

A Power-Efficient Audio Acquisition System for Smart City Applications

Evan Fallis^a, Petros Spachos^{a,*}, Stefano Gregori^a

^a*School of Engineering, University of Guelph, Guelph, Ontario, Canada*

Abstract

Acoustic noise has adverse effects on human activities. Aside from hearing impairment and stress-related illnesses, it can also interfere with spoken communication, reduce human performance and affect the quality of life. As urbanization is intensifying, the potential benefits of reducing noise pollution in smart-city environments are extensive. Noise levels can be collected and analyzed using a wireless sensor network which can monitor the noise level by using microphones. However, every wireless system struggles in terms of the battery requirements needed for continuous data collection and monitoring. In this paper, the design of a testbed for a smart microphone system is presented. To save power, a microcontroller and an Analog-to-Digital Converter (ADC) dynamically switch between high and low power modes in response to environmental noise. Specifically, the high powered components are triggered by a spike in the acoustic noise level. Three wireless technologies, WiFi (2.4 GHz), Bluetooth Low Energy (BLE) 4.0 and Zigbee were examined. According to the results, the power consumption of a node can be lowered by 97% when idle based on the testbed.

Keywords: Low-power audio; audio processing; audio coherence; audio classification, noise pollution.

1. Introduction

A common problem in urban cities is acoustic noise pollution [1]. Noise pollution is an abundance of noise in a concentrated area that can harm humans both physically and mentally [2]. To alleviate this problem, the first step is to understand and analyze what acoustic noise citizens are being exposed to daily. Smart cities enable the introduction of new technology to the public through different services that optimize resource usage, transportation, utilities and more. In a smart city, large amounts of data are collected and utilized [3]. This is done to predict, analyze, and observe trends that happen throughout the city based on various sensors.

Audio data is an example of sensory information being collected. The amount of human comfort in different locations within a city can be found by analyzing noise levels collected by multiple microphones. By creating a tool that can visualize the levels of acoustic noise around the city, its people are given the ability to react and adjust accordingly. In future iterations, it may also be possible to show the type of sounds present if a neural network is utilized [4]. An example is the classification

of different sounds such as human voice, traffic, construction, or air transport that is currently present in any school or hospital. Mapping the amount of sound classified as human voice within an area grants the ability of the public to observe the amount of human presence at the time. This can also help citizens to determine if an area within the city such as a park or restaurant is comfortable to visit at a specific time in the day.

Analyzing the audio data in an urban environment can become complicated depending on the location within the city [5]. There is a variety of challenges such as privacy and placement [6]. Privacy has always been a big issue with regard to the collection of data in a public setting. An invasion of privacy could be claimed by anyone who feels uncomfortable about the idea of the voices being recorded.

Another challenge is the reliability of the data, which depends on its accuracy in modeling the noise within an area. Data is considered reliable if the extraneous variables that could skew the noise model of the area are minimal. A major concern is the power consumption of various microphones and analog-to-digital converters (ADCs) when collecting audio data [7]. This is especially true when the microphone is set to sample throughout the day, meaning that it will have minimum downtime. Many systems try to solve this problem by

*Corresponding author

Email address: petros@uoguelph.ca (Petros Spachos)

sampling at different points in the day rather than being continuous. Although this is a good method for saving energy, it can miss key information due to the scheduled sampling.

The work presented relates to a framework for reducing power consumption in Micro-Electro-Mechanical System (MEMS) microphone ADCs for an acoustic environmental monitoring system. A testbed to prove the power savings has been developed. The contributions of this paper are as follows:

- A testbed for environmental sound detection using a microphone was designed. The microphone was selected after experimentation on the energy requirements and the accuracy of four different microphones.
- An amplifier and a threshold circuit were designed for use with audio collection systems.
- Three wireless technologies were used to examine the effectiveness of the proposed energy-saving scheme. Specifically, WiFi (2.4 GHz), Bluetooth Low Energy (BLE) 4.0 and Zigbee were used.

The results are compared to a reference system to prove the viability and demonstrate the improvements. The rest of the paper is organized as follows; Section 2 reviews the related works and Section 3 discusses the proposed system architecture. Section 4 details the procedure involved and the results found during analysis are presented in Section 5. The conclusion is in Section 6.

2. Related Work

In recent years, many research works involve wireless sensor networks for acoustic noise collection [8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. Their main focus is the development of systems that have low enough power to reliably operate with little to no human intervention.

A promising implementation has been developed in [8]. The work focuses on developing an ultra low power sensor node that is designed to wake up periodically to collect data, it was programmed to wake for ten minutes every hour. To accomplish this, an ultra-low-power microcontroller was used along with an XBee. The total power when transmitting was 810 mW while the transmitting power was 132 mW. A handheld Sound Pressure Level (SPL) meter was used to verify the accuracy of the audio level collection. Although this prototype may be sufficient to get the overall average noise

level during a day, it can easily miss many important events since it is only awake less than 20% of the time.

The work in [9] presents a node that collects audio level information as well as other types of data. It is designed to wake up based on a timer interrupt or when the accelerometer is triggered. The method of wireless transmission involves the Zigbee protocol to ensure low power. The model was verified by measuring the audio level of varying sinusoidal waves, ranging from 31.5 Hz to 16 kHz. This was recorded with an MI 6201 MULTINORM sound level meter. The average power consumption of the prototype is 45 mW. This is extremely low when compared to most devices, allowing operation for up to 7 days. This prototype is a good starting point but requires a longer operating time to reduce the need for continuous maintenance.

The work in [10] features a prototype using a mini PC. Although the system is not fully wireless, it is a good solution for indoor environments. It was verified using a calibrated sound level meter using an A-weighted filter. In [11], a prototype was built for use with city buses. This had a power supply unit connected to the city bus and used 3G as well as GPS to wirelessly transmit data along with a location stamp. As described in the article, one challenge faced was attempting to filter out the noise generated from the bus, this presents a big problem with the accuracy of the data if it is meant to represent noise level from a human perspective.

A pilot study was presented in [12], which involved an embedded system to collect acoustic noise. The power consumption was not specified but the accuracy of collected data was discussed extensively. A standard sound level meter was used to calibrate the system and verify the analog filters. A noise monitoring node based around a Raspberry Pi 2B and USB microphone was developed in [13]. A sound level meter was used to verify all measurements and power over Ethernet supplied each node with energy. Although not wireless, it was still considered a flexible solution for an acoustic sensor network.

In [14], a low-cost sensor network was developed to target the problem of noise pollution. It has an operating power of 89 mW when collecting data. A sound level meter was used to calibrate the system, ensuring accurate measurements. Despite the low power, the prototype can run for less than 4 days, which is not very realistic for a full implementation. A similar implementation was followed in [15]. The authors in [16] show preliminary results of a wireless noise collection system, mainly discussing the accuracy of this method and giving recommendations for future designs. In [17], the authors focus on a prototype involving low power lis-

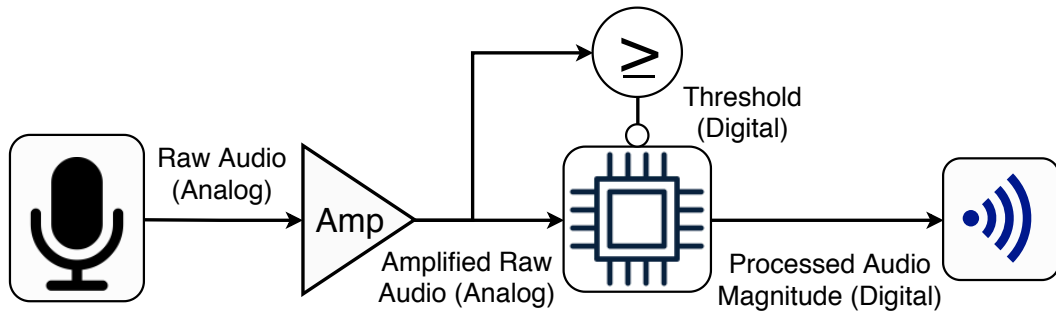


Figure 1: System overview.

tening. They mainly focus on the networking protocols involved and recommend the collection tree protocol.

Smartphones have been considered for noise collection via crowdsourcing. The authors in [18] developed an Android application for the use of acoustic data collection. It gives the user two collection methods, manual mode or automatic collection. Both require no extra hardware for the user to start collecting the acoustic noise level around a city. The exact power being used by the application is unknown. To verify the model, a Sauter SU-130 sound level meter was used and compared with the data collected via the app. A visual dashboard is available to the end-user which displays the collected data on to a map. Another Android application was developed in [19]. This application allows the user to record and visualize the acoustic comfort in an area as well as push the results to a server for analysis from the authors. The accuracy of their data was analyzed and verified in a separate study [20].

The main focus of the papers reviewed so far have been focused on developing a low power solution for ambient sound collection. The other concern this paper attempts to address is finding an effective way to collect and handle the sound data while being able to integrate it into the design. The authors in [21] discuss a mobile system dedicated to the sensing of noise pollution in urban environments. It focuses on determining the level of attenuation as part of the sound collection process in addition to gathering the actual audio levels. The work in [22] specializes in the tracking of audio data with an emphasis on vehicles. The data being sent is based on a certain threshold value which will omit any data below it. Unfortunately, no schematic of the system was made and the power consumption was not presented. This severely limits the usefulness of the claimed results and methodologies.

In [27], a factory was examined and profiled based on the acoustic levels. Their goal is to reduce the amount of

noise pollution by redesigning the air compressor. The work presented in [28] highlights an improved way to collect audio data for voice activity detection. It uses both audio and visual signals with a supervised learning algorithm to detect what audio frames correspond to human voice. The authors in [29] use a Gaussian mixture model-based classification approach to speech detection. It's trained using spectral flow direction, a novel feature for this type of classification. In [30], a multistage system is developed to analyze environmental sound and classify it. The authors claim to achieve an accuracy higher than 90% for audio concept identification.

In comparison with these works, this paper introduces a novel method that is portable and can be implemented into many in-place solutions. Specifically, this testbed utilizes a wake-up circuit to conserve power when minimal audio activity is present, while three different technologies are examined to find the most energy-efficient.

3. System Architecture

The introduced system has four main components: the microphone, the circuit, the microcontroller, and the wireless unit. An overview of the system can be seen in Fig. 1. A microphone collects raw audio data and then the signal is amplified. This amplified signal is then fed to both a microcontroller and a custom threshold circuit. The microcontroller only collects data when the audio threshold has been broken, then finally sends the processed data to a server via a wireless technology.

3.1. Microphone

Multiple microphones were considered to determine a good candidate which had a high ratio of accuracy to power. They all have varying manufacturers, voltage ranges, and configurations. A summary showing the details of each microphone can be found in Table 1. While





Manufacturer	Microphone	Voltage Range (V)	Configuration
InvenSense	 ICS-40310 [23]	0.9 - 1.3	Analog
Projections Unlimited, Inc. (PUI)	 PMM-3738-VM1010-EB-R [24]	1.6 - 3.6	Analog
STMicroelectronics (ST)	 MP23ABS1 [25]	1.52 - 3.6	Analog
STMicroelectronics (ST)	 MP34DT05-A [26]	1.6 - 3.6	Digital

Table 1: MEMS microphone overview.

most are analog, one was picked to be digital and used as a control to compare with. The integrated ADC will lead to a higher accuracy than others, the disadvantage being a significant increase in power.

The chosen microphone needed to be analog to properly interface with the threshold circuit. To make the system more sensitive to audio, an amplifier was used.

3.2. Circuit

The circuit that was designed has three main components: the amplifier, the envelope filter, and the threshold circuit.

3.2.1. Top level design

Multiple circuits were designed to collect and analyze the data. An overview of the electrical components can be seen in Fig. 2. This shows the three simplified components. An external amplifier was needed to allow the ADC to accurately capture small changes in signal amplitude, essentially increasing the resolution. The envelope filter averaged the positive amplitude using resistors and a capacitor. This is used to bypass the need for pre-processing before data transmission. The main feature of this prototype, the wake-up circuit, was designed to output a high or low value based on a threshold.

3.2.2. Amplifier

An amplifier was needed to make the signal recognizable to an ADC with a standard resolution. This was also important for the wake-up circuit, making it much more sensitive to changes in noise. The amplifier was

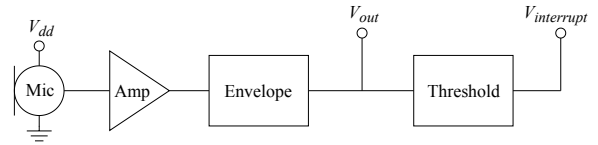


Figure 2: Top level view of analog circuit.

designed to have low noise and a high gain. The amplifier is shown in Fig. 3 The negative feedback adds stability to make a much cleaner signal. The gain of the amplifier can be determined from R_f and R_1 as seen in the following:

$$A_v = 1 + \frac{R_f}{R_1} \quad (1)$$

where A_v is the gain of the amplifier. This is the transfer function for a typical non-inverting amplifier design. The common-mode input and output can be seen by the following equation:

$$CM = \frac{V_{dd}}{2} \quad (2)$$

where the common-mode voltage is simply the supply voltage divided by two. This is from the voltage divider defined by R_2 .

3.2.3. Envelope

An envelope circuit was designed to average the magnitude of audio. The schematic of this circuit can

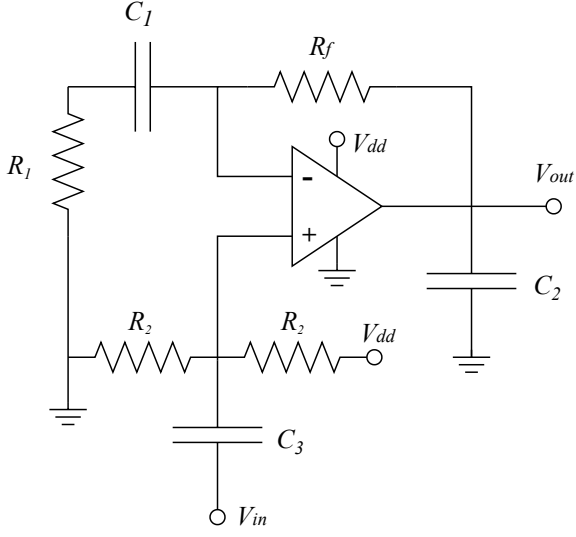


Figure 3: Amplifier circuit.

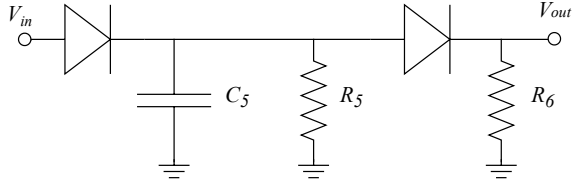


Figure 4: Envelope circuit.

be seen in Fig. 4. This surpassed the need for pre-processing. The purpose of C_5 and R_5 is to increase the time constant, filtering the output. The second diode and R_6 are used to lower the common-mode voltage since the XBee ADC uses a voltage reference of 1.2 V.

3.2.4. Threshold circuit

The threshold circuit was used to control an interrupt on the microcontroller. This disabled sleep mode and allowed the microcontroller to send audio data through the various wireless technologies for a brief period before going back into sleep mode. The schematic for the wake-up circuit can be seen in Fig. 5.

Two transistors are used, M_1 regulates the DC offset of the input, M_2 is biased in such a way that it will saturate when the input is high. Once M_2 is saturated, it ties the output to ground. Alternatively, V_{out} will be close to V_{dd} when M_2 is in the cut-off region. This means that the output will be grounded when the threshold is broken and V_{dd} when no significant noise is heard. At this time, the threshold can only be configured through hardware but may be configurable in future iterations.

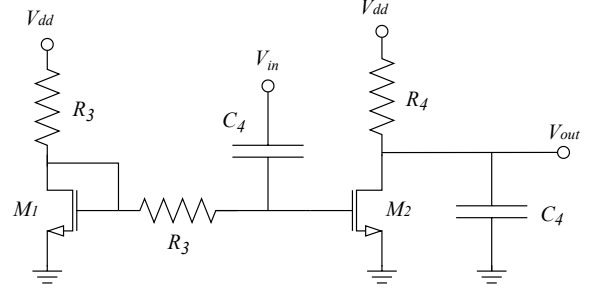


Figure 5: Threshold circuit.

Parameter	Value
V_{dd}	3.3 V
R_f	100 k Ω
R_1	1 k Ω
R_2	1 M Ω
R_3	10 M Ω
R_4	500 k Ω
R_5	10 M Ω
R_6	100 k Ω
C_1	22 μ F
C_2	100 nF
C_3	4.5 μ F
C_4	1.5 nF
C_5	9 μ F

Table 2: Circuit parameters.

All of the circuit parameters used for this set of experiments can be found in Table 2. Using Eq. (1), the gain can be calculated based on these parameters:

$$A_v = 1 + \frac{100k\Omega}{1k\Omega} = 101 \frac{V}{V} \quad (3)$$

The gain becomes $101 \frac{V}{V}$, amplifying the signal by approximately 20 dB. This is necessary for the ADC to properly recognize relatively small changes in the ambient acoustic noise.

3.3. Microcontroller

A microcontroller was needed to correctly process the analog data incoming from the amplifier. The

NodeMCU was picked for its low power, flexibility, and ability to harness many types of wireless technologies. The NodeMCU uses an ESP8266 microprocessor with a 10-bit ADC.

When the microcontroller was not in sleep mode, it connected to the server using either Wi-Fi (2.4 GHz), BLE 4.0, or Zigbee. The microcontroller then slept if the noise threshold was not broken. The XBee S2C is capable of converting an analog value to digital and transmitting the data using Zigbee. This bypasses the need for a microcontroller and will be tested alongside the microcontroller variant of the prototype.

3.4. Wireless technology

Three wireless technologies were tested to determine which was the lowest power, and show the sleep functionality in a variety of cases.

3.4.1. Wi-Fi

Wi-Fi is a common technology found in most homes. It allows users to connect to the internet wirelessly by using an access point. The power consumption and bandwidth are typically high, keeping up with the needs of the consumer [31]. For the application of an audio acquisition system, the bandwidth should not be as much of an issue when compared to the power being expended.

The NodeMCU has an integrated proprietary Wi-Fi (2.4 GHz) chip, giving it the capability to transmit data. It was able to directly send data to a server.

3.4.2. Bluetooth

Bluetooth is a popular technology used for many wireless devices. It is designed to be a low-power solution that conserves energy to promote battery life in electronics. The power is typically less than Wi-Fi, but the throughput suffers [31].

The DSD TECH HM-10 was used to connect the microcontroller using BLE 4.0 [32]. It communicates with a separate device to wirelessly transmit the data.

3.4.3. Zigbee

Zigbee is a lesser-known protocol capable of low power transmission, meant to be used specifically for IoT (Internet of Things) [33]. This makes it a good match for the application of low power collection of audio.

The XBee S2C module was being used [34]. This provided a platform for the microcontroller to transmit using the Zigbee protocol. Similar to Bluetooth, the Zigbee experiments required a secondary unit with a matching XBee to provide wireless communication.

An advantage of the XBee is the integrated ADC, which allows for wireless transmission without the need for an external microcontroller.

4. Experimental Procedure

4.1. Microphone evaluation

To examine which microphone was the most accurate, the coherence between the original waveform and the recorded waveform was found. This shows how similar the recording is to the original signal in terms of the frequency response. Magnitude squared coherence was used on the pre-processed signals to create an output which shows how much the frequency domains of the signals match.

The coherence was conducted in Matlab by using functions in the signal processing library. A procedure was used to filter the two signals and end up with a percentage value that represents how closely the recording resembles the source. The simple list of steps includes:

- Down-sample all recordings to 8 kHz.
- Lineup source and recording.
- Limit time to exactly 80 seconds.
- Find magnitude squared coherence.
- Take envelope of the signal.
- Average the envelope.

The purpose of down-sampling is to simplify the data and give every dataset a fair comparison without removing the frequencies that matter as well as increase the performance of the script. This uses decimation to make the signal simpler, and thus easier to compare.

Although the magnitude squared coherence will convert the signals to the frequency domain, it is still important to properly line up the source and recording. This eliminates any frequencies that happened before or after the recording interval. The offset can be found by comparing the similarities of the two signals and performing a correlation that is much more basic when compared to the magnitude squared coherence. This offset was used to perfectly line up the signals by trimming the microphone clip to exactly 80 seconds. The offset was based on a 10 second clip of the source and recording which was shown to be enough to properly line up the signals. The magnitude squared coherence finds the cross power spectral density in the numerator which can be seen below:

Algorithm 1 : Find the effective coherence value

```
y_source = resample(y_source, 8000, Fs_source)
y_mic = resample(y_mic, 8000, Fs_mic)

y_source_clip = y_source(1:8000*10)
y_mic_clip = y_mic(1:8000*10)

mic_delay = finddelay(y_source_clip, y_mic_clip)
y_mic = y_mic(mic_delay:mic_delay + 8000*80 - 1)

[corr, f] = mscohere(y_source_clip, y_mic_clip, [], [], [], 8000)

corr = envelope(corr, 100, 'peak')

corr_value = mean(corr)
```

$$C_{xy}(f) = \frac{|G_{xy}(f)|^2}{G_{xx}(f)G_{yy}(f)} \quad (4)$$

where G_{xy} is the cross-spectral density defined by:

$$G_{xy}(f) = \int_{-\infty}^{\infty} R_{xy}(t)e^{-j\omega t} dt \quad (5)$$

where R_{xy} is the cross-correlation of the two original signals, $x(t)$ and $y(t)$. The cross-correlation of the two signals can be described as:

$$R_{xy}(t) = [\overline{x(-t) * y(t)}](t) \quad (6)$$

which involves the two signals being convoluted.

The envelope of this attempts to create a coherence that more accurately represents how closely the two signals match, this finds the outline of the signal. Finally, the mean is taken of the resulting envelope to determine the accuracy value. The code showing all the modifications to the signal can be seen in Algorithm 1.

The experimental conditions were kept constant through all the experiments. The list of conditions for the experiment include:

- 80 second audio clip of various urban sounds.
- Nexus 5 at 53.3% volume, connected to speakers.
- Audio clips out of speakers are 96 kHz.
- Distance of 30 cm from speaker and microphone.

A Nexus 5 was used to play the 80 seconds clip for each experiment. This amount of time was chosen to get an average accuracy for each microphone, reducing the effect of any outliers. The Nexus 5 was connected

to a pair of Nex Tech analog speakers to provide a much higher quality sound when compared to the embedded speakers on the Nexus 5 [35]. This was kept at a constant 30 cm away from the respective microphone which ensured consistent data. Each microphone was pointed directly at the speakers for a fair comparison. The volume of the Nexus 5 was set to a constant of $\frac{8}{15}$ bars, or 53.3% while the speakers were kept at a maximum gain.

The Urban Sound dataset was used to test with, it contains audio categorized into four main sections [36]. This includes human, nature, music, and mechanical based sounds. Twenty seconds of each category was used for testing, totaling 80 seconds for each data point. The dataset features multiple sampling rates, channels, and resolution. To normalize this data, all clips were up-sampled to 96 kHz and each clip was limited to one channel, also known as mono. This was done with the Sox Linux tool by utilizing linear interpolation to reduce any inaccuracies during the coherence process. Down-sampling has the potential to remove data points so it was excluded as an option.

4.2. Power evaluation

It is important to measure the power savings created by the wake-up functionality. To do this, the Monsoon power monitor was being used. It is capable of providing very accurate measurements of energy consumption [37].

Measuring the power consumption of both the sleep and active modes of each permutation of the prototype will show

The Urban Sounds data set was being used for testing. Audio clips were created and designed to simulate a realistic outdoor environment. Specifically, 20 seconds of audio from a category was played, followed by 100 seconds of silence. This was repeated for each category, totaling 8 minutes of audio. The clips were organized by loudness to test the power consumption at different noise levels.

4.3. Calibration

The voltage being recorded by the microcontroller ADC does not mean much until it can be related to a more accurate measuring device. Putting data into a human-readable format is important both for interpretation and comparison to other data sets.

A handheld Sound Pressure Level (SPL) meter was used to calibrate the prototype. An equation was formed to convert values on the ADC to one representing decibel noise levels. The Decibel X Android app was also calibrated and used to measure the average noise level

Microphone	Power (mW)	Accuracy
InvenSense	0.0079	0.7359
PUI	0.3545	0.6966
ST (Analog)	0.3480	0.7187
ST (Digital)	2.0813	0.7577

Table 3: Microphone accuracy and power.

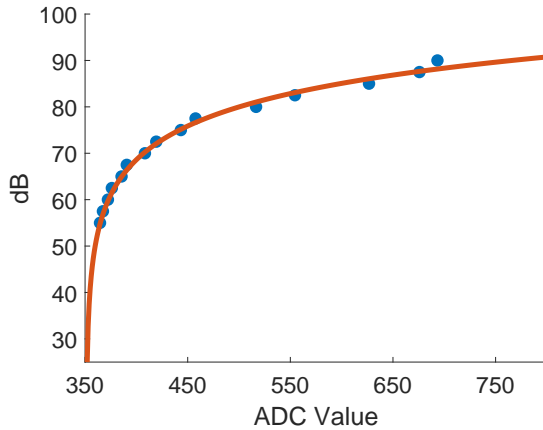


Figure 6: SPL calibration curve.

when recording and testing with the prototype. The application was seen to only have an error of approximately ± 2 dB.

5. Results

5.1. Microphone selection

After the magnitude-squared coherence was conducted, the accuracy values could be analyzed as well as the power dissipation. As shown in Table 3, it is clear that the InvenSense has the lowest power consumption. This is partially because the supply voltage is 1 V as opposed to all the other microphones which have a nominal voltage of 3.3 V. However, the disadvantage is requiring a voltage regulator to step down from 3.3 V. This would require an excessive amount of power.

$$dB = -290.5(x - 350)^{-0.04258} + 314.7 \quad (7)$$

The highest accuracy is the digital ST microphone. This makes sense as the integrated ADC is configured to work specifically for the microphone. Consequently,

it also has the highest power consumption. Additionally, exclusively using a digital microphone would prohibit the use of the threshold circuit since the output of the microphone is PDM rather than a pure analog signal. The analog ST microphone shows high accuracy and low power, working without the need for a voltage regulator. These reasons are why the analog ST microphone was chosen as the best candidate for the remaining experiments.

5.2. Decibel calibration

A handheld SPL meter was used to calibrate the data being transmitted from the microcontroller. An equation was formed to transfer the voltages being reported to the server into a Z-weighted decibel value. The visual calibration can be found in Fig. 6. The R^2 value is 0.995, indicating a very strong correlation. The equation is as follows:

where x is the ADC value after processing on the microcontroller. This power trend makes sense as decibels are logarithmic, meaning it should follow an exponential trend.

5.3. Power consumption

The power consumption of the prototype was measured with the Monsoon power monitor. Each module was considered individually to analyze the biggest power draw in the system. The power consumption of each component can be seen in Table 4. As expected, the microcontroller and Wi-Fi module have the largest power draw by a very significant amount when transmitting. Both Bluetooth and Zigbee are comparable in terms of power, though the power of the XBee is slightly lower, especially when sleeping. The ESP8266 Wi-Fi module of the NodeMCU was considered to have a power of zero when sleeping since it's integrated into the microcontroller itself.

The power of each wireless technology was specifically compared in Fig. 7a and Fig. 7b. These figures show a comparison of each wireless technology in both the sleep mode and transmitting mode. A clear trend of Wi-Fi consuming significantly more power than the other two technologies can be seen. Zigbee uses the least while BLE is in the middle.

Table 5 puts each technology in perspective in terms of the total power being saved by the sleep functionality of the system. The most apparent savings occur when using Wi-Fi since the active power of the Wi-Fi module is so much greater than the sleeping power. The standalone system consisting of the XBee without a microcontroller has the lowest overall power. The increase

Component	Power (mW) [Transmit]	Power (mW) [Sleep]
Microphone	0.35	0.35
Amplifier	0.20	0.20
Threshold	0.07	0.07
NodeMCU	91.84	16.76
Wi-Fi (2.4 GHz)	265.75	0.00
BLE 4.0	49.02	8.47
Zigbee	33.68	0.07

Table 4: Power results by component.

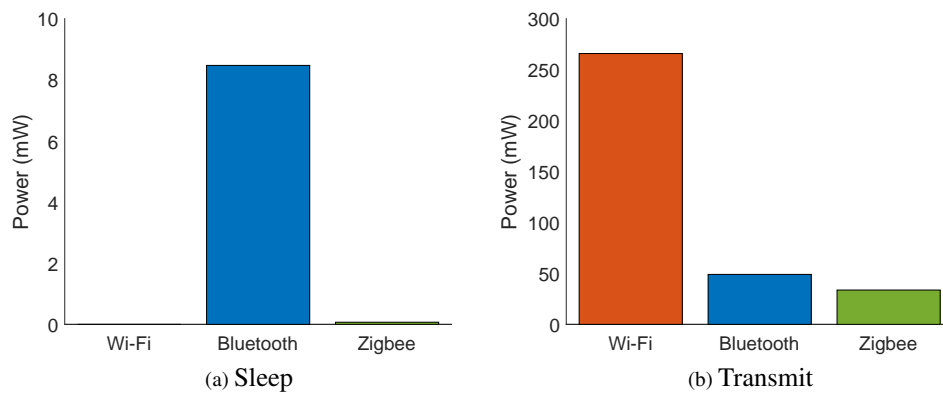


Figure 7: Wireless power consumption.

Wireless Technology	Power (mW) [Sleep]	Power (mW) [Transmit]	Savings (%)
Wi-Fi (2.4 GHz)	16.76	357.59	95.3
BLE 4.0	25.23	140.86	82.1
Zigbee	16.83	160.43	89.5
Zigbee (Standalone)	1.00	34.30	97.1

Table 5: Total power results of prototypes.

in sleeping power is most likely due to extra leakage current from the use of the ADC on the XBee.

5.4. Comparison with similar approach

To prove the relevance of the power savings achieved, an in-place solution will be compared and analyzed. This will be compared to the Zigbee standalone implementation in this paper since it has the lowest overall

power consumption. When looking at the work in [9], the average power consumption claimed is 45 mW. This is based on a periodic interrupt to transition from sleep to active, a common technique used to save power in such systems. If a quiet environment is considered, no significant sound would be reported. In this situation, the average would still be 45 mW in the work being compared while the presented system will use 1 mW.

This leads to a power savings of over 97%. The paper being compared assumes the prototype has a battery size of 2900 mAh which means their prototype can last for 9 days based on the average power consumption. The amount of improvement shown in this prototype leads to the system lasting approximately 400 days when no significant audio is detected.

If a moderately noisy environment is considered, this will change the average power of the presented system. The system presented system will never use more power than the system being compared since a 100% duty cycle would lead to a power of only 34 mW. This means that in the worst case, the prototype will last just over 11 days. Based on this result, the improved version presented in this paper will always outperform that of the comparison work.

Another big advantage of the presented system is that it won't miss significant events, unlike a periodic sampling system. This is because the threshold circuit will always activate the transmission of data once the noise threshold has been reached. The envelope circuit will also mean the sound that triggers the threshold will also be accounted for when reading the sample. Improving the active standard power consumption of the prototype by removing the microcontroller led to a significantly improved power consumption, meaning that in any state it will be more efficient than the comparison system. The implementation of the threshold circuit further improved the power by allowing the prototype to sleep in most cases, thus improving leading to a much more efficient prototype overall.

6. Conclusion

In this work, a wireless ambient sound collection system was designed and tested. The novelty is a threshold circuit that only wakes the digital components upon a threshold being broken. This lowers the average power consumption by 97% when using the lowest power permutation of the prototype is used for data transmission, making a full-scale wireless microphone array much more feasible in smart cities. When compared to a system of similar performance, the presented system lowers power consumption. In comparison to a periodic sleep cycle, this solution ensures that relevant audio data is collected. This can also be tailored to regulate an environment that is meant to stay below a certain level of dB.

The work presented is a flexible solution that can easily be tailored to other audio collection systems. Three wireless technologies were used to both measure the power consumption and find the effectiveness of a

threshold circuit in each. In the future, large-scale wireless audio collection systems will be possible, this is one step towards a power-efficient system capable of operating for months with no human maintenance.

Testing a greater variety of MEMS microphones could also potentially improve accuracy and power. Developing an analog A-weighted filter would make the noise level better represent the human ear when collecting data. Developing a dynamic threshold circuit to regulate the duty cycle would help reduce power further.

References

- [1] J. Khan, M. Ketzler, K. Kakosimos, M. Sørensen, S. S. Jensen, Road traffic air and noise pollution exposure assessment - a review of tools and techniques, *Science of The Total Environment* 634 (2018) 661 – 676. doi:<https://doi.org/10.1016/j.scitotenv.2018.03.374>. URL <http://www.sciencedirect.com/science/article/pii/S0048969718311343>
- [2] D. Shepherd, K. Dirks, D. Welch, D. McBride, J. Landon, The covariance between air pollution annoyance and noise annoyance, and its relationship with health-related quality of life, *International Journal of Environmental Research and Public Health* 13 (8). doi:10.3390/ijerph13080792. URL <https://www.mdpi.com/1660-4601/13/8/792>
- [3] C. Xu, X. Huang, J. Zhu, K. Zhang, Research on the construction of sanya smart tourism city based on internet and big data, in: 2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS), 2018, pp. 125–128. doi:10.1109/ICITBS.2018.00040.
- [4] H. Shu, Y. Song, H. Zhou, Time-frequency performance study on urban sound classification with convolutional neural network, in: TENCON 2018 - 2018 IEEE Region 10 Conference, 2018, pp. 1713–1717. doi:10.1109/TENCON.2018.8650428.
- [5] F. Rong, Audio classification method based on machine learning, in: 2016 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS), 2016, pp. 81–84. doi:10.1109/ICITBS.2016.98.
- [6] D. Christin, Privacy in mobile participatory sensing: Current trends and future challenges, *Journal of Systems and Software* 116 (2016) 57 – 68. doi:<https://doi.org/10.1016/j.jss.2015.03.067>. URL <http://www.sciencedirect.com/science/article/pii/S0164121215000692>
- [7] K. Shehzad, H. Kang, D. Verma, Y. J. Park, K. Lee, Low-power 10-bit sar adc using class-ab type amplifier for iot applications, in: 2017 International SoC Design Conference (ISOC), 2017, pp. 224–225. doi:10.1109/ISOC.2017.8368864.
- [8] C. Peckens, C. Porter, T. Rink, Wireless sensor networks for long-term monitoring of urban noise, *Sensors* 18 (9) (2018) 3161.
- [9] V. Risojević, R. Rozman, R. Pilipović, R. Češnovar, P. Bulić, et al., Accurate indoor sound level measurement on a low-power and low-cost wireless sensor node, *Sensors* 18 (7) (2018) 2351.
- [10] C. Mydlarz, J. Salamon, J. P. Bello, The implementation of low-cost urban acoustic monitoring devices, *Applied Acoustics* 117 (2017) 207 – 218, acoustics in Smart Cities. doi:<https://doi.org/10.1016/j.apacoust.2016.06.010>. URL <http://www.sciencedirect.com/science/article/pii/S0003682X1630158X>

- [11] R. M. Alsina-Pagès, U. Hernandez-Jayo, F. Alias, I. Angulo, Design of a mobile low-cost sensor network using urban buses for real-time ubiquitous noise monitoring, *Sensors* 17 (1). doi:10.3390/s17010057.
URL <https://www.mdpi.com/1424-8220/17/1/57>
- [12] J. Gubbi, S. Marusic, A. S. Rao, Y. W. Law, M. Palaniswami, A pilot study of urban noise monitoring architecture using wireless sensor networks, in: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 1047–1052. doi:10.1109/ICACCI.2013.6637321.
- [13] J. E. Noriega-Linares, J. M. Navarro Ruiz, On the application of the raspberry pi as an advanced acoustic sensor network for noise monitoring, *Electronics* 5 (4). doi:10.3390/electronics5040074.
URL <https://www.mdpi.com/2079-9292/5/4/74>
- [14] I. Hakala, I. Kivela, J. Ihalainen, J. Luomala, C. Gao, Design of low-cost noise measurement sensor network: Sensor function design, in: 2010 First International Conference on Sensor Device Technologies and Applications, 2010, pp. 172–179. doi:10.1109/SENSORDEVICES.2010.39.
- [15] I. Kivela, C. Gao, J. Luomala, J. Ihalainen, I. Hakala, Design of networked low-cost wireless noise measurement sensors (issn 1726-5479), *International Journal on Sensors & Transducers* 9 (2010) 171–190.
- [16] S. Santini, B. Ostermaier, A. Vitaletti, First experiences using wireless sensor networks for noise pollution monitoring, in: Proceedings of the Workshop on Real-world Wireless Sensor Networks, REALWSN '08, ACM, New York, NY, USA, 2008, pp. 61–65. doi:10.1145/1435473.1435490.
URL <http://doi.acm.org/10.1145/1435473.1435490>
- [17] L. Filippini, S. Santini, A. Vitaletti, Data collection in wireless sensor networks for noise pollution monitoring, in: S. E. Nikolettseas, B. S. Chlebus, D. B. Johnson, B. Krishnamachari (Eds.), *Distributed Computing in Sensor Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 492–497.
- [18] M. Zappatore, A. Longo, M. A. Bochicchio, Crowd-sensing our smart cities: A platform for noise monitoring and acoustic urban planning, *Journal of Communications Software and Systems*.
- [19] I. Aspuru, I. García, K. Herranz, A. Santander, Citi-sense: methods and tools for empowering citizens to observe acoustic comfort in outdoor public spaces, *Noise Mapping* 3 (1).
- [20] I. Aspuru, A. Santander, I. García, M. González, A. Bartonova, Citi-sense project: Acoustic comfort in the empowerment initiative about environmental quality of public spaces, in: *Seminario gaa smart cities e nuovi sensori per la misura del rumore ambientale*, Associazione Italiana di Acustica Pisa 17.
- [21] I. Kirillov, V. Bulkin, The mobile system of urban area noise pollution monitoring, in: 2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S T), 2015, pp. 200–203. doi:10.1109/INFOCOMMST.2015.7357312.
- [22] P. Patil, Smart iot based system for vehicle noise and pollution monitoring, in: 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017, pp. 322–326. doi:10.1109/ICOEI.2017.8300941.
- [23] InvenSense Microphone, (retrieved: 2019-04-08).
URL <https://www.invensense.com/wp-content/uploads/2015/02/ICS-40310-datasheet-v1.2.pdf>
- [24] PUI Microphone, (retrieved: 2019-04-08).
URL <http://www.puiaudio.com/pdf/PMM-3738-VM1010-EB-R.pdf>
- [25] ST Analog Microphone, (retrieved: 2019-04-08).
URL <https://www.st.com/resource/en/datasheet/mp23abs1.pdf>
- [26] ST Digital Microphone, (retrieved: 2019-04-08).
URL <https://www.st.com/resource/en/datasheet/mp34dt05-a.pdf>
- [27] J. Zhou, J. Zhao, P. Li, Noise pollution and control measures of rubber factory compressor room, in: 2009 Third International Symposium on Intelligent Information Technology Application Workshops, 2009, pp. 241–242. doi:10.1109/IITAW.2009.35.
- [28] D. Dov, R. Talmon, I. Cohen, Audio-visual voice activity detection using diffusion maps, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23 (4) (2015) 732–745. doi:10.1109/TASLP.2015.2405481.
- [29] Ü. Zubari, E. C. Ozan, B. O. Acar, T. Ciloglu, E. Esen, T. K. Ateş, D. O. Önür, Speech detection on broadcast audio, in: 2010 18th European Signal Processing Conference, 2010, pp. 85–89.
- [30] I. Feki, A. B. Ammar, A. M. Alimi, Audio stream analysis for environmental sound classification, in: 2011 International Conference on Multimedia Computing and Systems, 2011, pp. 1–6. doi:10.1109/ICMCS.2011.5945607.
- [31] E. Fallis, P. Spachos, Power consumption and throughput of wireless communication technologies for smartphones, in: 2018 Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1–4. doi:10.1109/GIIS.2018.8635665.
- [32] Dsd tech hm-10, (retrieved: 2019-06-19).
URL <http://www.dsdtech-global.com/2017/08/hm-10.html>
- [33] Zigbee website, (retrieved: 2019-06-19).
URL <https://www.zigbee.org/>
- [34] Sparkfun xbee s2c, (retrieved: 2019-06-19).
URL <https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf>
- [35] Nex Tech Speakers, (retrieved: 2019-04-15).
URL <https://www.thesource.ca/en-ca/computers--and--tablets/computer-accessories/computer-speakers/nexstech-2-0-stereo-pc-speakers/p/108033507>
- [36] Urban Sound Data Sets, (retrieved: 2019-04-08).
URL <https://urbansounddataset.weebly.com/>
- [37] Monsoon Power Monitor, (retrieved: 2019-01-30).
URL <http://www.msoon.com/LabEquipment/PowerMonitor/>