	0.4125
Microphones	1.8	1	0.0036
Speaker (Amplifier Input)	N/A	212	0.4500

Table 4: Power Requirements for Key Components

The calculations performed in the previous section assumed that a custom board could be designed for the Quietr device. Unfortunately due to the chip shortage, our group is unable to purchase the STM32H735 microcontroller that the device design is based upon. As a result, the prototype leverages an evaluation board for the STM32H735 microcontroller: STM32H735G-DK. This board

integrates the MCU, Class-D amplifier, and reference microphone, and requires a 5 V input [19]. The total power consumption of the board when used in the Quietr device should be similar to the estimations performed in the previous section.

With the power requirements calculated (5 V, 339 mA), options for a power source may now be explored.

AC/DC Wall Adapter

The component selected for this alternative is the VEL05US050-US-MB AC/DC Wall Mount Adapter from XP Power. This adapter is capable of providing 5 W at an output voltage of 5 V [20]. It is more than capable of providing the 339 mA required for the Quietr device. Additionally, the wall adapter is well-suited to utilize the anti-noise speaker at its maximum power input (4 W) if necessary. With this component, FS6, which outlines device power requirements, is fulfilled.

While this component can easily provide the power required by the Quietr device, using a wired power source limits where the device can be placed. According to the datasheet, the maximum wire length for the VEL05US050-US-MB AC/DC Wall Mount Adapter is 1.5 m [20]. As a result, NFS3 and NFS4 are not fulfilled.

Rechargeable Battery

The second power source alternative is a rechargeable battery. A possible battery choice is the NL1834, manufactured by Nitecore. It is a 3.7 V lithium-ion battery that is capable of providing 3.4 Ah [21]. This battery was selected based on the amp-hour requirements of the Quietr device, calculated assuming an 8 hour continuous operating period:

$$\begin{aligned} Capacity_{req} &= t_h(I_{total}) \\ &= 8(0.339) = 2.712 \text{ Ah} \end{aligned}$$

The Nitecore battery is capable of supplying 0.339 A for approximately 10.3 hours, fulfilling FS6 [21]. Additionally, the use of a rechargeable battery allows for the Quietr device to be easily placed anywhere in a room, without the limitations of a power cord. This fulfills NFS3 and NFS4. Since the rechargeable battery meets all relevant functional and non-functional requirements better than the wall adapter alternative, it is used as the power source for the Quietr device.

3.3.2 Prototype Power Distribution Board

Based on the power requirements and power source alternatives discussed in the previous subsections, a custom power PCB is designed and used in the Quietr system. The PCB utilizes two lithium-ion batteries in series to provide 7.4 V which is stepped down to 5 V using a buck regulator. The batteries provide 3.4 Ah to the system, and are monitored using a battery protection IC.

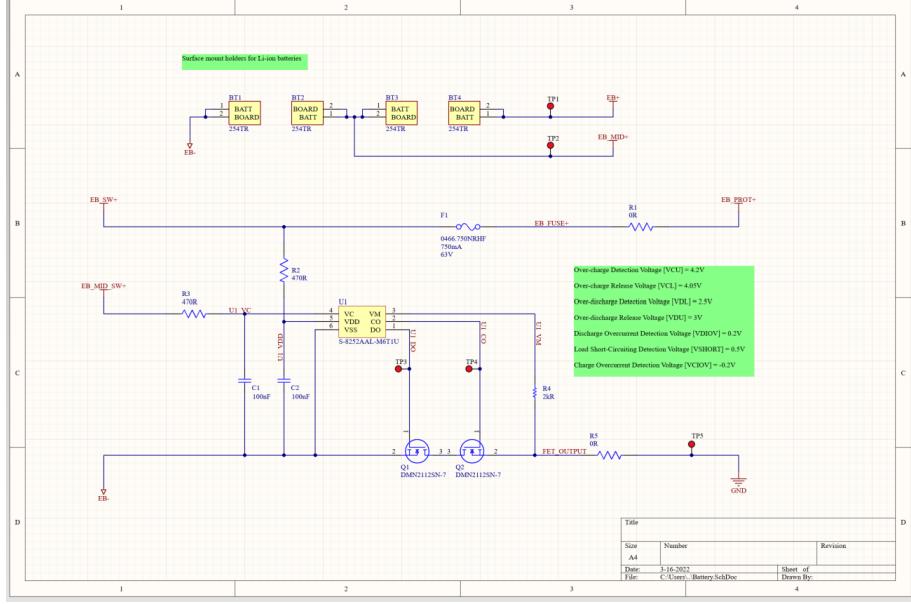


Figure 6: Schematic sheet for battery and battery protection IC.

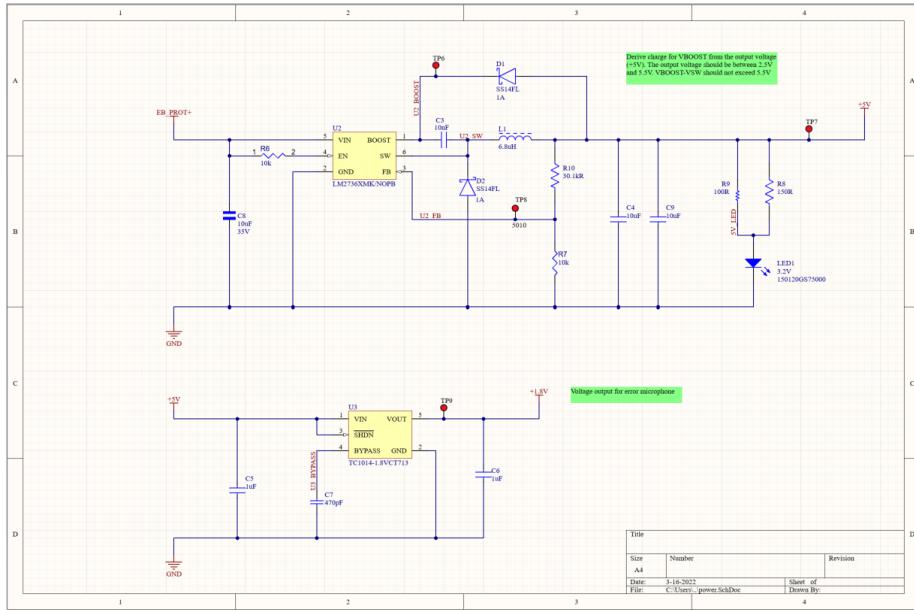


Figure 7: Schematic sheet for voltage regulators.

3.4 User Control Subsystem Design

This subsystem allows the input signal provided by the user to be processed by a microcontroller. The requested is then executed, and the results then displayed on the User Interface. The main hardware components of the system are push buttons and LEDs that perform several actions. For instance, several push buttons exist to turn the device on and off, and control other modes of operation. Calibration is also initiated via push buttons. The presence of LEDs allow the triggered

actions (through buttons) to be displayed and monitored visually, in a simple and intuitive manner.

Through the use of an Embedded Software Interface (written in C), the input specified by the user is processed and communicated with the STM32H735G microcontroller, and the results are then displayed via LEDs, resulting in a simplistic and easy to use UI.

This implementation satisfies FS7, NFS1, NFS3 and NFS4. The user control subsystem allows the device to be powered on/off, monitor device status, and initiate calibration routines through the Software Interface (FS7). The design consists of a simple UI that is intuitive, visually pleasant and minimalist. This allows the user-facing software components to be easy to operate (NFS1).

3.4.1 Device Activation/Deactivation

Within the user control subsystem, various buttons must be monitored to switch between modes and enable LEDs. Polling and interrupts were considered. Ultimately, polling the button could prove troublesome in longer device loops, so interrupts were chosen.

3.4.2 Low Power Mode

According to the adaptive Power Management Subsystem Design described in Section 3.5, the device goes into Low Power Mode if the noise level has been below 30 dB for 60 s. An LED switches on/off depending on the power state. The code below demonstrates how this is done.

```
while(1) {
    if(Low_Power_Mode() == ACTIVE) { //check if device in low power mode
        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET); // LED on
        HAL_Delay(100); //100 ms delay
    }
    else { //device not in low power mode
        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET); // LED off
        HAL_Delay(100); //100 ms delay
    }
}
```

3.4.3 Design Alternatives Considered

Several alternatives were considered while deciding on the design of the User Control Subsystem. The two main design/component alternatives to be considered for a User Interface (UI) were:

- Alternative 1: Utilization of push buttons and LEDs
- Alternative 2: Non-touch LCD screen, push buttons and LEDs

Table 5 depicts aforementioned design options and compares them with a weighted decision matrix. Some of the most important figures of merit in UI component selection are power consumption, cost, flexibility, simplicity and appearance.

Evaluation Criteria	Weight	Alternative 1	Alternative 2
Cost	0.15	+	-
Simplicity	0.25	+	-
Flexibility	0.20	-	+
Power Consumption	0.15	+	-
Appearance	0.25	-	+
Weighted Total	1	0.55	0.45

Table 5: Decision Matrix for User Interface Design/Components

Alternative 1, consisting of push buttons and LEDs is not only the simplest option to implement, but also is the most cost and power efficient. The addition of an LCD display in Alternative 2 would increase the power consumption, cost and complexity of the device. Since both alternatives possess the ability to satisfy the Functional and Non-Functional Specifications associated with this subsystem, and Alternative 1 is simpler to implement, the components to be used for the UI are chosen to be LEDs and push buttons.

3.5 Adaptive Power Management Subsystem Design

The adaptive power management subsystem shifts between two states and operates on a sliding window of audio samples measured at the reference microphone. As the system runs, the average decibel level of each sample is stored in a FIFO structure containing the last minute of audio. As seen in the Acoustic Signal Processing Subsystem Design, the system runs an infinite loop. A sliding average of the decibel level of the last minute of samples is maintained. Then, if that sliding average falls below 30 dB, the device simply does not emit anti-noise. Likewise, if the sliding average goes higher than 30 dB, the system is allowed to emit anti-noise once more. Figure 8 shows a state diagram for system operation. This design fulfills FS4, as it allows the system to adapt to its environment and limit its power use when needed.

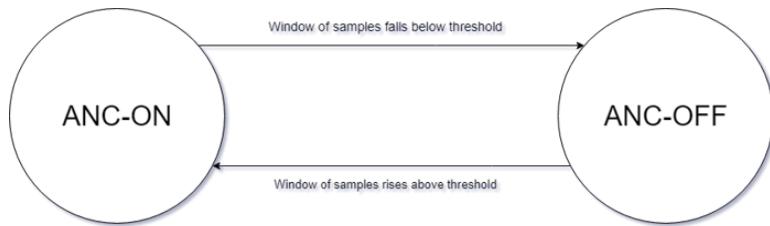


Figure 8: High-level overview of the adaptive power management subsystem.

An alternative design considered was to make use of one of the various low-power modes of the STM32. The device offers the ability to cut power to different peripherals and even the ability to turn off the clock. The device can then wake up upon the firing of a timer interrupt. As such, looking at Figure 8, the 'ANC-OFF' state would be transitioned to from 'ANC-ON' when the sliding average fell below 30dB. When this would have happened, the device would have initiated a timer for 1 minute and then entered standby mode. After a minute, the timer IRQ would fire and the current noise level would be detected - if the level had risen above 30dB, the device would re-enter active mode. Otherwise, it would go back to sleep.

Ultimately, this alternative design was replaced for a few reasons. First, it was determined that most of the power draw in the system is incurred when operating the speaker - the microcontroller and the auxiliary transducers constitute a smaller proportion of the current sunk. Therefore, similar power saving gains can be achieved with this approach with much less complexity. Second, the system using timer interrupts lacks the ability to respond to a sudden increase in noise. For example, if noise increases above the threshold 10 seconds after the system goes to sleep, then there is no ability to attenuate that noise. Contrast this against the final design, which would detect high noise and 'wake up' in order to cancel it.

4 Prototype Data

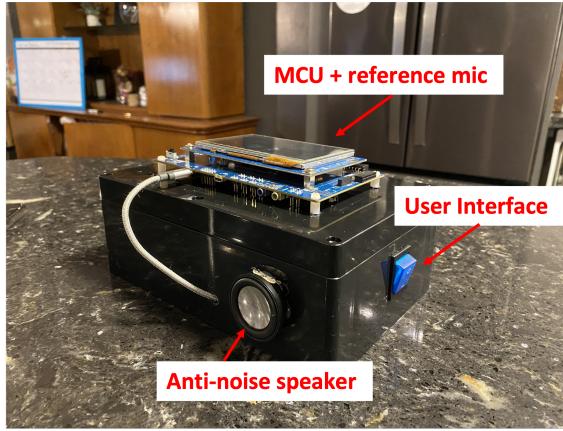


Figure 9: Physical Prototype

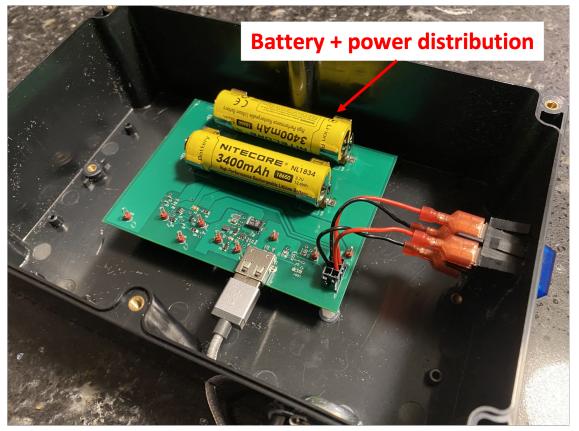


Figure 10: Prototype Battery, Power Distribution

4.1 Transducer Data

As the anti-noise from the speaker needs to arrive at the target at the same time as the noise, the data has to be transferred and processed with low latency. The frequency range for human hearing is 20 Hz to 20 kHz, and the sampling rate is required to be more than the 40 kHz Nyquist rate.

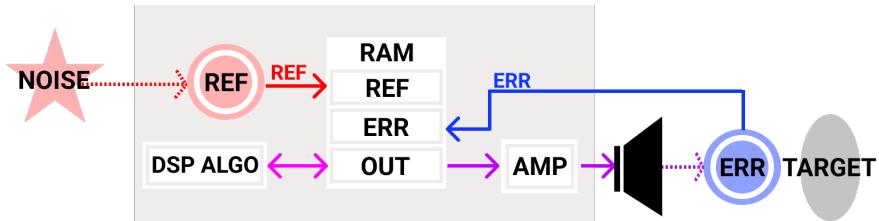


Figure 11: Transducer data flow diagram

As shown in Figure 11, the microphone data is sampled through the reference microphone into an SRAM buffer. The error microphone is read at a faster rate than the reference microphone, as per the convergence time requirements for the signal processing algorithm. The FIFO buffers that are accessible to the audio peripherals are 8 bits wide, and data is received from the microphone in

32-bit frames. To apply DSP to the waveform, the data needs to be in PCM format. This is done by filtering and decimating the PDM bitstream to achieve an audio sampling rate of 48 kHz.

4.2 Microphone Data

As shown in Figure 12, microphone data is read in 8-bit increments from the audio interface FIFO buffer and transferred into SRAM by the DMA. A PDM bitstream transmits a density of pulses, where the average value corresponds to an analog sample. This data is sampled from the reference microphone at a rate of 1.024 MBits/s, and from the error microphone at a rate of 4.096 MBits/s.

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
0x38000000	FF	00	FD	00	A5	00	A7	00	FF	00	FD	00	9D	00	9F	00	ÿ.ÿ.ÿ.ÿ.ÿ.ÿ....
0x38000010	FF	00	F9	00	59	00	5F	00	FF	00	F9	00	61	00	67	00	ÿ.ÿ.Y.ÿ.ÿ.a.g.
0x38000020	FF	00	FD	00	6D	00	6F	00	FF	00	F9	00	61	00	67	00	ÿ.ÿ.m.o.ÿ.u.a.g.
0x38000030	FF	00	FD	00	9D	00	9F	00	FF	00	F9	00	51	00	57	00	ÿ.ÿ....ÿ.Q.W.
0x38000040	FF	00	FE	00	66	00	67	00	FF	00	FE	00	8E	00	8F	00	ÿ.b.f.g.ÿ.b....
0x38000050	FF	00	FD	00	9D	00	9F	00	FF	00	FA	00	5A	00	5F	00	ÿ.ÿ....ÿ.Z....
0x38000060	FF	00	F8	00	E8	00	EF	00	FF	00	F9	00	91	00	97	00	ÿ.ø.ë.ÿ.ÿ....
0x38000070	FF	00	FD	00	6D	00	6F	00	FF	00	F9	00	61	00	67	00	ÿ.ÿ.m.o.ÿ.u.a.g.
0x38000080	FF	00	FD	00	6D	00	6F	00	FF	00	F9	00	59	00	5F	00	ÿ.ÿ.m.o.ÿ.u.Y....
0x38000090	FF	00	F9	00	51	00	57	00	FF	00	FE	00	A6	00	A7	00	ÿ.ÿ.Q.W.ÿ.ÿ.ÿ.ÿ.ÿ....
0x380000A0	FF	00	FD	00	65	00	67	00	FF	00	FD	00	65	00	67	00	ÿ.ÿ.e.g.ÿ.ÿ.e.g.
0x380000B0	FF	00	FD	00	95	00	97	00	FF	00	FE	00	66	00	67	00	ÿ.ÿ....ÿ.b.f.g.
0x380000C0	FF	00	FE	00	56	00	57	00	FF	00	FD	00	95	00	97	00	ÿ.ÿ.p.V.W.ÿ.ÿ....
0x380000D0	FF	00	FE	00	96	00	97	00	FF	00	FD	00	6D	00	6F	00	ÿ.ÿ.b....ÿ.y.mo.
0x380000E0	FF	00	F9	00	59	00	5F	00	FF	00	F9	00	69	00	6F	00	ÿ.ÿ.Y.ÿ.ÿ.i.o.

Figure 12: Transducer data flow diagram

Figure 13 shows a PCM data read after the PDM data is filtered and decimated. PCM data stores values that correspond to analog samples, which is useful data that can be processed and output to the audio codec. The decimated data has an audio sampling rate of 48 kHz.

DMA's are constantly monitoring the SRAM buffers to transfer new data along in the flow. This is important as it means that algorithm processing can happen at the same time as the data transfer.

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
0x24000000	00	00	00	00	00	90	C0	01	01	A8	50	01	01	38	11	03À..P.8..
0x24000010	03	B8	F0	00	03	48	B1	02	04	60	41	02	04	F0	01	04	,.ð.H..À.ð..
0x24000020	06	C0	A0	00	06	50	61	02	07	68	F1	01	07	F8	B1	03	À..Pa.hñ.ø.ø..
0x24000030	09	78	91	01	09	08	52	03	0A	20	E2	02	0A	B0	A2	04	...x....R..à..ø..
0x24000040	0A	C0	60	00	0A	50	21	02	0B	68	B1	01	0B	F8	71	03	À..P.l.h.ø.ø..
0x24000050	0D	78	51	01	0D	08	12	03	0E	20	A2	02	0E	B0	62	04	xQ.....ø..
0x24000060	10	80	01	01	10	C2	02	11	28	52	02	11	B8	12	04À.(R.,..	
0x24000070	13	38	F2	01	13	C8	B2	03	14	E0	42	03	14	70	03	05	.ø.È.à.B.p..
0x24000080	0F	B8	30	00	0F	48	F1	01	10	60	81	01	10	F0	41	03	,.ø.Hñ..ø.ø..
0x24000090	12	70	21	01	12	00	E2	02	13	18	72	02	13	A8	32	04	.ø.ø.ø.ø..
0x240000A0	15	78	D1	00	15	08	92	02	16	20	22	02	16	B0	E2	03	xÑ.....ø..
0x240000B0	18	30	C2	01	18	C0	82	03	19	D8	12	03	19	68	D3	04	.ø.À..ø..ø..
0x240000C0	19	78	91	00	19	08	52	02	1A	20	E2	01	1A	B0	A2	03	x....R..à..ø..
0x240000D0	1C	30	82	01	1C	C0	42	03	1D	D8	D2	02	1D	68	93	04	.ø.À.B.ø.ø..
0x240000E0	1F	38	32	01	1F	C8	F2	02	20	E0	82	02	20	70	43	04	.82.È.ø.à..ø.C..

Figure 13: Transducer data flow diagram

4.3 ANC Algorithm

A simulation was used to demonstrate the ANC (Active Noise Cancellation) algorithm outlined in subsubsection 3.2.1. The result of this simulation was plotted to yield Figure 14 and Figure 15. The average attenuation of 10 dB in Figure 15 indicates that FS5 has been satisfied.

Parameters such as filter order and step size were then varied to demonstrate their effect on convergence time. The results of this simulation were plotted to yield Figure 16. From Figure 16, it can be seen that a tuned model reaches convergence in under 10.0 seconds, thus satisfying FS3.

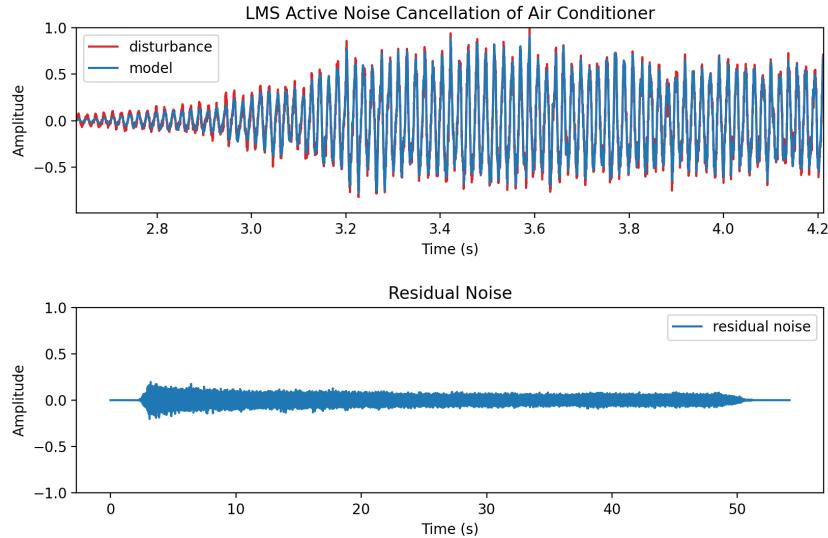


Figure 14: Simulation of LMS Algorithm to Cancel Noise Emitted by an Air Conditioner

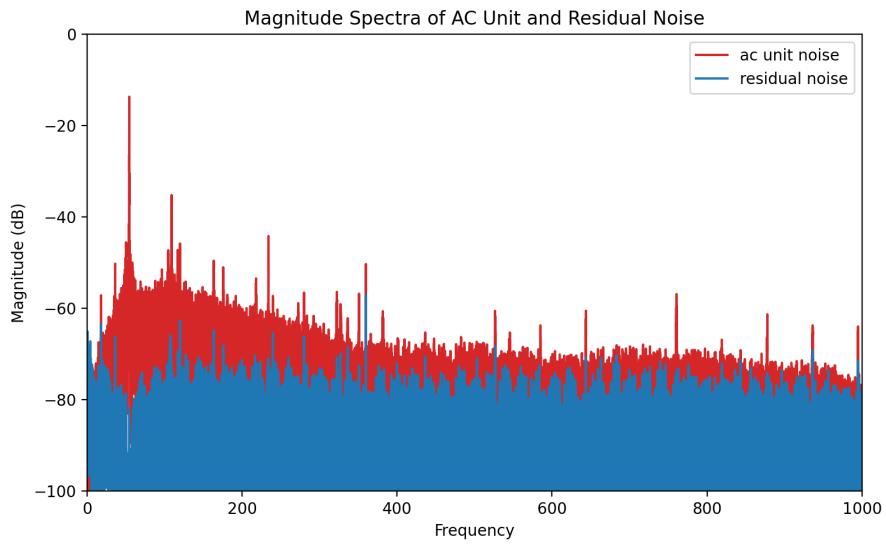


Figure 15: Simulated Magnitude Spectra of Residual Noise vs Input Disturbance

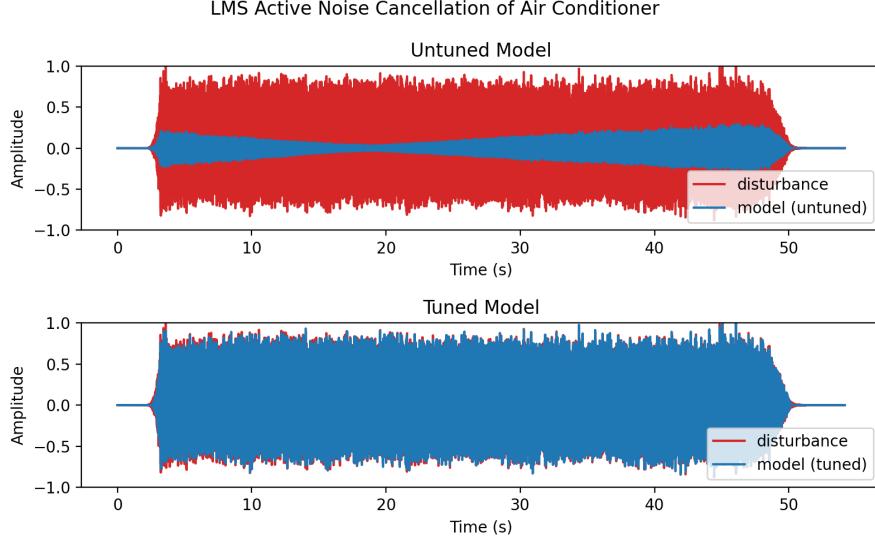


Figure 16: Convergence Profile of Untuned vs Tuned Model

4.4 Device Calibration

A simulation was used to demonstrate the process of device calibration outlined in subsubsection 3.2.2 which satisfies FS2. The calibration waveform used by the simulation is a composite tone made up of frequencies within the target cancellation band. The result of this simulation was plotted to yield Figure 17. In this figure we can see that the error in the model is initially large but decreases over time as the weights converge on optimum values. It can be seen that the model converges on these optimum weights after a period of 0.10 seconds. For completeness, the impulse response of the modelled plant was plotted to yield Figure 18:

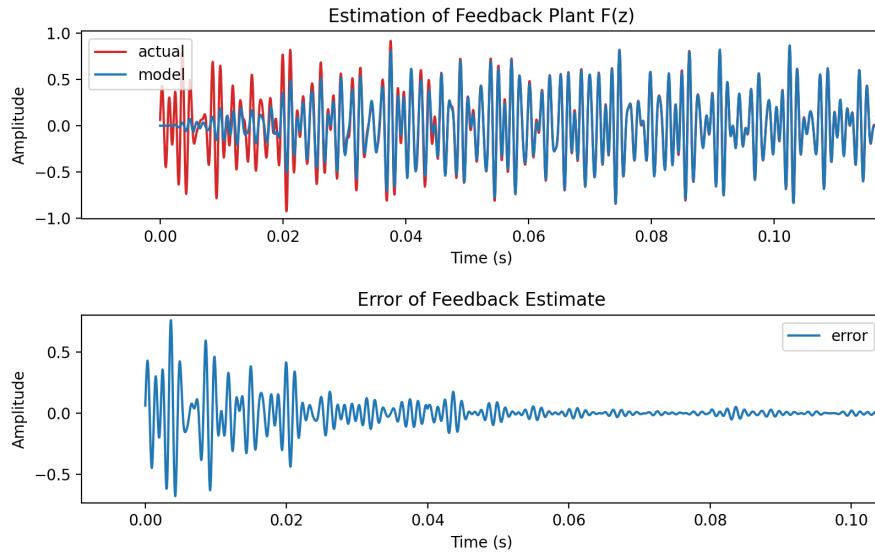


Figure 17: Simulation of Feedback Path Estimation

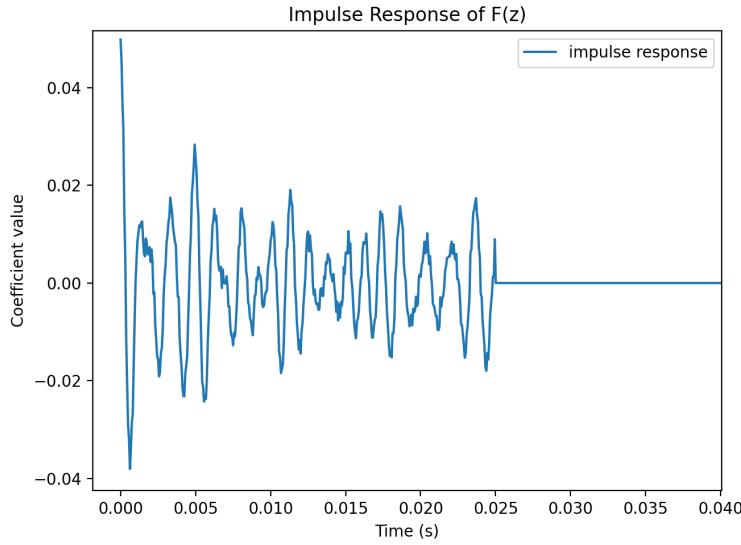


Figure 18: Impulse Response of Feedback Path Estimate

4.5 Device Power

The device power PCB described in subsubsection 3.3.2 was tested in the lab to determine the maximum current that it can provide to the rest of the system. The output current is limited by the battery protection IC, which is included in the PCB design to meet device safety requirements (NFS6). The point at which the output current exceeded the limits of the battery protection IC was found by connecting a passive load to the output of the PCB and decreasing the value of this load until the IC tripped and disconnected the battery from the rest of the system. The results from this test can be found in Table 6.

Load	Output Voltage (V)	Output Current (mA)
50	5.18	100
20	5.12	250
10	5.06	500
9	0.05	550

Table 6: Load testing of Device Power PCB

The output voltage of the PCB drops to 0 V when the output current reaches 550 mA, meaning the battery protection IC has tripped. This current threshold is much greater than the actual current requirements of the system, which is approximately 339 mA. Since the Nitecore NL1834 batteries provide up to 3.4 Ah of energy and the PCB can handle a load of 339 mA, FS6 is easily met. Additionally, Table 6 proves that NFS6 is met by showing that the battery protection IC monitors the batteries and disconnects the system when safe operating conditions are exceeded.

4.6 Adaptive Power Management Algorithm

A simulation was used to show the value of this power-saving algorithm specified in FS4. The simulation took in a .wav file and implemented the adaptive power algorithm described in subsection 3.5. As seen in Figure 19, the algorithm spends 70% of its time below the threshold, and thus saves a great deal of power. Without the algorithm, the speaker would be attempting to emit anti-noise below the threshold, draining battery when it isn't needed.

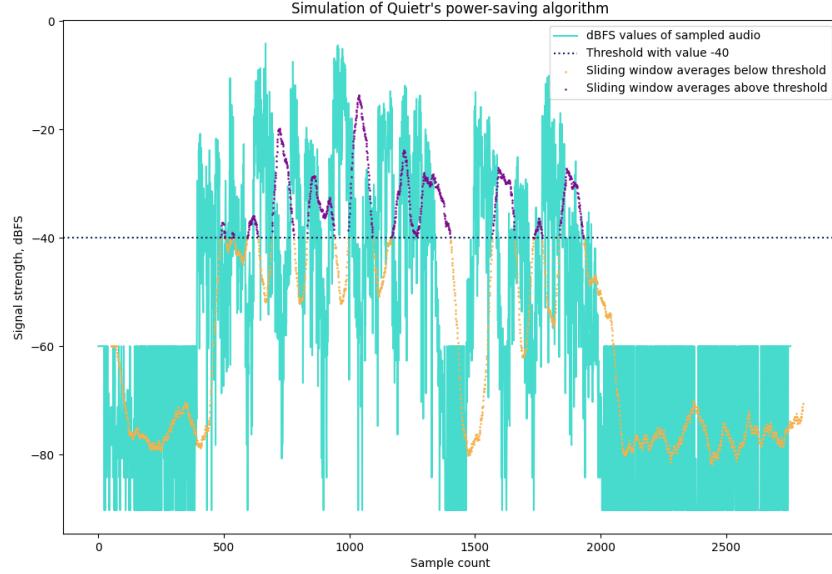


Figure 19: Simulation of adaptive power management

5 Discussion and Conclusions

5.1 Evaluation of Final Design

The transducer subsystem meets the functional specifications outlined by FS1, FS5, FS8 by selecting components that meet the sensor sensitivity requirements, and designing the wiring to enable a 3 m range. In terms of non-functional specifications, the transducer subsystem helps to meet NFS3, NFS4, and NFS5, which describe the cost and size of the device. The Acoustic Signal Processing subsystem design covers FS2, FS3, and FS5. The Adaptive Power Management subsystem covers FS4. Furthermore, FS6, NFS3, NFS4, and NFS6 are all met by the Device Power subsystem, which utilizes an appropriately sized lithium-ion battery with a battery protection circuit to deliver power safely to the Quietr device. As shown in the Appendix under Device Footprint, dimensions of the enclosure that the Quietr system is housed in is smaller than an A4 sheet, thus accomplishing NFS4. The User Control Subsystem Design accomplishes specifications FS7, NFS1, and NFS3. It is intuitive to use, allows the device to be powered on/off, monitor device status, and initiate calibration routines through the Software Interface. Overall, all subsystems of Quietr

are designed with affordability in mind, and therefore satisfies NFS5. A detailed Bill of Materials demonstrating total system cost and affordability can be found in the Appendix. Finally, documentation of the products use and safety considerations was prepared and can be found under Product Documentation in the Appendix, achieving NFS2.

5.2 Use of Advanced Knowledge

This project relies heavily on physics, acoustics, wave propagation, and digital signal processing algorithms. There is a significant level of hardware and software design and integration required, and most of the work draws from core engineering fundamentals and builds on it with technical papers and research. The circuit design draws from concepts taught in ECE 373, 375, 340, 463, and 432. Communications and signal processing concepts used across software and hardware (e.g. the Fourier transform and Nyquist's theorem) were taught in ECE 318, ECE 380, and ECE 207. Embedded programming used on the software side was taught hands-on in ECE 298, ECE 350, ECE 455, and ECE423.

5.3 Creativity, Novelty, Elegance

The novelty and elegance of Quietr lies in its simplicity. Bose, Apple, and Sony offer competing noise cancelling headphones. However, these products must all be worn in or over the ears, and are not necessarily suitable for sleeping. There are concerns with comfort and battery life with these devices. On the other side of the spectrum, providing noise cancelling throughout an entire room presents a number of difficulties. For example, one would have to track the user's position throughout the room and arrange an array of devices to track their position throughout the room. This would be expensive and complex. By simplifying the problem - most of the time people spend in their rooms is spent sleeping, when they are naturally stationary - Quietr becomes cheaper and more convenient than either option. In addition, Quietr is theoretically more robust than existing competitors. Noise cancelling headphones are restricted in the frequencies they can cancel since the reference microphone and error microphone are at the same position - the system has very little time to do the processing required for higher frequencies. Quietr can in theory perform 'look-ahead' since the sensors are so far apart, and because a computer can work much faster than the speed of sound. Therefore, more frequencies can be attenuated than with noise-cancelling headphones.

5.4 Quality of Risk Assessment

Three risks were identified in our assessment document - high power consumption, perceivable latency, and insufficient skill/knowledge. Of all of these, the one that we identified as highest probability was the last one. Unfortunately, the selected board did not have functioning audio libraries. Typically, vendors provide a Board Support Package (BSP) to abstract away difficult-to-use Hardware Abstraction Layer (HAL) functions. The audio IO BSPs did not work and many man-hours were spent writing our own HAL-level code for audio. This problem was exacerbated by our lack of expertise, but was ultimately caused by the vendor's inability to provide functioning utilities.

Another issue arose, which was that of audio interfaces. We selected the board (the STM32H735-DK) due to its ability to handle many audio devices, but unfortunately, it only had a few usable audio interfaces that multiplexed external audio peripherals. This prevented us from connecting and using three transducers (two microphones and a speaker), and due to the chip shortage, we could not source a replacement board.

Ultimately, the greatest problem our project faced was the board itself. While it initially seemed attractive, faulty libraries and board design presented many roadblocks and prevented our system from reaching its highest potential.

5.5 Student Workload

Name	Student Number	Workload
Yoav Arbiv	[REDACTED]	20%
Matthew Nielsen	[REDACTED]	20%
Emma Blatt	[REDACTED]	20%
Hanna Muratovic	[REDACTED]	20%
Fey Asan	[REDACTED]	20%
Total		100%

Table 7: Student hours.

References

- [1] D. Abbott. [Online]. Available: <https://sound.pressbooks.com/chapter/frequency-and-loudness-perception/>
- [2] H. R. Colten and B. M. Altevogt. Institute of Medicine, 2006.
- [3] R. Pujol, “Journey into the world of hearing - specialists,” Jun 2018. [Online]. Available: <http://www.cochlea.org/en/hear/human-auditory-range>
- [4] “Microphones.” [Online]. Available: <https://www.digikey.ca/en/products/filter/microphones/158>
- [5] J. Lewis, “Understanding microphone sensitivity,” May 2012. [Online]. Available: <https://www.analog.com/en/analog-dialogue/articles/understanding-microphone-sensitivity.html>
- [6] “Battery selection guide,” Nov 2020. [Online]. Available: <https://www.saftbatteries.com/products-solutions/products/ls-lsh-lsp?text=ls&tech=&market=&sort=newest&submit=Search>
- [7] A. Lefton, “Solved! the standard ceiling height for homes,” Dec 2019. [Online]. Available: <https://www.bobvila.com/articles/standard-ceiling-height/>
- [8] R. Murphy. [Online]. Available: <https://www.ramelectronics.net/Woofer-Tweeters-Crossovers.aspx>
- [9] “Silentium’s s-cube™ development kit.” [Online]. Available: https://www.acousticals.org/Silentium%20%99s-S-Cube%20%84%A2-Development-Kit-ANC-Active-Noise-Control-p_197.html
- [10] C. D. Bruce Rose, “Mems versus ecm: Comparing microphone technologies,” Feb 2019. [Online]. Available: <https://www.digikey.com/en/articles/mems-vs-ecm-comparing-microphone-technologies>
- [11] B. Rose, “Analog or digital: How to choose the right mems microphone interface,” Feb 2019. [Online]. Available: <https://www.cuidevices.com/blog/analog-or-digital-how-to-choose-the-right-mems-microphone-interface#analog-mems-microphone-interfaces>
- [12] “Interfacing pdm digital microphones using stm32 mcus and mpus,” July 2019. [Online]. Available: https://www.st.com/resource/en/application_note/an5027-interfacing-pdm-digital-microphones-using-stm32-mcus-and-mpus-stmicroelectronics.pdf
- [13] “As07708ps-2-r 77mm speaker datasheet.” [Online]. Available: <https://www.puiaudio.com/media/SpecSheet/AS07708PS-2-R.pdf>
- [14] “Stm32h735xg datasheet.” [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32h735ag.pdf>
- [15] “Dmm-4026-b-i2s-r.” [Online]. Available: <https://www.puiaudio.com/media/SpecSheet/DMM-4026-B-I2S-R.pdf>
- [16] “Calculations with loudspeakers.” [Online]. Available: <https://www.tao.eu/service/soundcheck/calculations-with-loudspeakers/>

- [17] “Comparative examples of noise levels,” Mar 2021. [Online]. Available: <https://www.iacoustics.com/blog-full/comparative-examples-of-noise-levels.html>
- [18] E. Gaalaas, “Class d audio amplifiers: What, why, and how,” Jun 2006. [Online]. Available: <https://www.analog.com/en/analog-dialogue/articles/class-d-audio-amplifiers.html>
- [19] DigiKey, “Stm32h735g-dk.” [Online]. Available: https://www.digikey.ca/en/products/detail/stmicroelectronics/STM32H735G-DK/13171216?utm_adgroup=General&utm_source=google&utm_medium=cpc&utm_campaign=Smart+Shopping_Product_Zombie+SKU&utm_term=&productid=13171216&gclid=Cj0KCQjw29CRBhCUARIsAOboZbIW5RT_eyqKgt8BzydlBvphZ0l5m4HNsWGshZLfqzRG45wRyYQZ0TUaAmCSEALw_wcB
- [20] “Vel05 series datasheet.” [Online]. Available: https://www.xppower.com/portals/0/pdfs/SF_VEL05.pdf
- [21] Nitecore, “Nitecore nl1834 18650 3400mah 3.7v protected lithium ion (li-ion) button top battery.” [Online]. Available: <https://www.batteryjunction.com/nitecore-nl189.html>
- [22] FlashLightWorld Canada, “Nitecore nl1834 18650 3400mah 3.7v protected lithium ion (li-ion) button top battery.” [Online]. Available: <https://www.flashlightworld.ca/products/nitecore-nl1834-18650-3400mah-3-7v-protected-lithium-ion-li-ion-button-top-battery>
- [23] Amazon, “Junction box, zulkit project box ip65 waterproof dustproof abs plastic electrical boxes electronic enclosure black 9.1 x 5.9 x 3.4 inch(230 x 150 x 85 mm)(pack of 1) : Amazon.ca: Tools home improvement.” [Online]. Available: https://www.amazon.ca/Zulkit-Waterproof-Dustproof-Electrical-Electronic/dp/B09JYP3BDD?pd_rd_w=GT9tp&pf_rd_p=929e906d-28ce-409b-9c33-ee482b3f4cc9&pf_rd_r=21D12DAR7JXYZVHQ1CK6&pd_rd_r=1f5f0261-4572-4666-8c46-4f3a44aa3e0d&pd_rd_wg=m9cj&pd_rd_i=B09JYP3BDD&ref_=pd_bap_d_rp_1_i&th=1
- [24] DigiKey, “Cds-40304.” [Online]. Available: <https://www.digikey.ca/en/products/detail/cui-devices/CDS-40304/5355547>
- [25] ——, “Dmm-4026-b-i2s-r.” [Online]. Available: <https://www.digikey.ca/en/products/detail/pui-audio-inc/DMM-4026-B-I2S-R/11587483>
- [26] ——, “Rb2f42c102r-147.” [Online]. Available: <https://www.digikey.ca/en/products/detail/e-switch/RB2F42C102R-147/3778105>
- [27] “What’s the difference between us letter and a4 paper sheets?” June 2020. [Online]. Available: <https://uk.onlinelabels.com/articles/difference-between-us-letter-a4-paper-sheets>

A Appendix

A.1 Bill of Materials

Component	Part Name	Qty	Unit Price
STM32 Discovery Kit	STM32H735G-DK	1	\$119.83 [19]
PCB	Custom	1	\$106.94
Battery	Nitecore NL1834 18650 3400mAh 3.7V Protected Lithium Ion Battery	2	\$29.61 [22]
Enclosure	Junction Box, Zulkit Project Box IP65	1	\$24.99 [23]
Speaker	CDS-40304	1	\$8.12 [24]
Microphone	DMM-4026-B-I2S-R	1	\$3.91 [25]
Switch	CDS-40304	1	\$2.75 [26]
Total Cost			\$325.76

Table 8: Bill of Materials

A.2 Product Documentation

A.2.1 Product Information

- Quietr is an active noise control system that can be easily installed in a room to lessen the impact of incoming ambient noise for one target during sleep.
- Quietr possesses the capability to reduce perceived noise by an average of 5 dB at the target for noise in the frequency range of 300 Hz - 1 kHz.
- A sleep mode feature is available to reduce power consumption. If a window of consecutive noise samples is below a decibel threshold, Quietr enters sleep mode and stops active noise reduction until noise levels rise above the threshold.
- The device runs on 3400 mAh rechargeable lithium-ion batteries, with an expected battery life of at least 8 hours.

A.2.2 User Manual

- Toggle the blue switch located on the side of the enclosure to turn Quietr ON/OFF.
- Press the reset button located on the top of the enclosure (and next to the touch screen) to initiate calibration routines.
- Turn device off before transportation. Once moved, the device must be re-calibrated to ensure proper operation.

A.2.3 Safety Considerations

- Keep an eye on the wires to ensure they are always in good condition. Stop using Quietr if the wires start to fray or melt.
- If you notice Quietr getting hot, discontinue use immediately. Keep cables away from any sources of heat.
- Turn Quietr off when not in use.
- Do not remove battery from device when in use.

A.3 Device Footprint



Figure 20: Dimensions of selected enclosure
[23]

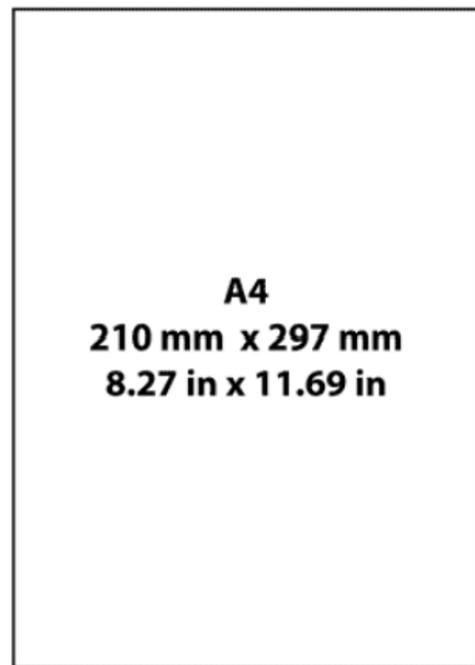


Figure 21: Dimensions of an A4 sheet
[27]