Developing an Automatic Cough Counting Device for Canines

A Major Qualifying Project Report submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfilment of the requirements for the Degree of Bachelor of Science

Submitted by:

Michael Hellmich

___

Allison McCarthy

___

Larissa Naidoo

___

Project Advisor:
Professor Songbai Ji

Sponsor Liaison:
Doctor Claire Fellman, Tufts University Cummings School of Veterinary Medicine
Doctor Elizabeth Rozanski, Tufts University Cummings School of Veterinary Medicine

Table of Contents

Table of Figures

Table of Tables

Authorship

Abstract	AM
Chapter 1: Introduction	AM, LN, MH
Chapter 2: Literature Review	
2.1 Overview of Coughing in Canines	AM
2.1.1 Medical Ailments that Cause Coughing in Canine	AM
2.1.2 Cough Mechanics	AM
2.2 Sound Frequencies and Pitch	MH
2.3 Relevant Studies on Other Species	
2.3.1 Pilot Study of Objective Cough Monitoring in Infants	LN
2.3.2 Automatic Cough Detector Using a Vibroacoustic Sensor	AM
2.3.3 Current Cough Monitoring Devices	LN
Chapter 3: Project Strategy	
3.1 Client Statement	
3.1.1 Initial Client Statement	LN, AM, MH
3.1.2 Objectives	LN
3.1.3 Constraints	LN, MH, AM
3.1.4 Revised Client Statement	AM
Chapter 4. Design Process	
4.1 Needs Analysis	MH, LN
4.2 Design Requirements	AM, LN
4.3 Key Industry Standards	MH, LN
4.4 Design Process and Final Design Selection	AM, LN
4.4.1 Initial Conceptual Design Components	LN, AM, MH
4.4.2 Alternative Designs	AM, LN, MH
4.4.3 Final Design Components	AM
Chapter 5. Design Verification and Testing	
5.1 Signal Processing Sound Samples to Determine Frequency Threshold	MH, AM, LN
5.2 Testing of Device Using Sound Samples	AM
5.3 Mechanical Testing of Device Casing	AM, LN
5.3.1 Compression Testing of Device Casing	LN, MH
5.3.2 Drop Testing of Device Casing	LN
5.3.3 Water Testing of Device Casing	LN
Chapter 6. Final Design	
6.1 Final Design	AM, LN
6.2 Economics	LN, AM
6.3 Environmental Impact	LN, AM, MH
6.4 Societal Influence	LN, AM
6.5 Ethical Concerns	AM, MH
6.7 Health and Safety Issues	AM
6.8 Manufacturability	AM

Abstract

The purpose of this project was to design and build an automatic cough counting device for small canines. In collaboration with veterinarians from Tufts Cummings School of Veterinary Medicine, we developed a device that uses sound analysis to distinguish coughing from barking, and automatically displays the number of coughs on a digital display. The device is protected by a custom printed casing, which is water resistant, drop proof, and can withstand loads upwards of 120 N. To optimize comfort and mobility, the device is worn in the form of a harness that can be easily interchanged with various sizes. Our test results conclude that the device has a sensitivity and specificity of 85.7% and 88.8%, respectively. This project was successful in its goals of distinguishing coughing from barking, and automatically detecting and displaying the total number of times the dog coughed without interfering in its daily activities.

Chapter 1. Introduction

There is a multitude of medical conditions that result in symptoms pertaining to coughing in canines. It can often be difficult for veterinarians to diagnose the severity of the symptoms, as there is no definitive way to gauge how frequent and detrimental the coughing is. Veterinarians rely on the patient's owners to report how often their dog is coughing, which is often subjective and unreliable. Therefore, there is a need for an accurate, objective, and methodical way to determine the frequency of coughing. At Cummings School of Veterinary Medicine at Tufts University, veterinarians are interested in obtaining a device which can be used to count and track coughing in smaller breeds of dogs. This device will help owners and veterinarians track the healing progress of the dog after medical treatment. Additionally, it can be used to help pharmaceutical companies analyze and test the effectiveness of their drugs on coughing dog patients. We have been tasked with designing a durable, reliable, and easy-to-use device to objectively monitor coughing in small dogs.

A rise in technological advancement has made it possible for a variety of wearable accessories that can monitor human activity. This rise has also created interest in the development of technology for animal use. There are many opportunities for wearable technologies to be used in the medical industry, since often it is difficult for doctors to accurately track a patient's recovery. For example, diagnosing the severity of coughing has been reliant on subjective methods which consist of the patient's perception of the symptoms using cough visual analogue scores, patient's diaries and questionnaires [1]. Many practitioners have adopted non-pharmacological interventions to help patients suffering from refractory-chronic cough. This is also partly because many intrusive cough medications in the market are not effective, and they contain active antibiotics and ingredients which have no effect on cough symptoms. Non-pharmacological treatments consist of therapy, education, counseling and cough suppression techniques [1]. Many articles point to the fact that non-pharmacological interventions can help in improving the patient's quality of life, as well as significantly decreasing cough reflex sensitivity. However, there is no concrete evidence, study, or review on the efficacy of these treatments [1]. Humans are not the only patients who suffer with difficulties treating coughing symptoms. The veterinary medicine industry has interest in utilizing technological advancements to progress their therapeutic methods. In canines, the implementation of a cough monitoring device is in high demand. There are currently no methods for veterinarians to accurately analyze the effectiveness of therapeutic regimens on chronic cough. Alternatively, they must rely on observations and remarks from the patient's owner. These observations are often unreliable and inconsistent. It is important that veterinarians have a reliable method for observing the severity and frequency of coughing in the patient, in order to provide adequate treatment.

The purpose of this project was to create a device that automatically detects coughing in canines, and determines how frequently the patient is coughing in a given period of time. The device needed to be portable, durable, and small enough in size to comfortably fit a 6-8 lb dog. After receiving medical or pharmaceutical treatment for coughing symptoms, it is intended that the dog wear the harness for a period of time determined by the veterinarian. This enables the veterinarian to collect quantitative data, and accurately track the dog's healing progress. Because coughing is a common symptom among many ailments, this device has many applications. There are different types of coughing which are caused by a variety of medical reasons, including Tracheobronchitis (kennel cough), sore throat, foreign body object, pulmonary problems, heart disease, and tracheal collapse [2]. To properly diagnose and treat coughing patients, veterinarians must understand the severity of the symptoms. Furthermore, for veterinarians to determine how effective a certain medication is, they need a method for tracking the animal's progress. Without having a technique to quantitatively measure the healing progress, it is difficult to determine the impact of the prescribed medicine being used to treat the patient. For many medical ailments, it is crucial that the patient's coughing is carefully monitored and controlled. For example, stent implants, which are used to treat dogs with a tracheal collapse, are small tubes which force a trachea open to enable proper breathing. If a dog with a stent implant coughs too frequently, the implant can become loose, or even dislodged [3]. It is therefore vital that the dog's coughing is carefully observed to prevent the implant from failing. There is a need for a better method or device that can effectively help measure if various pharmaceutical and therapeutic regimens help reduce cough symptoms in dogs.

Our team worked with veterinarians from Tufts Cummings School of Veterinary Medicine to design a cough monitoring device that can be used in a clinical setting. The device was built using a Teensy 3.2 Development Board, which was outfitted with a microphone that ranges from 20Hz to 20kHz [4]. The device uses sound frequencies collected from the microphone to detect coughing episodes based on a predetermined frequency threshold. Based on our research of similar devices, we determined the most effective method for differentiating between a cough and bark was through the selection of target sound frequencies. The device was designed to be portable and durable enough to withstand the daily activities of the dog. Additionally, the weight of the device was no more than 20 percent of the dog's weight to be safe for the dog to wear [5]. Lastly, the device was built to monitor the patient automatically without any user input from the dog's owner. Our device was powered by three AAA batteries, and operated with an on/off switch. Our device aimed to achieve at least 50 percent sensitivity and specificity, which is considered better than random guessing.

Chapter 2. Literature Review

2.1 Overview of Coughing in Canines

2.1.1 Medical Ailments that Cause Coughing in Canines

Coughing in canines may signify a variety of different ailments. When a dog owner brings their pet to the veterinarian for coughing, there is often a series of tests that must be completed before diagnosing the root cause of the cough. One common medical condition in dogs which causes coughing is Tracheobronchitis, more commonly known as "Kennel Cough". This highly contagious respiratory disease is often mild; however, it can lead to pneumonia or chronic bronchitis if not addressed properly. The coughing symptom is described as a deep, dry "honking" cough. Treatment of Kennel Cough patients includes quarantine from other dogs, and treatment from the veterinarian to help treat the symptoms [2].

Another medical condition which causes coughing is a sore throat. A sore throat may indicate a mouth infection, sinus infection, or foreign body stuck in the throat. A foreign body object is very serious, and can be life threatening. Sore throats are often indicated by a high-pitched cough or gagging sound [2].

Pulmonary problems may also induce coughing. These symptoms originate from the lower airway or lungs and are often caused by pneumonia. The cough is caused by fluid buildup in lungs, which induces a cough described as wet and phlegmy. Illnesses with this symptomatic cough needs immediate medical attention [2]. Coughing may also signify heart disease. This particular cough occurs when the dog is lying down or asleep because it is caused by fluids building up around the lungs since the heart is not pumping properly [2]. Coughing may also be an indicator of Tracheal Collapse. This chronic, progressive, and irreversible disease is most common among smaller dog breeds such as Yorkshire terriers, Pomeranians, Poodles, and Chihuahuas, as well as dogs who are obese. The disease occurs in the trachea which keeps airways open allowing dogs to breathe, move, and cough. When the C-shaped cartilage rings become flat, it causes the upper roof of the trachea to stretch and eventually collapse. The symptoms include a dry cough described as a "goose honking" cough, coughing when pulling on a collar or being picked up, difficulty breathing, wheezing sound when inhaling, intolerance to exercise, and fainting.

Diagnosing this disease may include chest x-rays to help rule out other diseases. However, since the collapse may not always be visible, often other methods of diagnosing a collapse are required. Such methods include fluoroscopy to help determine the trachea condition and size during the dog's breathing cycle. Furthermore, endoscopy can be used to view the inside of the trachea using a fiber optic camera. This gives a detailed view of the inside of the airway, and enables samples of the fluid to be taken. An echocardiogram may also be used to ultrasound of the heart. This provides insight on the cardiac function. Once a tracheal collapse is diagnosed, there are a variety of treatment options. Medications and sedation can be used to help reduce airway spasms, coughing, and anxiety. Surgical treatments include the implantation of a stent, which is a spring-like device to hold the trachea open. Post-surgery symptoms include coughing, bleeding, airway damage or larynx paralysis. Many dogs will continue coughing for the rest of their lives, but it is often milder than prior to the treatment. According to the American College of Veterinary Surgeons, 70% of treated patients show improvement from medical treatment and 75% show improvement after the stent implant. Of those stent implant patients, 95% are immediately improved and 90% are markedly improved after post checkups. It is vital that coughing symptoms are tracked and controlled during the recovery period, as it is possible for the stent to become loose or removed if the dog is coughing too frequently, as shown in Figure 1.1 [3].

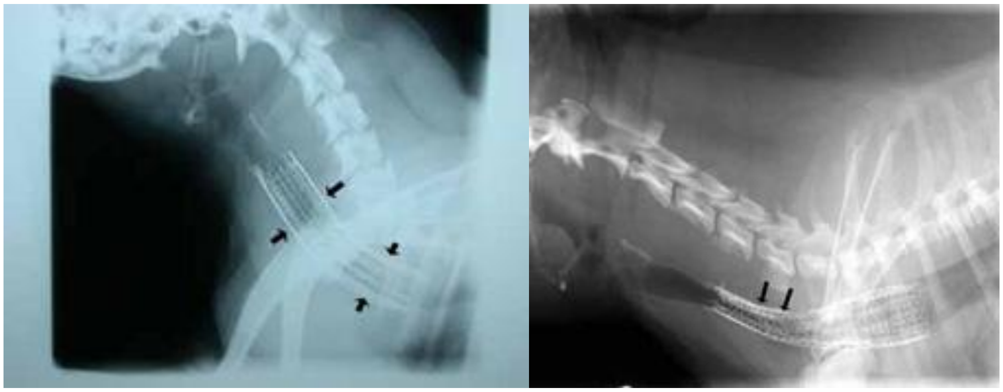


Figure 2.1. X-ray of small dog with a stent implant in which the stent broke due to stress caused by neck movement, resulting in the return of coughing symptoms [3].

2.1.2 Cough Mechanics

Coughing is commonly experienced symptom that provides protection for the lungs through the removal of mucus, foreign particles, and infections. Considered a defense mechanism, coughing allows for the clearing of airways, and is essential for maintaining a healthy respiratory system. However, when coughing is considered chronic, it may become detrimental, and cause pain. Coughing is caused by the stimulation of a reflex arc, which occurs when the trachea, main carina, branching points of large pathways, and distal, smaller airways become irritated [6]. Mechanical and chemical signaling causes a reaction from laryngeal and tracheobronchial receptors. The stimulation of these receptors results in signaling through the vagus nerve to the medulla. This process generates an expiratory muscle response, which induces the cough. There are three main pathways that form the cough reflex arc; the afferent pathway, central pathway, and efferent pathway. The afferent pathway consists of sensory nerve fibers which are located in the upper airways in the ciliated epithelium. Impulses from these nerves diffuse to the medulla, which leads to the central pathway in the upper stem of the brain. Impulses from the central pathway travel to the efferent pathway through the vagus, phrenic, and spinal motor nerves. The diaphragm, abdominal wall and muscles are then stimulated, which induces the inspiratory and expiratory muscles, and leads to the coughing motion [6]. The mechanical process of coughing occurs in three phases; the inspiratory phase, compression phase, and expiratory phase, as shown in Figure 1. In the first phase, the lungs fill with the necessary air-volume to generate the cough. Next, during the compression phase, the larynx closes and the chest wall muscles, diaphragm, and abdominal wall contract, leading to an increase in pressure. Lastly, in the expiratory phase, a high expiratory airflow is released when the glottis opens, and the coughing episode occurs. There is a large compression of the airway, and flow of air expelled from the lungs. This action is how mucus and other particles are dislodged. During the last phase, the coughing sound occurs when the high pressure of air is released [6].

2.2 Sound Frequencies and Pitch

Sounds are vibrations of air molecules created from moving an object [7]. Sound travels in the form of a wave, which causes compressions and expansions in these waves. The speed of sound waves traveling through the air depends on the temperature of the environment. Sound causes the air molecules to move, but if it hits a solid object, the wave will bounce back and create an echo. Sound energy can be changed into other forms of energy like electrical energy, and turned back into sound again [4]. This property allows for the creation of microphones that cause electrical reactions based on the sound properties, such as lights that turn on and off as a reaction to the sound of clapping.

Sound travels differently in different fluids, such as water and air [4]. The molecules move back and forth only a tiny distance, but it is enough to bump into other molecules and cause pressure. A sound wave is created when several of these pressure changes move through the air. This creates places where there are many compressed molecules, and places where molecules are rarefactions [7]. These compressions and rarefactions expand from the source of the sound in circles.

2.3 Relevant Studies on Other Species

The idea of a cough counting device has been explored by various studies for human applications. Many of these studies use sound analysis as an essential part in the cough detection process; however, the specific methods used vary between devices.

2.3.1 Pilot Study of Objective Cough Monitoring in Infants

One study conducted to explore the use of cough monitoring devices for infants, used the Logan Research LR 100, which is a commercial cough counter that was modified for this study. In addition to this monitor, simultaneous sound and infrared video was recorded. For the audio video-recording, the Vista VR, a 24-hour real-time recorder, and the JVC TK-N1100, an infrared camera, and an Optimus omnidirectional boundary microphone were used [17]. This system allowed for both movements and sound to be recorded for 24 hours at a time using a standard 4-hour videotape that needed to be changed in and out. Following the data collection, researchers then reviewed tapes manually, but were still limited to viewing 24 hours of footage in 8 hours. The commercial cough monitor, the Logan Sinclair LR 100, consists of a microphone and EMG leads attached to the chest wall on the middle, left, and right sides. The specific locations used were recommended by the manufacturer. The raw sound and EMG signals were processed through the device and then stored on a memory card. Recording was limited to 18-hour segments, and analysis software used user-set thresholds on each sound and EMG signals to trigger automatic recording.

Initial trials were unsuccessful because of large voltage waveforms from the infant's heartbeat. Additionally, an infant's EMG signal was much weaker than in adults, so the data filtration system had to be adjusted to be more accurate at differentiating a cough from the heartbeat. This adjustment was made by adding a small screw on the back of the LR 100 that altered the signal amplification, which reduced the amount of background noise. This study overall deemed that with some modifications to the existing LR 100, it was possible to objectively identify coughs in infants. This study reported 80% sensitivity and a 0.8 positive predictive value [15].

2.3.2 Automatic Cough Detector Using a Vibroacoustic Sensor

During a study conducted on 14 human subjects, scientists from the Institute of Metrology and Biomedical Engineering in Warsaw, Poland tested the applicability of artificial neural networks and vibroacoustic signals to determine the frequency of coughing. The automatic analysis of the cough "events" uses parametrization of the signals using calculation vectors. The act of coughing is treated as a vibroacoustic signal, meaning it is calculated using both sound frequency analysis and chest vibrations. An omni-directional condenser microphone with a sampling frequency of 20kHz was used to measure the sound component. The vibrations of the chest were measured using a piezoelectric sensor with a sampling frequency of 200Hz. The sensor used was an MLT1132 Piezo Respiratory Belt Transducer by ADInstruments [16]. This sensor responds to change in the thoracic or abdominal circumference and generates a voltage. It is built to function on humans or animals, and can be used for a variety of functions such as breathing rates, or in this case, coughing frequency [17]. Multiple sounds were recorded including an imitated coughing, single hand clapping, shouting, and speaking. The testing procedure consisted of three sections using these four sound samples. The methods were tested on 14 healthy individuals with no reports of respiratory or asthmatic diseases: 6 males (age 22-24), and 8 females (age 23-26).

The first step consisted of episode segmentation using a visually determined threshold technique which calculated changes in the standard deviation of the signals over time. Five parameters were then calculated for the piezoelectric and sound signals. The parameters were calculated based on the signal shapes, and include the following: the mean value of the episode signal, the standard deviation of the episode signal, the area under the mean value, the percentage of the samples' values greater than the mean value, and the ratio of the mean values of the episodes first and second halves. In addition to the 5 parameters listed, spectral parameters from mel-frequency cepstrum analysis and linear predictive coding were also used to analyze the microphone signal. Based on the determination of the inverse transform of signal spectrum logarithm, mel-frequency cepstrum analysis, or MFCA, uses a dedicated filter bank to relate results in a mel scale to subjective sound perception. The coefficient matrix created a set of 13 parameters from the MFCA; furthermore, for each episode's duration, the ratios of the sum of the specific frame values to the sum of all the frames values were calculated. There were nine 8th-order model coefficients from linear predictive coding (LPC). This resulted in a total of 32 parameters calculated for each episode. An artificial neural network was used to differentiate cough episodes from the other sounds. Each sound was classified as 4 distinct output vectors during the learning phase: cough, single hand clap, speech, and sound. The speech portion included five phonemes of the polish pronunciation ("h", "j", "k", "s", "y"). MATLAB software toolboxes were used to test signal processing, parameter calculations, neural network training, and validation. Testing included using 16, 24 and all 32 parameters. The first set of 16 parameters included the time-related sound parameters, the first 3 LPC model parameters, and the first 6 MFCA coefficients of lower frequencies. The set of 24 parameters used the remaining LPC parameters and the third and fourth vibration-related parameters in addition to the set of 16. The last set used all 32 parameters. Testing also considered various network topologies including analysis of two-layer and three-layer perceptrons. The classifier using 32 input parameters and two hidden layers resulted in a sensitivity of 77%, and a specificity of 97%, with a 0.04 standard error. Limitations acknowledged in this study include the absence of background environmental noises used in testing, and the limited number and variation in testing subjects. Future testing which includes testing outside the laboratory, longer test periods, and greater variation in output states is recommended [16]. This study provides insight into the use of multiple types of sensors working in conjunction to determine cough episode frequency. The authors also explore various artificial neural networking techniques to "teach" a program to perform automated cough detection.

2.3.3 Current Cough Monitoring Devices

There are currently a number of cough counting devices; however, they have only been tested on human subjects. Although our project focused on canine coughing, many of the main principles used in these human devices can be directly applied to our own device. The CoughCOUNT™ detects cough sounds using sound recorded from the neck as well as motion recorded on the chest wall. This device has been tested for recording up to 2 hours in various situations, such as sitting, walking, climbing stairs, and in loud environments. We will need to test our device in similar environments, but with more rigorous testing to withstand the daily activities of a dog. Currently trials of the CoughCOUNT™ report a 95% sensitivity and false positive rate of less than 0.7 coughs per hour [18]. This device is important to analyze for our study because of its high success rate, and it serves as a good model for comparing the results of our product.

Another key device for our study is the Leicester Cough Monitor. This device uses a free-field microphone and an off-the-shelf mp3 player. The main concept behind this is to semi-automate the device with the use of speech recognition software and inputs to differentiate between cough and non-cough sounds. The main tests of this device yielded highly variable cough counts, which researchers determined to be a result of inherent issues with speech recognition systems [18]. The Leicester Cough Monitor is key for us as it outlines some of the major pitfalls our device could experience. Some of our initial design plans involve differentiating between coughs and barks. This could lead to some variation as different dogs' barks can have differing sound frequencies, much like humans' voices can have differing sound frequencies.

Chapter 3. Project Strategy

3.1 Client Statement

3.1.1 Initial Client Statement

The veterinarians at Tufts Cummings School of Veterinary Medicine have identified a need for an automatic cough counting device for canines to determine the effectiveness of medical treatments for healing a variety of medical ailments diagnosed from coughing symptoms. The device must be easy to use, portable, durable, and small enough to comfortably fit small canine breeds. We focus our efforts on small canines to satisfy the project parameters specified by our client.

3.1.2 Objectives

Our goal for this project was to design and build a device that can achieve more than 50% sensitivity and specificity in identifying a cough episode of small canine breeds. We selected this value because it signifies that the device is better than random guessing. The main objectives of the project were to:

- Identify regulations and restrictions for our design criteria
- Evaluate current methods for coughing detection, and identify areas of improvement

- Determine a frequency threshold to distinguish coughs from other canine and environmental sounds
- Develop a program which can filter sound inputs and accurately track coughs
- Build a circuit which can be programmed to record and log coughing episodes, and interface with a computer to upload the data
- Create a user interface that displays the data collected by the device, and allows veterinarians to track patient recovery
- Evaluate the performance of the device using real-world sound recording data

All the objectives were compared against each other using a pairwise comparison chart shown below.

Table 3.1. Pairwise Comparison Chart.

Objectives	Durable	Portable	Light-Weight	User-Friendly	Record Data	Distinguish Cough and Bark	Total
Durability	-	3	3	3	3	3	15
Portable	1	-	2	3	3	3	12
Light-Weight	1	2	-	3	3	3	12
User-Friendly	1	1	1	-	3	3	9
Record Data	1	1	1	2	-	2	7
Distinguish Cough and Bark	1	1	1	1	2	-	6

In the chart above, 1 represents that the top row objective is less important than the left column objective, 2 represents that they are equal and 3 represents that the top row objective is more important than the left column objective. Following this, the lowest number is the most important objective. As seen above, the most important function for our device was to distinguish between dog coughs and barks. The second most important function was to be able to record the collected data. Without this aspect, the veterinarians would not be able to tell whether the device was functional or how to proceed with treatment. Given the veterinarians' intent to use our device, we determined our device should be easy to use and read the collected data. Following these objectives, comes the weight of our device. As our intended recipients of this device are small dogs, our device should be lightweight, ideally less than 1 lb, or 454 g. We determined this target weight based on our intention to have the device be less than 20% of a 6 lb dog's bodyweight. The device also needed to be durable and portable, such that the patient was not restricted while wearing it. Although these are all ideal objectives that our device will fulfill, the most important are the ability to distinguish between a cough and a bark, and the ability to record collected data.

We researched and tested possible designs to build a prototype for testing on actual patients. Following our initial study, we established future recommendations and device improvements for our clients at Tufts, and for future WPI students.

3.1.3 Constraints

The main constraints of this project were time, cost, and resources. The timeline for this project totaled to 8 months. Unfortunately, this lack of time affected the scope of the project. Our budget was limited to $1,000, which affected the number of prototypes we could develop. Additionally, to test our device, we required a large number of canine coughing and barking sound samples. This amount of sound data was limited by the number of dogs in the clinic, and proved to be the biggest impediment for this project. Another major factor that constrained our work was the technical capabilities from our team. As mechanical and biomedical engineers, our technical abilities in programming and electrical work were limited. Although we did make progress in coding and building our circuit, we were limited to basic programming and modification of software.

3.1.4 Revised Client Statement

Following meetings with our clients, we revised our client statement to better fit their needs. We have determined that an owner must be able to take the device home to be used throughout a predetermined period of testing, typically 24-48 hours. Ideally this device should be able to count coughs for veterinarians and drug companies for use as a diagnostic tool in assessing the progression of ailments and other prescribed treatments. Based on our findings, we developed the following revised client statement.

The veterinarians at Tufts Cummings School of Veterinary Medicine have identified a need for an automatic cough counting device for canines to determine the effectiveness of medical treatments for healing a variety of medical ailments diagnosed from coughing symptoms. The device must display the cough count in real-time, and be easy to operate. Additionally, the device must weigh no more than 454 grams, and be nonrestrictive, durable, and water resistant, to withstand the daily activities of a small dog. Furthermore, the device must have above a 50% sensitivity and specificity for detecting coughs.

Chapter 4. Design Process

4.1 Needs Analysis

Our project statement is an overview of the main goals of this project, but do not fully encompass all that we hoped to achieve. In our first prototype we aimed to reach more than 50% sensitivity and 50% specificity. Ideally for a marketable product, our target sensitivity and specificity rate would be above 95%, as this is generally expected for the device to be used commercially and by drug companies. Ideally, we would have liked to create a unique Graphic User Interface (GUI) to display the data in a variety of different metrics, such as graphs. Unfortunately, creating a GUI like this requires a higher level of software development expertise that our team members did not have.

The device needed to be durable enough to withstand a dog's daily activities, such as laying down, running, and playing. Additionally, we intended on making the device water resistant., such that the patient could drink water unrestricted. Having a water-resistant device made our device more durable and less sensitive to daily wear and tear. Another feature we needed was an on/off switch or button to manually reset the program with ease.

Table 4.1. Needs vs. Wants.

Needs	Wants
50% sensitivity and specificity	95% sensitivity and specificity
Count display	Universal GUI
Durable case	Waterproof case
Battery life > 48 hours	Case with on/off/reset buttons

The table above outlines our most important needs and wants. We planned to focus on satisfying our needs before moving on to our wants.

4.2 Design Requirements

To successfully complete our project, our device needed to satisfy certain functions that we determined based on interviews with our client. The overall timespan for this project was twenty-five weeks, and was completed with a budget of $1,000. As a basic starting point for our project, we determined that our device should achieve above 50% sensitivity and 50% specificity. As the patients utilizing this device are small dogs, weighing between 6-8 pounds, we designed a device weighing 454 grams or less. Furthermore, the case for the device was designed to be water resistant and durable.

In addition to physical specifications, our device needed to be easy to use by both the dog owner and the treating veterinarian. This meant we needed a simple way to turn on/off the device, and rest it, as well as a simple display mechanism for the cough counter.

4.3 Key Industry Standards

To ensure our device would be safe for clinical use, we established certain standards in which the testing procedures needed to follow. The first standard is ISO 16142-1:2016, which describes essential principles of safety and performance that apply to all medical devices including non-in vitro diagnostic (non-IVD) medical devices [19]. This standard is needed for the hardware components of our design to ensure that no harm is caused by use of the medical device. The second standard is ISO 14971:2007, which states the related methods of risk management as applied to safety-related aspects of medical device user interfaces [20]. This will ensure that our software is safe to use and meets the industry standards and to terminate the program in case of malfunction. These standards ensure that our hardware and software are comply with industry safety standards for living subjects. They also cover the basic ethical guidelines on medical devices. These requirements include operation at a safe temperature of no more than 32 degrees Celsius, a clear failure shutdown program to immediately end a test in case of an emergency, and the creation of an emergency button to manually stop testing of the device. Identifying these requirements was an important aspect of our component selection and device design process.

4.4 Design Process and Final Design Selection

4.4.1 Initial Conceptual Design Components

The first iteration of our design revolved around the use of a pre-established MATLAB sound analysis program. This existing program was developed by other students and recommended for our use by another professor. This program is designed to receive a sound input and return a variety of analysis in the form of graphs and tables. Unfortunately, this program did not have the appropriate tools we needed. We had been hoping to use it as a spectrum analyzer to identify a threshold for distinct frequencies. We were unable to modify this program and with feedback from various professors, decided to abandon this line of design. We determined that changing the program was well out of our project scope and would have been an inefficient use of our time.

Along with this MATLAB code, we originally planned to use an Arduino Nano board. The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). We had planned to use this board because of its general ease of operation and functionality. Based on preliminary research, this was also the simplest means of building our circuit. Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts, which is compatible with our power supply. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. For our power supply we have chosen to use three AAA batteries, which can be used with the Arduino Nano. Other power options include via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or a 5V regulated external power supply (pin 27). To record and collect data with Arduino, we selected the ATmega328 which has 32 KB of storage capacity.

For our initial design, we chose to follow an existing spectrum analysis program from an Adafruit tutorial. Based on the recommendation of a trusted professor, our design progressed from using MATLAB code with an Arduino Nano, to coding directly in the Arduino program with a Teensy board from Adafruit. The Arduino programs use Python libraries as a base for all processes, even though direct coding is performed in C/C++.

Our first circuit was built based from a working prototype from an existing design that analyzes sound using a programmed Teensy 3.2 Board. This circuit uses a Fast Fourier Transform (FFT) to analyze sound frequency inputs through an Adafruit Electret Microphone

Amplifier MAX-4466 with Adjustable Gain. We switched from the Arduino Nano to the Teensy board because it was better suited to follow this tutorial. Additionally, the device's small, compact design suited our size and weight requirements. The Adafruit microphone was selected because it is specifically made for applications involving FFT and sound analysis programs. The circuit design used to build our prototypes is shown in Figure 4.1. This circuit design was taken directly from the tutorial found on Adafruit's website [19]. We used this circuit as a basic starting point from which to create our device's circuit.

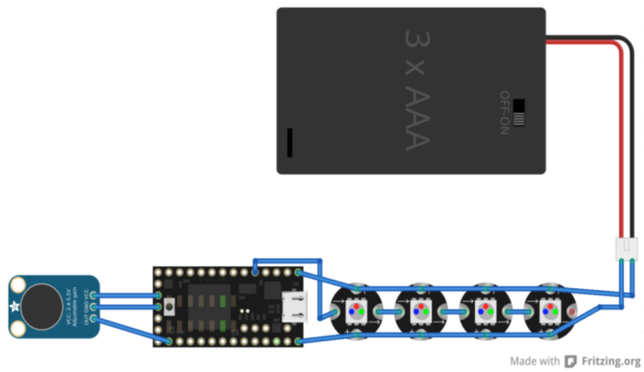


Figure 4.1. Circuit design using a Teensy 3.2 board, Adafruit Electret Microphone, and Adafruit Neopixel LEDs. The device runs on a power from 3 x AAA batteries [21].

As shown in Figure 4.1, the circuit uses Adafruit Flora RGB Smart Neopixel LEDs as an indicator of when the device detects a sound within a predetermined frequency threshold. The circuit uses a sampling rate of 4000 Hz, and uses different frequency windows assigned to each LED. By building a general frequency analyzer, we were able to modify the code to apply the basic principles to our device.

The next step in creating our device was implementing a tone detection code. A frequency threshold was set at the beginning of the code, and each time a sound was detected within the threshold, the LEDs would blink.

In order to modify these existing programs to suit our project, we targeted three specific areas. The primary changes were the frequency threshold, the addition of a count, a way to display the number of coughs detected. The first section we modified was the frequency threshold. For our device to differentiate between a cough and a bark we had to define the target frequencies that a cough would cover. To do this we analyzed cough recordings collected by our clients at Tufts. We discovered that thorough testing was needed to determine a frequency threshold that would detect coughs and reduce false negatives. Another key aspect of this project was a way to count the coughs. This was achieved by introducing a toneCount function, which stores the total number of coughs detected. To visualize the total count in real time, we utilized a 7-segment digital display. Because we were able to display the count with the 7-segment display, we eliminated the need for our device to write to a text file and to store data. We found that the 7-segment display would be a more reliable and time efficient way to display the data.

4.4.2 Alternative Designs

Throughout our design process, we considered a few design options. Based on our background research into this subject we determined that using multiple methods of cough monitoring would create a more reliable device. Many of the existing devices for human cough monitoring analyze some variation of pressure and sound. Options for looking at pressure changes include throat vibration or chest inflation analysis using sensors directly on the skin or with monitors tightly around the patient's chest or throat. We determined that this was not a viable option for our device, because most of our small dog patients cannot have a collar around their necks post-surgery, or a tight and restrictive belt. We also encountered the issue of fur getting in the way of skin sensors, and determined it would not be feasible to shave the entire neck and chest area for the use of our device. Ideally, vibration analysis using pressure transducers could be another way to strengthen the sensitivity of our device, which we have outlined in our future recommendations found in Section 8.2.

Another option we considered for our device was to utilize a "voice" recognition software to analyze each cough event. This method would involve creating a neural network to differentiate a cough based on its unique sound profile or fingerprint. We considered using MATLAB to write program that analyzes a comprehensive set of characteristics in each sound, then compares the detected sound to a pre-set sound profile of a typical cough. To determine what a typical cough would be, we needed to analyze thousands of samples to account for the wide variation seen in coughs. This variation is a result of each dog having an immeasurable amount of sounds it can generate, as well as the wide variety of cough types. A single dog may have many different coughs, i.e. a dry hacking cough or a wet cough. Each cough has a different sound profile and to account for each of these types, our program would need to be able to recognize each cough type versus any other sound. This includes environmental noises and any other noises the dog itself could make, whether that be barking, panting, whining, etc. For our program to be effective, it would need to recognize this variation across many dogs. Unfortunately, creating this type of program is well out of the scope of our project. We were limited by time, number of available sound samples, and by our skill level. As mechanical and biomedical engineers, we do not have the expertise required to code and generate a program of this magnitude. We consulted many professors specializing in electrical engineering, computer science, robotics, and music (for sound analysis) and were advised that a project of this scale should be a postgraduate endeavor for a group specializing in this type of sound analysis and programming.

An additional part of our previous design with the Teensy board was to modify the existing code to make a time stamp whenever the device detected a cough. The timestamp would allow the veterinarians to know exactly when a dog coughed to make a better diagnosis based on continuous coughing or short periods of coughing during activities. This design also had an SD card holder wired to the Teensy to collect all of the data and make a GUI. However, this design resulted in problems sending the data from the Teensy to the SD card. After consulting ECE and CS professors, we discovered that the Teensy is not setup to handle this operation, and most of the coding would need to be done from scratch. Additionally, the data was displayed on a raw text file which only kept track of the seconds since the start time, not the universal time. We determined that this way of visualizing the data was not ideal, as it was not user friendly. We concluded that although the 7-segment display did not include a timestamp for each cough episode, it was a much simpler way of displaying the data.

4.4.3 Final Design Components

Our final circuit design was comprised of five distinct components. The hardware consisted of one Teensy board, one electret microphone, one Neopixel LED, one 7-segment display, and one battery pack with three AAA batteries. The device was segmented into two parts; the battery pack and the device box which contained our circuit. We secured the wiring with a heat-shrink plastic seal, and concealed it under the harness component. The battery pack was a pre-existing casing for three AAA batteries, with an on/off switch. This satisfied our need for a simple way to power the device, and to turn it on and off easily.

Our final circuit design is shown in Figure 4.2. The software used in our device follows the basic tone detection software, with significant modifications to fit our needs. Much of the code in the tutorial was unnecessary to our objective, and the overall output of the code did not reflect the purpose of our device, so it was heavily edited. The code functions by performing a Fast Fourier transform on the raw imputed sound signal, and analyzing the frequency to determine if the sound recorded is within a certain threshold defined by the user. If a sound frequency within the threshold defined is detected, the program signals the LED to light up, and the 7-Segment display to present the correct number of sounds that have been detected since time zero. The counter can easily be reset by simply turning the device off and on. A list of all components used for our device is shown in Figure 4.3.

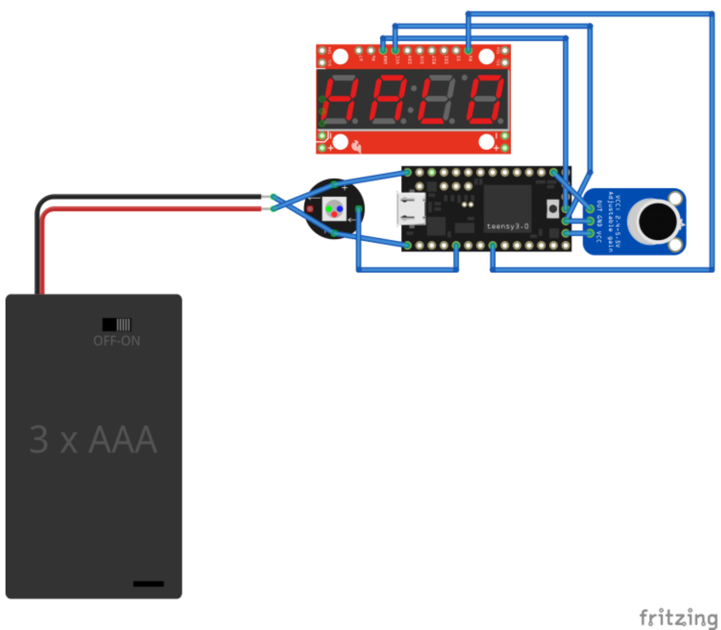


Figure 4.2. Design of our final circuit including a Teensy 3.2 board, Adafruit electret microphone, 7-segment LED display, Neopixel LED, and battery pack containing three triple-A batteries.

Figure 4.3. List of components used to build our final device design.

To secure and protect the circuit, we designed a case which was 3D printed in WPI's Rapid Prototyping Lab. We consulted with a 3D printing specialist from the mechanical engineering department to verify our design, and receive advice on the best material and dimensions to satisfy our needs. We selected a clear Lexan plastic for the case lid due to its strong mechanical properties. Our final casing design is shown in Figure 4.4.

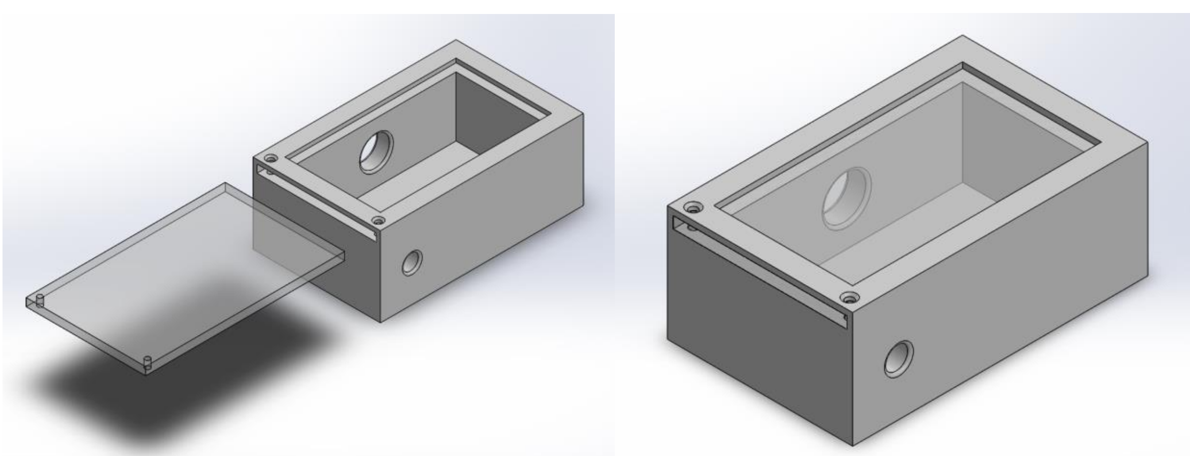


Figure 4.4. Plastic casing for our device with clear sliding lid.

Two through holes were designed to allow for proper placement of the microphone and connecting wires to the battery case. There are two through holes on the left side of the case, and the clear sliding to secure the lid with two 1/16" screws purchased from McMaster-Carr. The case was printed using the Dimension SST 1200es 3D printer in WPI's rapid prototyping lab.

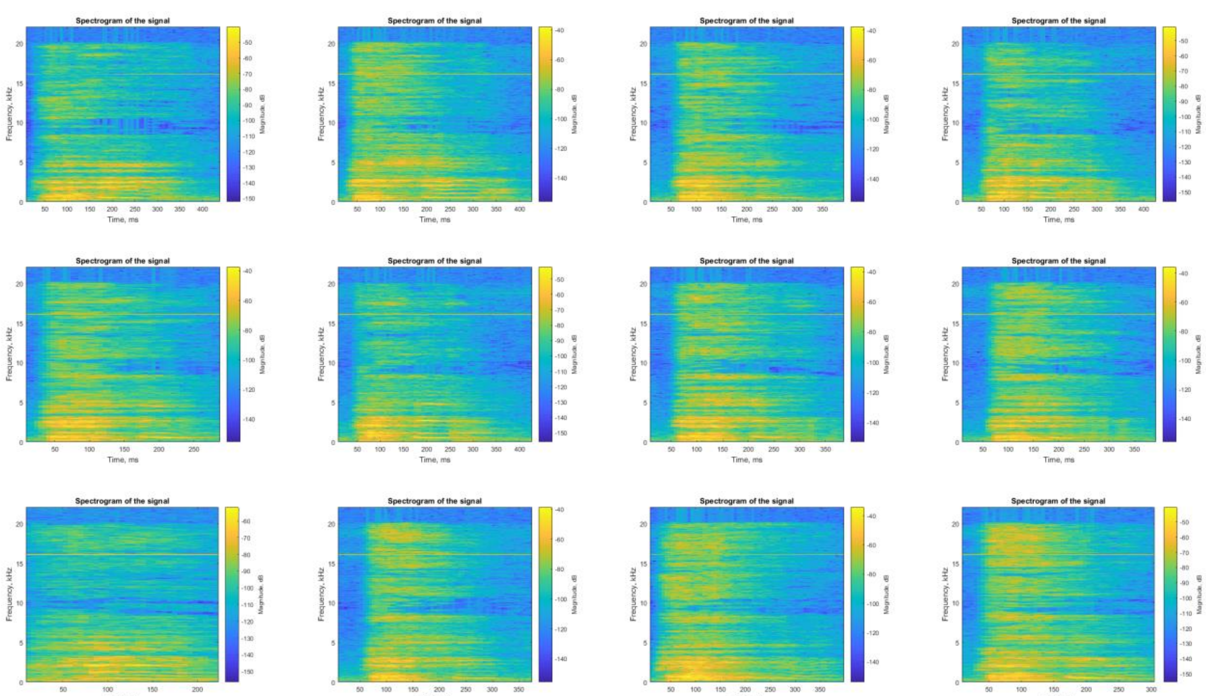
For the final steps, we used a sturdy mesh harness to mount the device. This type of harness, seen in Figure 4.5, is sturdy, and can easily support the weight of our device. We attached the battery pack and circuit case of the device to the harness via industrial Velcro so the components can be easily removed in needed, and transferred to various harness sizes.

Figure 4.5. Small mesh dog harness our device was mounted on to reduce interference with the dog's daily activities, and hold the components of our device in a compact, and stable way [24].

Chapter 5. Design Verification and Testing

5.1 Signal Processing Sound Samples to Determine Frequency Threshold

A vital aspect of our device testing was determining what the frequency threshold of our program should be based on the coughs of small canines. The veterinarians at Tufts provided us with a selection of audio recordings of small dog patients suffering from coughing symptoms. Because our project required specific sampling, the veterinarians were not able to provide the group with a large library of samples; however, we worked with the samples we were given, and based our final threshold on the provided samples. First, we imported the sound files to Audacity to filter out background noise, and cut the audio file into individual cough clips. We then used MATLAB to analyze the samples and test their frequency characteristics. We first tested these cough clips by running them through a Fast Fourier Transform and plotting their spectrograms. This is a helpful visual aid to observe common attributes of coughing from that particular patient. Shown in Figure 5.1 are the spectrograms of twenty-one coughs from a small dog with Bronchial Collapse. The code written to accomplish this method can be found in Appendix A.



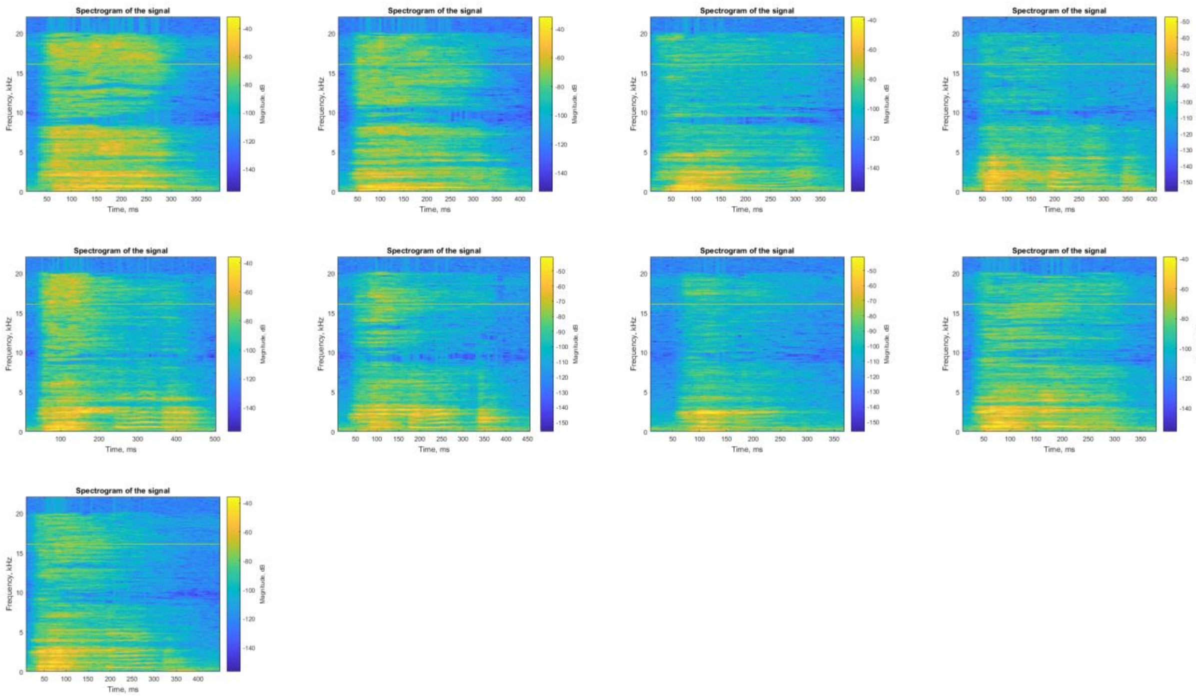


Figure 5.1. The spectrograms of cough samples from a small dog with Bronchial Collapse.

As shown in Figure 5.1, there is a clear pattern that represents the unique profile of the sound generated by a canine cough. There is a distinct increase in signal strength in the lower end of the frequency between 250 - 800 Hz. To further test these samples and narrow down our threshold, we conducted another Fast Fourier Transform on the samples and plotted the frequency domain, as shown in Figure 5.2.

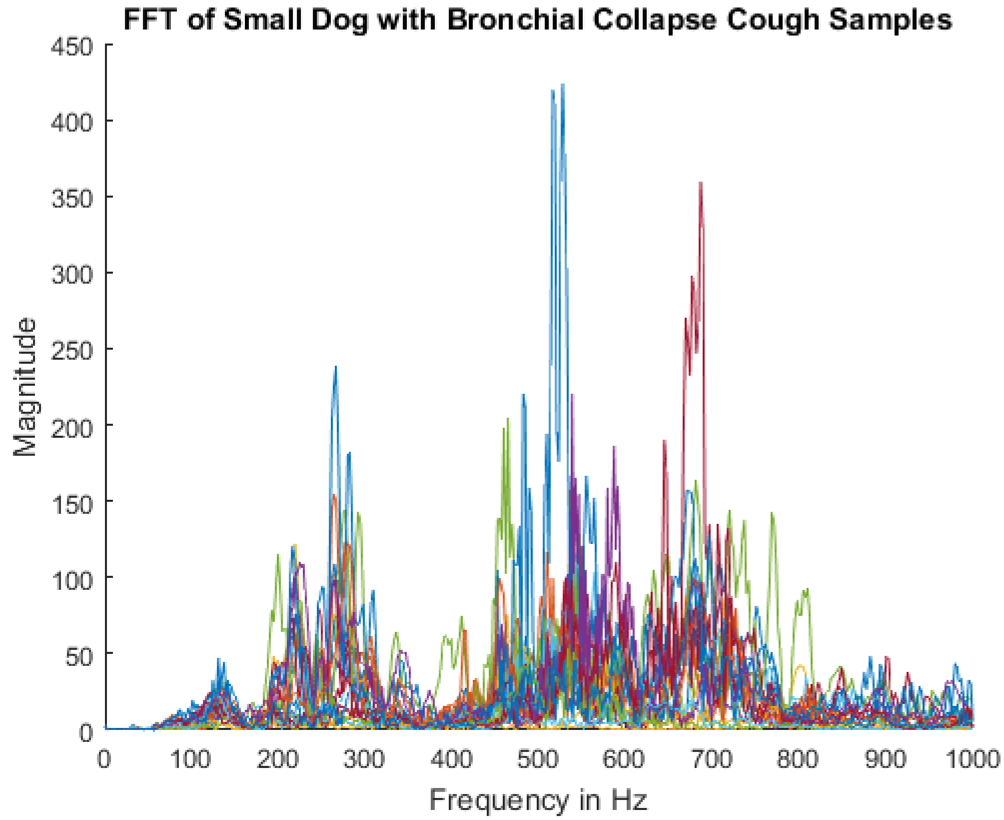


Figure 5.2. FFT of 21 Bronchial Collapse Coughs. The graph represents the magnitude of the frequencies in the sound signal.

We observed that there are a few frequencies in which the sound signal peaks These peaks range between about 450 Hz and 750 Hz.

An important factor to consider with our device is how the properties of other sounds made by dogs compare to coughing. One concern with our device is that barking could be misclassified as a cough episode. To investigate this potential problem, we analyzed sound samples from small dogs barking, Samples from three small dogs of varying breeds were used. These subjects include a 12 lb beagle, a 6 lb Bichon Frise, and an 8 lb Toy Poodle mix. An FFT was performed on recordings from these dogs barking, and the results were plotted as shown in Figures 5.3 - 5.6.

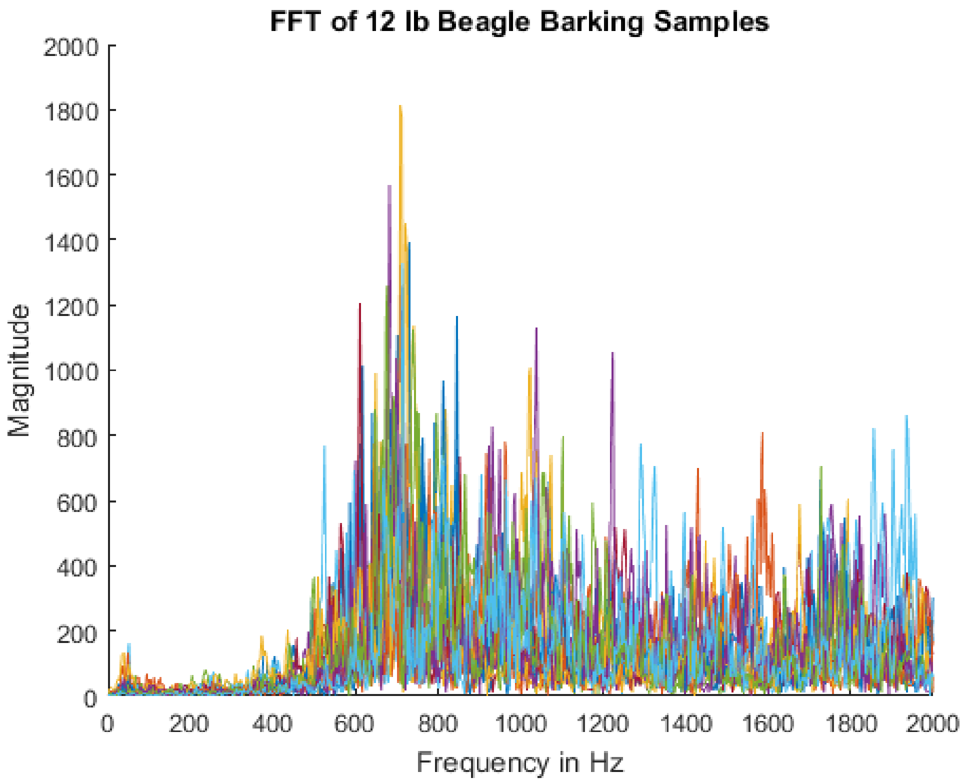


Figure 5.3. FFT of barking samples obtained from a 12 lb beagle.

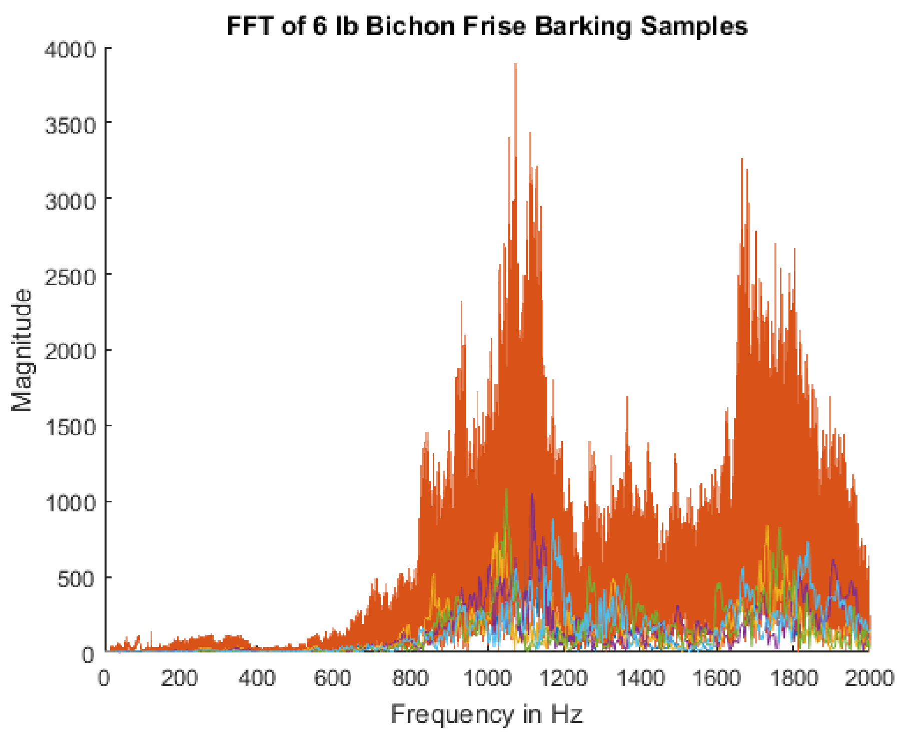


Figure 5.4. FFT of barking samples obtained from a 6 lb Bichon Frise.

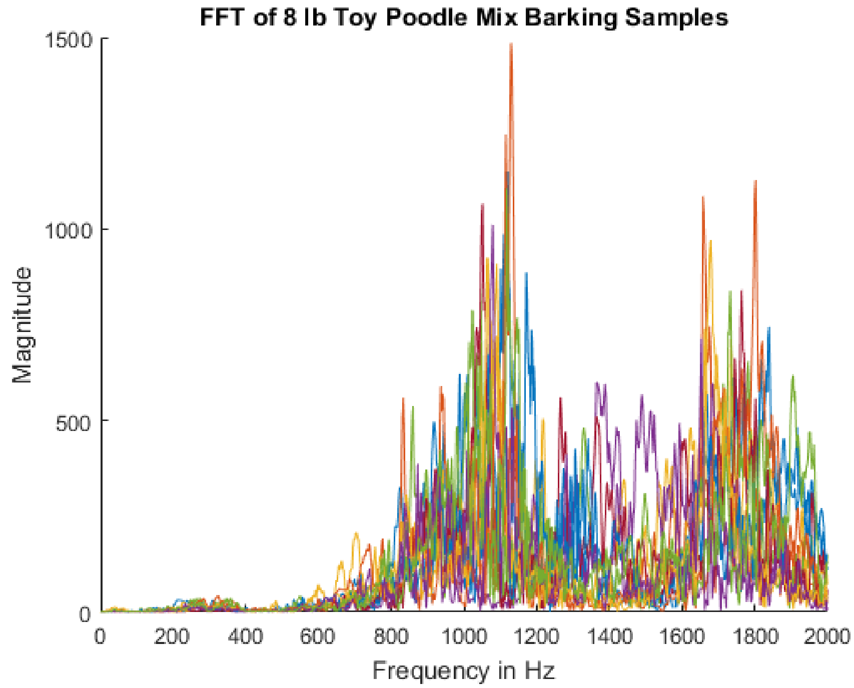


Figure 5.5. FFT of barking samples obtained from an 8 lb To Poodle mix.

We then overlaid the FFT's of all the barking samples, to obtain the graph show in Figure 5.6.

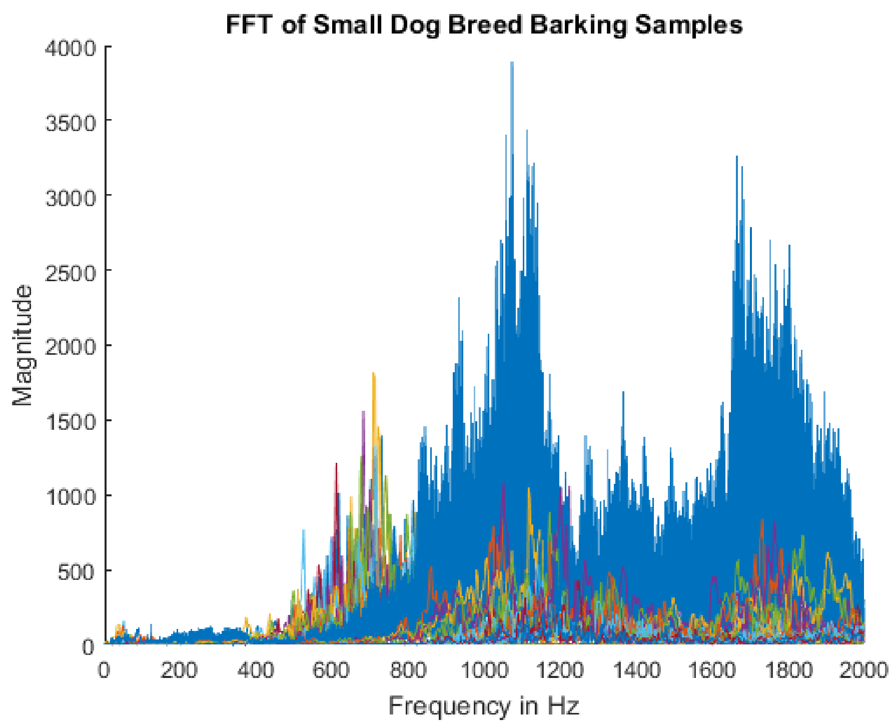


Figure 5.6. FFT of barking samples obtained from all three dogs.

Based on the results obtained from the FFT's of these dogs, we observed that the peak frequency of barking occurs between 800 and 2000 Hz. This suggested that the frequency of barking is significantly greater than coughing. Using the results from these plots, as well as the plots from the coughing samples, we determined the frequency threshold for our device to be 450-750 Hz. We selected these boundaries because the magnitude of the coughing samples was highest between these frequencies, and the barking samples displayed the highest magnitudes at greater frequencies. The MATLAB code written to create Figures 5.2 - 5.6 can be found in Appendix B.

5.2 Testing of Device Using Sound Samples

To determine the sensitivity and specificity of our device, we developed a confusion matrix. We used the sound samples provided by the veterinarians to test how effective our device was in detecting coughs without picking up other sounds made by canines. As distinguishing coughs from barks was one of our biggest concerns, we tested the device with coughing and barking samples. The confusion matrix layout is shown in Table 5.1.

Table 5.1. Confusion Matrix Used to Determine the Effectiveness of the device.

		Truth		
		Cough	**Bark**	
Test Result	+	True + (coughs)	False + (barks detected as coughs)	**Total Test +**
	-	False - (undetected coughs)	True - (undetected barks)	**Total Test -**
		Total Cough	**Total Bark**	**Total**

With the frequency threshold set to 450 - 750 Hz, the device was placed between two high-definition speakers, which were set to play a series of sound samples, as shown in Figure 5.7.

Figure 5.7. Speaker and device setup for testing sound detection.

Four sound files were tested, consisting of a total of 55 sound episodes; 21 of which were coughs and 34 of which were barks from various breeds of small dogs. Each sound file was played 5 times, and the number of sounds detected by the device was recorded. The results from each trial are shown in Table 5.2.

Table 5.2. Results from Device Sound Testing.

Sound File	True Number of Coughs/Barks	Test 1	Test 2	Test 3	Test 4	Test 5	Average
Bronchial Collapse Coughs	21	19	18	18	18	17	18
12 lb Beagle Barking	13	4	3	4	4	4	3.8
6 lb Bichon Frise Barking	11	0	0	0	0	0	0
8 lb Toy Poodle Mix Barking	10	0	0	0	0	0	0
Total Coughs	21	19	18	18	18	17	18
Total Barks	34	4	3	4	4	4	3.8

Out of the true 21 coughs, the average number of coughs detected was 18. From the 34 barks, on average only 3.8 barks were falsely detected as coughs. Using the average numbers of coughs and barks, the confusion matrix shown in Table 5.3 was developed to calculate the sensitivity and specificity of the device.

Table 5.3. Confusion Matrix of Sound Detection Test Comparing Coughs and Barks.

		Truth			
		Cough	*Bark*	Total Number	
Test Result	*Positive*	True + = 18	False + = 3.8	Total Test +	21.8
	Negative	False – = 3	True – = 30.2	Total Test -	33.2
		Total Cough	Total Bark	Total =	55
		21	34		

Using the data from this confusion, Equations 1 and 2 were used to determine the sensitivity and specificity of the device.

$$Sensitivity = \frac{True\ Positive}{(True\ Positive + False\ Negative)}$$

Equation 1. Formula used to calculate the sensitivity of our device

$$Specificity = \frac{True\ Negative}{(True\ Negative + False\ Positive)}$$

Equation 2. Formula used to calculate the specificity of our device

Using these equations, the sensitivity and specificity were found to be 86% and 89% respectively. These results suggest that using sound analysis is a viable method for distinguishing coughs from barks.

In addition to the sensitivity and specificity, we also used the confusion matrix to determine the accuracy, misclassification rate, true positive rate, false positive rate, and precision. The values from these tests helped us determine how often the device is wrong, and how often it will correctly and incorrectly predict yes. The values from these tests are shown in Table 5.4.

Table 5.4. Confusion Matrix Testing.

Accuracy	**Misclassification rate**	**True Positive Rate**	**False Positive Rate**	**Precision**
0.88	0.12	0.86	0.11	0.83

Statistical analysis of the sound testing was performed to determine the significance of the data. A paired, two-tailed t-test was performed to compare the average coughs and barks detected from the five trials, and the hypothesized, or true values. The t-test resulted in a value of 0.019, which means there was no significant difference between the true number of coughs and barks, and the detected number of coughs and barks. The results of this test are shown in Table 5.5.

Table 5.5. Statistical Analysis of Sound Testing.

Sound File	**True Number of Coughs/Barks**	**Average Number of Coughs/Barks Detected**	**Standard Deviation**	**Two-tailed, Paired T-Test**
Bronchial Collapse Coughs	21	18	0.71	0.019
12 lb Beagle Barking	13	3.8	0.45	
6 lb Bichon Frise Barking	11	0	0	
8 lb Toy Poodle Mix Barking	10	0	0	

It should be noted that we acknowledged this testing method was not ideal, as the device should be tested on real coughing patients. However, due to time constraints and testing restrictions on patients at Tufts Veterinary clinic, we were unable to test the device on coughing patients.

5.3 Mechanical Testing of Device Casing

5.3.1 Compression Testing of Device Casing

To meet the needs of the client statement, our device required a certain level of durability to withstand the daily activities of a canine patient. To ensure our device met these durability requirements, we implemented compressive testing on the casing. We determined the maximum impact force our object may be subject to using the work-energy principle. Our device was expected to be used on canines with a maximum weight of 8 lbs, so we used a maximum mass of 3.63 kg plus the weight of the case which was 170.2 g. The maximum drop height was determined to be 0.01016 meters based on the estimated height of a Yorkshire Terrier's legs, which is an example of our target patient [25]. We used this height as it measures the actual falling distance of the device from the dog's chest, directly above its legs. We estimated the distance after impact to be 0.03 m based on how we predicted the device would fall. Equations 3-5 show the calculations used to determine an impact force of 126 N.

Mass = m = 3.8 kg

Height = h = 0.01016 m

Gravity = g = 9.81 m/s^2

Impact distance = d = estimated to be 0.03 m

$$Velocity = v = \sqrt{2gh}$$

$$= 1.411 \text{ m/s}$$

Equation 3. Calculation to determine velocity

$$Kinetic\ Energy = KE = \frac{1}{2}mv^2$$

$$= 3.784 \text{ J}$$

Equation 4. Calculation to determine the kinetic energy

$$F(impact)\ x\ d = \Delta KE$$

$$F(impact) = 3.784\, J\ /\ d$$

$$= 126 \text{ N}$$

Equation 5. Calculation to determine the impact force

Our device was compressively tested on an Instron. To comprehensively assess the loading limits, we tested three different orientations of the device, as seen in Figure 5.8 – 5.10.

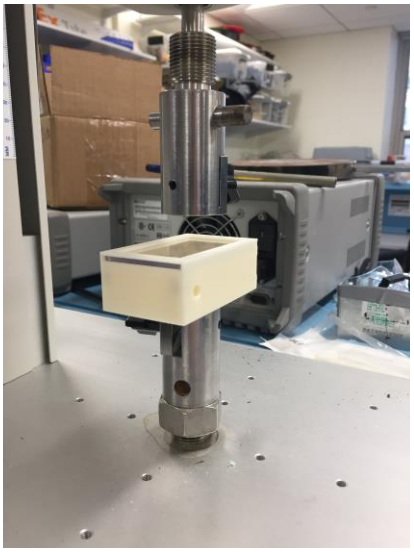


Figure 5.8. Compression testing procedure to determine the strength of our case oriented heightwise.

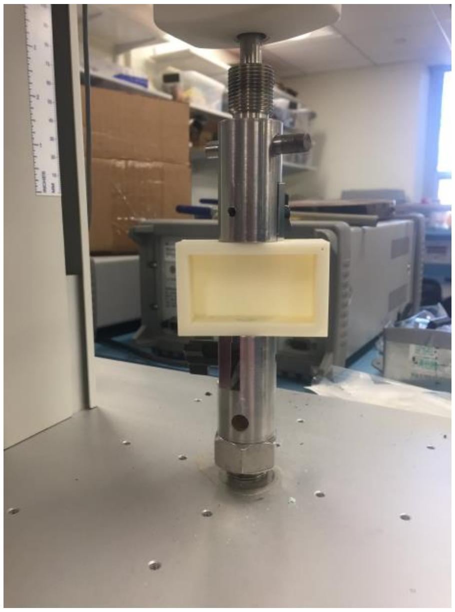


Figure 5.9. Compression testing procedure to determine the strength of our case oriented widthwise.

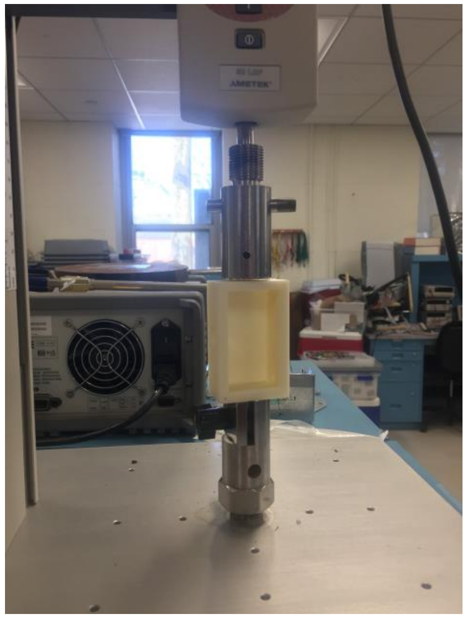


Figure 5.10. Compression testing procedure to determine the strength of our case oriented lengthwise.

We chose to conduct non-destructive tests on this casing, as we intended to use this case in our final design. The compressive loads we applied on the casing were much higher than anything we expected it would endure in a clinical setting. We determined the maximum force on the clear Lexan sliding top of the case to be 50 N. We stopped the Instron at 50 N because we saw visible bending in the top. It should be noted that due to the design of the instron, the compressive force was applied directly to the Lexan plastic, and therefore was not distributed to the plastic casing support frame that holds the lid in place. We suspect if a dog were to fall on the case, the force would be distributed evenly among the top surface, and the maximum force would be increased. Based on the way the case is oriented on the harness, if the dog were to fall directly on the case, the orientation would most likely be widthwise, or height-wise. In the width-wise orientation, the case was able to withstand forces upwards of 120 N. Based on the force curves, we suspect the case would withstand forces significantly greater; however, the test was stopped to ensure with confidence that no breakage would occur. Using the values calculated in our free fall impact calculations, we are confident that our device would not break given the condition that the patient was to fall directly on the case.

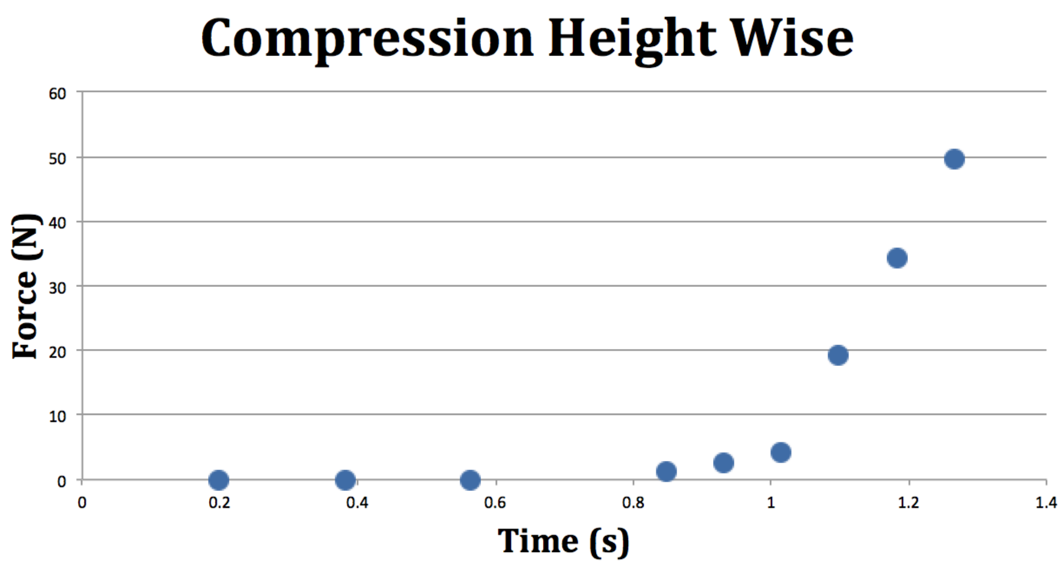


Figure 5.11. Data from Instron compression test in orientation shown in Figure 14.

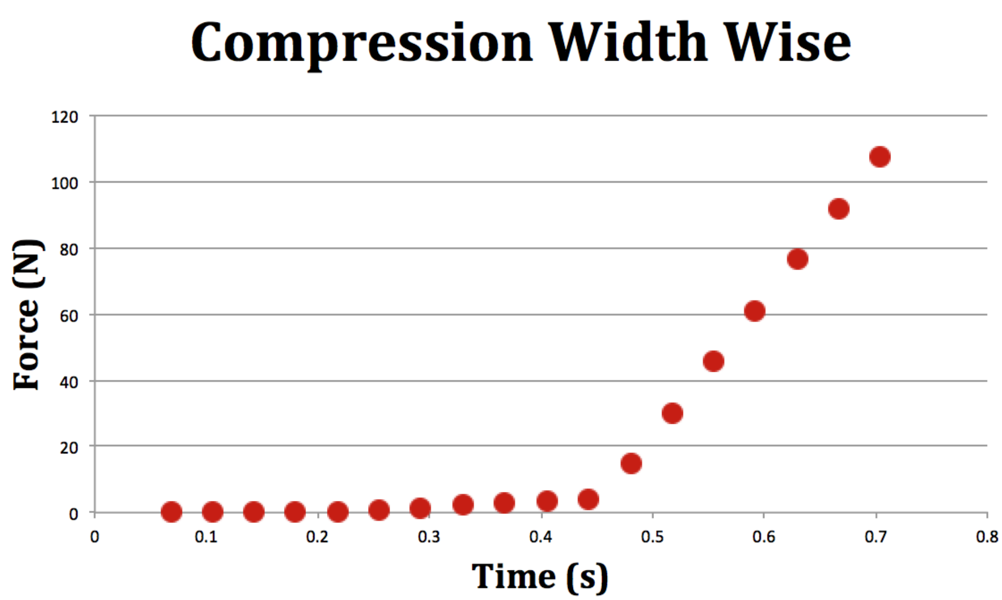


Figure 5.12. Data from Instron compression test in orientation shown in Figure 5.8.

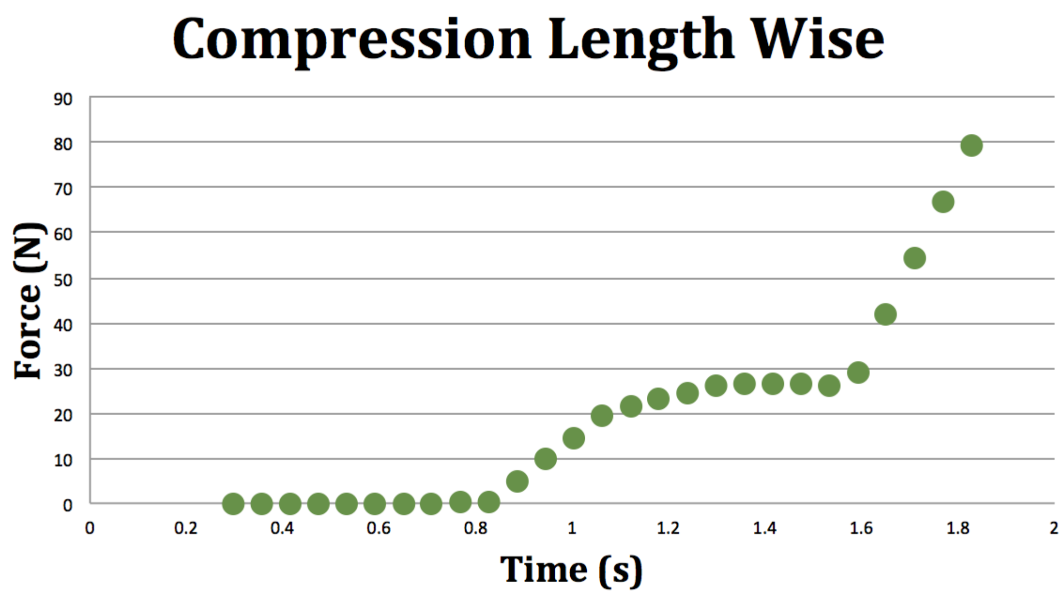


Figure 5.13. Data from Instron compression test in orientation shown in Figure 5.9.

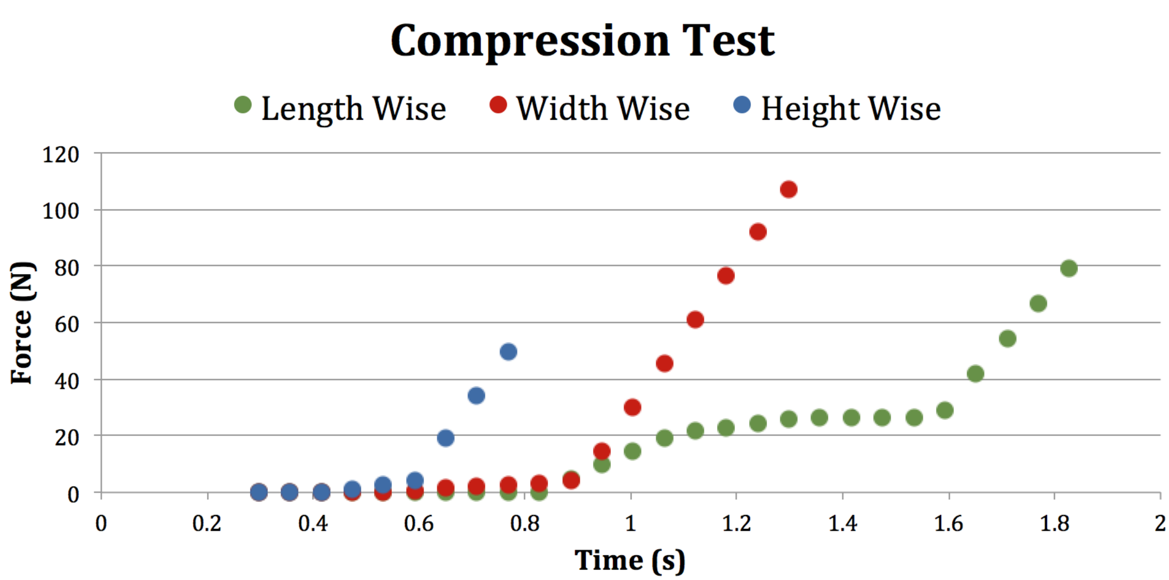


Figure 5.14. Overall data from Instron compression test 5.10.

Statistical analysis of the compression test determined a confidence of 99% that the material of the case will not break under 50 N of force, as shown in Equation 6. The casing was tested in 30 trials and in each test the casing withstood a load greater than 50 N. These results allowed us to determine that the device could withstand the impact of a falling dog, and is therefore mechanically safe to use for our intended purposes.

$$Xinitial\colon Xaverage \ \underset{\sim}{+} \ t1 - \frac{a}{2}, dif.n - 1 \ * S * \sqrt{1 + \frac{1}{n}} = 99\% \ confidence$$

Equation 6. Individual Confidence Interval

5.3.2 Drop Testing of Device Casing

In addition to the compressive testing, we performed a drop test to quantify the overall durability of our casing. We followed the ASTM Standard Test Method for Drop Test of Loaded Containers by Free Fall to set up our testing procedure [23]. We tested our device at heights ranging from 6 ft to 1.5 ft, decreasing every 1.5ft. We selected 6 ft as the maximum height for our testing because we felt that the device would not be in a situation where it would fall from a height exceeding this. The typical heights this device should face should be approximately 4 ft. This could be due to a dog jumping off a (high) couch or table or the device accidentally falling off a table or shelf. To fully test the case's durability, we had to conduct the drop test at each height, for each orientation of the box which yields 24 different trials. After each trial we examined the case for any damage. Following the 24 trials, no damage or wear was visible on the case. We concluded that the case is drop-proof under 6ft. The results of this testing can be seen in Appendix C.

5.3.3 Water Testing of Device Casing

Another key test we performed on the device casing was a water resistance test. We followed the ASTM Standard procedure for water resistance testing for this procedure. For the set up, we constructed a rectangular prism, roughly equal to the size of the device's circuit component, out of paper to simulate the device, and show wetness more easily. We also placed foam over the holes on the interior of the box, but did not secure them down. They were held in place by the paper circuit model, as they would be in the actual design.

The first portion of the test we conducted was to test water resistance through with splashing. The purpose of this was to simulate a dog drinking from a water bowl. In this test, we held the case approximately 2 inches away from the water bowl. We then splashed the case repeatedly with increasing vigor. We were sure that both holes and the entire front surface were reached. Upon examining the case after this first test, both the inside and outside of the clear top were wet. The foam at the holes showed dampness through half the thickness, with no dampness reaching the paper. The paper prism showed no signs of water damage. From this portion of the test, we asserted that the case was water resistant, barring extreme circumstances. The limiting factor to this would be the microphone, which is directly exposed. To get the microphone wet to the point of functionality loss would require intentional efforts, and would be unlikely to occur. Due to the placement of the microphone on the device, a dog drinking water, even messily, would likely not affect the microphone.

The second part of this test that we conducted was an immersion test to measure if the case was waterproof. The purpose of this is to simulate the worst-case scenario of the device being dropped in water. For this test, we held the device approximately 2 inches above the water bowl and dropped it. The case was dropped and retrieved immediately, with approximately 1 second full immersion. Upon opening the case, we examined that there was a small amount of water visible in the casing. The paper prism showed wetness on the bottom portion, but maintained its shape. From this test, we concluded that the device is not waterproof, and should not be submerged in water for any period of time.

Chapter 6. Final Design

6.1 Final Design

The final iteration of this design is comprised of a battery pack containing 3 x AAA batteries, 7-segment display, Teensy 3.2 Development Board, Flora Neopixel LED, Adafruit Electret Microphone, and a 3D printed custom case. We created a Teensy (Arduino) program that analyzes sound inputs from the microphone, determines if it is within a certain frequency threshold, adds to a total cough count if it is within the threshold, and displays the total cough count. A copy of the final code used to program our device can be found in Appendix D. As described in Section 5.1, we used a MATLAB program to determine our frequency threshold to be 450 - 750 Hz. Our circuit was protected by foam inserts, and secured in a plastic case. The battery wires connecting to the circuit were protected in a heat-shrink seal, and concealed under the mesh harness used to mount the device to the patient. The battery pack and case were both attached to the harness via industrial Velcro, such that both components could be easily removed if needed. The final design of our device is shown in Figure 6.1. The final circuit, case, and battery pack weighed a total of 78.4g. The full assembly consisting of the circuit, casing, and battery pack attached to the harness weighed a total of 170.2g. The harness can be changed based on veterinarian needs, but we selected the size and type based on average dog size of our target patients and stability to hold our device.

Figure 6.1. Final cough counting device mounted on a small dog harness, and being worn by a canine meeting the physical criteria of our targeted patient.

6.2 Economics

We do not expect our device to have a significant economic impact on the medical industry. Our device was designed for veterinarians to use in a clinical setting to help diagnose coughing ailments, and determine if the treatment they prescribe is effective. The total cost of our device was $77.18; however, we anticipate this cost would be greatly reduced if the device was manufactured commercially in larger quantities. Refer to Appendix E for a full Bill of Materials.

6.3 Environmental Impact

The device is environmentally sustainable because it is powered by lithium ion batteries, which are housed in a reusable plastic case. Furthermore, the batteries used can be immersed in salt water for a period of time, allowing the batteries to be fully discharged, and safe to throw away with other household trash [26]. Another benefit of using lithium ion batteries, is that they are not acidic and hazardous to the environment. The plastic casing for the circuit was made from a reusable filament that can be broken down and reused for different 3D printing projects, therefore it will not be thrown away if it becomes outdated [27].

6.4 Societal Influence

The cough counter will make dog owners feel more comfortable and in control because they will be able to see changes in their dog's condition in real time. We expect owners will be more confident in their veterinarian's ability to monitor and assess the state of their pet's health, and see justifications for treatment changes. Our device will help keep owners engaged in the process of their pet's treatment.

We do not envision there to be any political ramifications as a result of our device. We intend for this device to be used at the discretion of each individual veterinarian, and do not expect there to be any political-related concerns.

6.5 Ethical Concerns

The design and testing of devices intended for medical purposes are associated with a variety of ethical concerns. In particular, issues surrounding animal testing can be very sensitive, and have sparked animal activist groups, such as PETA, to boycott products that permit these practices. However, because the device is noninvasive, and functions as a diagnostic tool, we believe the focus of the device testing would be in a clinical setting to benefit the dog, and would not cause ethical concerns regarding animal testing. The main ethical concern to be addressed is how confident the user is with using the device as a reliable diagnostic method. Although technological advancements in the medical industry have greatly improved the diagnosis of a variety of medical ailments, there is still a need for medical professionals in order to monitor the accuracy of the devices. Our device should not be used as the primary diagnostic tool, and should only be implemented in conjunction with professional advice from qualified veterinarians.

6.7 Health and Safety Issues

Our device is not intended for human use, and is intended to be used as a diagnostic tool to help veterinarians treat their canine patients. To address device breakage concerns, we mechanically tested the device to ensure it will not break under the loads expected in a dog's daily activities. As this is a first iteration, our device needs further development to increase the sensitivity and specificity such that the veterinarians can confidently diagnose an ailment or the effectiveness of a treatment. Furthermore, as mentioned in Section 6.6, this device is intended to be used with the supervision of a qualified professional, and should not be used as the only diagnostic instrument owners use to determine their pet's health status.

6.8 Manufacturability

The components of our device circuit were purchased from various distributors of electronic parts. The costs of these items could be decreased by purchasing items in bulk, or outsourcing to another supplier. The harness could also be outsourced to a pet product supplier, or purchased in large quantity to reduce costs. We used additive manufacturing to produce the circuit case. This method allows for one machine to yield thousands of parts, and reduces the costs of manufacturing as there is minimal scrap.

Because this is a medical device, the U.S. Food and Drug Administration's current good manufacturing practices (CGMP's) must be followed to ensure the quality and safety of the device. Regulations regarding CGMP and Quality System (QS) requirements are outlined in the Code of Federal Requirements Title 21 Volume 8 Part 820 (21 CFR part 820). These regulations are applicable to "any device or accessory to any device that is suitable for use or capable of functioning, whether or not it is packaged, labeled, or sterilized" [26]. Although the CGMP and QS requirements are not written for specific manufacturing processes, as they are applied to a wide range of medical devices, they provide guidelines for manufacturing procedures. This framework requires certain details concerning the specifics of the device to be filled in according to current state-of-the-art manufacturing processes for devices of similar nature [28].

Chapter 7. Discussion

7.1 Analyzing Sound Testing Results

A major part of this project was analyzing sound to determine if a cough had occurred. To do this, we first had to determine an appropriate frequency threshold. As explained above, we determined our set threshold to be 450-750 Hz. One of the key issues we ran into was the lack of available data. We expect we could have further narrowed the threshold with more coughing samples, and coughing and barking samples from the same dog. This would have allowed us to more accurately fine-tune our threshold. Using the data we were given from the veterinarians and data collected by our team, we had a total of 21 coughs and 34 barks to test our device on. Using a confusion matrix of these results (from Section 5.1), we calculated the sensitivity and specificity of our device to be 86% and 89% respectively. This shows that our device is fairly accurate, based on our limited testing, particularly for a first iteration or prototype. With more time and data, we would theoretically be able to customize our parameters to a particular type of coughing for a particular breed of dog. Our device, as is, performs above expectations, and can be utilized in treatment, as intended by our clients, to distinguish between coughs and barks, and count coughs in small dog breeds. The limiting factor to this, would be defining the type of coughing. For example, canines suffering from a bronchial collapse have more distinct coughs, while canines suffering from other ailments, may result in a "honking" cough, and therefore produce different results. The "honking" dogs typically make wheezing sounds with each breath which can be very difficult for even humans to differentiate between coughing and breathing. Our device, as it is now, may not be well suited to this ailment, as it would likely count each breath as a cough episode. To counter this, we would need more time and data from our client to further develop a more comprehensive solution. Ideally, we would need thousands of sound samples to accurately analyze the sound profile of coughing. Using just frequency and amplitude to detect a distinct sound and distinguish it from other environmental sounds is not enough. A comprehensive sound library of high quality cough recordings is needed to build a machine learning program that could work similarly to voice recognition software. Based on the small selection of samples we were given, we determined using frequency analysis was the most viable option to base our device off of.

7.2 Analyzing Mechanical Testing Results

For its intended purposes, our device performed very highly according to our mechanical tests. Our compression testing on the Instron showed that our casing can withstand compressive forces well over what it would see in its final application. Based on the maximum expected weight of the target patient, 8lb, we started with 36 N of compressive force. The weakest position in this compressive test was the top face, which was expected due to the use of the clear Lexan material. At 50 N, slight bending was observed in the lid, so the trial was ended to avoid permanent deflection and damage to the piece. The other faces saw slight deflections at 80 N and 120 N, however we expect these values are much higher than what the casing would be expected to be exposed to, and that it could most likely withstand higher forces if tests had been conducted to the point of destruction.

To further test the durability of our device we conducted a series of drop tests. From testing each face and corners of the device up to 6 ft, we determined that the device casing was drop-proof. We tested up to this height to account for accidents, for example if the case were to fall from a high shelf. One concern we would have is damage to the device inside the casing from the impact and internal contact forces. We have mitigated this with foam inserts, and have found no damage to the casing or device as a result. It is important to note that this device and casing assembly should not be intentionally dropped or damaged.

We also conducted water resistance test, with the expectation that our casing and device could see small amounts of water, i.e. splashing from a dog drinking water. From our initial water resistance test and final water immersion test, we determined that our device casing is water resistant, but not waterproof. Given this, it would take a fair amount of splashing and water exposure to cause damage. The limiting factors here are the direct exposure to the microphone and the slight gaps in the top face, where the clear top slides in. It would be possible to cover the microphone and seal the casing, but would need further experimentation as to not block the microphone from picking up sounds. Based on the needs outlined by our client, waterproofing the device is not a major necessity, so we optimized sound quality and device operability over waterproofing.

Chapter 8. Conclusions and Recommendations

8.1 Conclusions

The purpose of this project was to create a device to detect and count the number of times a small canine coughs in a given period of time. We have created a small, portable, durable device suited to fit a 6-8 lb dog, and have worked with veterinarians from Tufts Cummings School of Veterinary Medicine to design the device for use in a clinical setting. Our device utilized a Neopixel LED and 7-segment display to indicate the detection of a cough, and display the total cough count. As a first iteration, we have determined that our device has certain limitations. One of the major obstacles we faced was the number of coughing and barking sound samples. We were given less than 5 usable files, and did not have paired coughs and barks from the same dog. Due to this, it was difficult to see if variation was due to how the sample was collected or actual differences in the barks versus the coughs. Though it is understandable to have some background noise in this setting, and our device will experience this in use, background noise was very difficult to work with during our initial testing phases. Isolated coughs and barks are crucial to determining an accurate frequency threshold, so lacking this could be a significant source of error in our device. To determine more accurate results, we would need many samples of varying types of coughing, and paired barks and coughs from the same dog to rule out variations across species and individuals. Because the frequency threshold can be easily altered, the device has the potential to be tailored to individual patients, which could greatly increase the specificity and sensitivity, therefore improving accuracy and device performance.

According to many of the experts that we consulted throughout our design process, as well as the research we conducted, we determined that sound analysis in general requires a high level of skill and knowledge of a variety of programs. We were limited by our skill level in coding, and experienced many difficulties in accurately finding a coughing threshold. Although we were able to find a general frequency range according to basic analysis of our limited supply of samples, the range is too broad, and the device will pick up any sound that falls into that range. For example, certain human voices have been shown to be picked up when the device is on, which is a major source of error in this project.

8.2 Recommendations

After meeting with a music professor at the end of C term, we discovered that our parameters of amplitude and frequency we're simply too basic to properly distinguish between different noises. We had initially assumed that amplitude determined the strength of sound, but the parameters that control the strength of sound are amplitude, power, and intensity. Upon further investigation, we found that these are quite different from each other even though they are closely and proportionally related. Music machines pick up these different parameters to copy the strength of the sound and replay it with a nearly identical intensity, amplitude, and power that cannot be distinguished by the human ear. We realized that we did not have enough time to thoroughly research and implement these parameters into our project, and had to move on in order to have a deliverable product for Tufts. We would highly recommend that the next group do research on these different parameters, and incorporate them in the code. We believe this may allow for significantly improved cough detect and background noise reduction. In order for the next group to successfully research and implement this project, we also highly recommend the next MQP team be composed of music, ECE, and CS majors.

Based on the advice we received from the music department, we also recommend the next team use the music department's software (MAX), and other music machine technology because the software is specifically designed to handle these types of projects. We would recommend that the next team ask for help from the music department, because the other engineering departments only know some of the basic theories of sound, and very few had experience in this specific field.

Furthermore, this type of project requires an extensive amount of sound data, specifically coughing and barking, which we were severely limited with on this project. We would recommend that the next group develop a sound library of both coughing and barking samples from varying breeds of dogs and coughing ailments. This library should consist of thousands of coughs and barks due to variations of different breeds, and different coughing sounds such as honking, dry coughs, wet coughs, and wheezing. Having a significantly larger selection of data will allow the next group to confidently develop a program in which the sound profile of coughing can be observed for different types of dogs, and additional parameters can be implemented to detect coughs. Our group was not able to develop this method due to a lack of sound data and coding difficulties of the Teensy board used. We believe a program specifically tailored to sound analysis may be able to determine and more specific sound profile for a variety of dog breeds and cough types.

We also recommend future teams consider data storage techniques to develop a time stamp feature for the cough counted. We were limited in our circuit design and knowledge of data storage methods, and therefore constructed a device which displays the coughs in real-time, and is reset every time the device is powered off. Arduinos and Raspberry Pis are common devices similar to Teensy, but have user-friendly designs that make data transfer easy. These devices should be compatible with the music departments different software, and have the potential to connect to the owner's device or the veterinarian's computer via Wi-Fi or Bluetooth. This connection could give a real-time view of the dog's condition with respect to time, and allow the veterinarians to intervene faster if the quantity of the coughing becomes severe. Using this method would also allow for a proper GUI to be created, which would make the device more technologically advanced and user-friendly.

Although using sound analysis is a good foundation, we ascertained that relying on only sound may not provide enough accuracy to detect coughs. We recommend future teams develop a biosensor that can detect expansion of the lungs or muscle contractions without requiring direct skin contact. We believe it could be beneficial to have two sensors that could be passed through a comparator to determine if the sound detected was a cough, or some other environmental noise. Our team considered using a respiration belt which could detect the expansion of the dog's lungs when a cough occurred; however, we were required to build a noninvasive device that did not require constrictive contact. Therefore, we recommend future teams research alternative methods that use a similar approach, but reduce the invasive aspects of current biosensors.

In conclusion, we have determined that sound analysis is a possible approach to detect coughing in canine, but the device we built needs further development from experts in the field of electrical engineering, computer science, and music software. As this was the first iteration of this project, we focused on developing a foundation for future teams to build off of, and identify the major difficulties and impediments associated with this project. Although the device we designed is not yet at market quality, we believe this project has potential to be developed further, and advise the suggestions we have recommended be considered for future designs.

References

[1] S. Chamberlain, S. Birring and R. Garrod, "Nonpharmacological Interventions for Refractory Chronic Cough Patients: Systematic Review," Lung, vol. 192, (1), pp. 75-85, 2014. Available: http://www.ncbi.nlm.nih.gov/pubmed/24121952. DOI: 10.1007/s00408-013-9508-y.

[2] (November 22,). Dog Coughing: Causes and Treatment Options. Available: http://www.akc.org/content/health/articles/dog-coughing/.

[3] Tracheal Collapse. Available: https://www.acvs.org/small-animal/tracheal-collapse.

[4] J. (). Background Information for Sound. Available: https://ingeniumcanada.org/scitech/education/tell-me-about-background-information-for-sound.php

[5] R. K. Martin et al, "Load-carrying effects on the adult beagle tibia," Medicine and Science in Sports and Exercise, vol. 13, (5), pp. 343, 1981. Available: http://www.ncbi.nlm.nih.gov/pubmed/7321835.

[6] Polverino, Mario, Francesca Polverino, Marco Fasolino, Filippo Andò, Antonio Alfieri, and Francesco De Blasio. 2012. Anatomy and neuro-pathophysiology of the cough reflex arc. Multidisciplinary Respiratory Medicine 7(1): 5.

[7] (). Can You Keep That Noise Down. Available :http://www.odec.ca/projects/2007/adam7s2/index_files/Page696.htm

[8] O. Lozada and G. Espinosa, "An improved high speed, and low voltage CMOS current mode logic latch," Analog Integr Circ Sig Process, vol. 90, (1), pp. 247-252, 2017. Available: https://search.proquest.com/docview/1880806598. DOI: 10.1007/s10470-016-0875-1.

[9] S. Matos et al, "Detection of cough signals in continuous audio recordings using hidden Markov models," IEEE Trans. Biomed. Eng., vol. 53, (6), pp. 1078-1083, June 01, 2006.

[10] S. T. Kulnik et al, "Cough frequency monitors: can they discriminate patient from environmental coughs?" Journal of Thoracic Disease, vol. 8, (11), pp. 3152-3159, 2016. . DOI: 10.21037/jtd.2016.11.02.

[11] G. A. MacDonald, "A review of low cost accelerometers for vehicle dynamics," Sensors & Actuators: A. Physical, vol. 21, (1), pp. 303-307, 1990. Available: http://www.sciencedirect.com/science/article/pii/092442479085060H. DOI: 10.1016/0924-4247(90)85060-H.

[12] K. Takeuchi, "Novel Co-Design of NAND Flash Memory and NAND Flash Controller

Circuits for Sub-30 nm Low-Power High-Speed Solid-State Drives (SSD)," Jssc, vol. 44, (4), pp. 1227-1234, 2009. Available: http://ieeexplore.ieee.org/document/4804997. DOI: 10.1109/JSSC.2009.2014027.

[13] Y. Kim et al, "FlashSim: A simulator for NAND flash-based solid-state drives," in 2009, Available: http://ieeexplore.ieee.org/document/5283998. DOI: 10.1109/SIMUL.2009.17.

[14] Y. Kim et al, "FlashSim: A simulator for NAND flash-based solid-state drives," in 2009, Available: http://ieeexplore.ieee.org/document/5283998. DOI: 10.1109/SIMUL.2009.17.

[15] D. l. Corrigan and J. y. Paton, "Pilot study of objective cough monitoring in infants," Pediatr. Pulmonol., vol. 35, (5), pp. 350-357, 2003. Available: http://onlinelibrary.wiley.com/doi/10.1002/ppul.10267/abstract. DOI: 10.1002/ppul.10267.

[16] M. Mlnczak, K. Pariaszewska and G. Cybulski, "Automatic cough episode detection using a vibroacoustic sensor," in 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 2808-2811.

[17] (). MLT1132 Piezo Respiratory Belt Transducer. Available: http://m-cdn.adinstruments.com/product-data-cards/MLT1132-DCW-15A.pdf.

[18] J. Smith. (October). Monitoring chronic cough: current and future techniques. Available: http://dx.doi.org/10.1586/ers.10.63

[19] (). Medical devices -- Recognized essential principles of safety and performance of medical devices. Available: https://www.iso.org/standard/63939.html

[20] (). Medical devices -- Application of risk management to medical devices. Available: https://www.iso.org/standard/38193.html

[21] Created Tony DiCola, "FFT: Fun with Fourier Transforms,".

[22] (). Good2Go Black Mesh Dog Harness, XS, X-Small. Available: https://www.bhg.com/shop/good2go-good2go-black-mesh-dog-harness-xs-x-small-p8e72538b9c131e7e9746723b09aca237.html.

[23]ASTM. “Standard Test Method for Drop Test of Loaded Containers by Free Fall.” ASTM COMPASS. https://compass.astm.org/EDIT/html_annot.cgi?D5276+98(2017)

[24] (). *The Original EcoBark Maximum Comfort & Control Dog Harness 4-65 lbs; No Pull & No Choke Design, Luxurious Padded Vest, Eco-Friendly, For Puppies and Dogs*. Available: www.amazon.com.

[25] (). *Yorkshire Terrier Information*. Available: http://www.dogbreedindex.com/dog/details/yorkshire-terrier.

[26] (). Lithium Ion Batteries: Exploring the Hazards of these Common Rechargeable Batteries.

Available: https://www.americandisposal.com/blog/lithium-ion-batteries
[27] (). Reduce and Reuse: ExtrusionBot's 'Cruncher' Recycles and Pelletizes 3D Printing Filament Waste. Available: https://3dprint.com/42595/extrusionbot-cruncher-recycle/
[28] (03/27/). *Quality System (QS) Regulation/Medical Device Good Manufacturing Practices*. Available: https://www.fda.gov.

Appendix A

Signal Processing Code to Plot Spectrograms - Developed in MATLAB R2017a

```
myDir = uigetdir; %gets directory with sound files in it
myFiles = dir(fullfile(myDir,'*.wav')); %gets all wav files in directory
for k = 1:length(myFiles)
  baseFileName = myFiles(k).name;
  fullFileName = fullfile(myDir, baseFileName);
  [x, fs] = audioread(fullFileName);
  spectrogram(x, 1024, 3/4*1024, [], fs, 'yaxis')
  box on
  xlabel('Time, ms')
  ylabel('Frequency, kHz')
  title('Spectrogram of the signal')

  h = colorbar;
  ylabel(h, 'Magnitude, dB')
  hold
  drawnow;
  pause(30);
end
```

Appendix B

Signal Processing Code to Plot FFT Frequency Domains of Coughs - Developed in MATLAB R2017A

Individual Plots:

```
myDir = uigetdir; %gets directory with sound files in it
myFiles = dir(fullfile(myDir,'*.wav')); %gets all wav files in directory
for k = 1:length(myFiles)
  baseFileName = myFiles(k).name;
  fullFileName = fullfile(myDir, baseFileName);
  [x, fs] = audioread(fullFileName);
  n = length(x) - 1;
  f=0:fs/n:fs;
  wavefft=abs(fft(x));
  figure(2);
  plot(f,wavefft);
  xlim([0,1000]);
  xlabel('Frequency in Hz');
  ylabel('Magnitude');
  title('The Wave FFT');
  drawnow;
  pause(5);
End
```

All Plots:

```
myDir = uigetdir; %gets directory with sound files in it
myFiles = dir(fullfile(myDir,'*.wav')); %gets all wav files in directory
for k = 1:length(myFiles)
  baseFileName = myFiles(k).name;
  fullFileName = fullfile(myDir, baseFileName);
  [x, fs] = audioread(fullFileName);
  n = length(x) - 1;
```

```
  f=0:fs/n:fs;
  wavefft=abs(fft(x));
  figure(2);
  hold all
  plot(f,wavefft);
  xlim([0,1000]);
  xlabel('Frequency in Hz');
  ylabel('Magnitude');
  title('The Wave FFT');
  drawnow;
end
```

Appendix C

Drop Test Results

Drop Test		No overall damage: case is resistant to wear
Height: (feet)	Orientation:	Results:
1.5	Corner	No wear shown
1.5	Edge	No wear shown
1.5	Large Side	No wear shown
1.5	Small Side	No wear shown
1.5	Base	No wear shown
1.5	Top	No wear shown
3	Corner	No wear shown
3	Edge	No wear shown
3	Large Side	No wear shown
3	Small Side	No wear shown
3	Base	No wear shown
3	Top	No wear shown
4.5	Corner	No wear shown
4.5	Edge	No wear shown
4.5	Large Side	No wear shown
4.5	Small Side	No wear shown
4.5	Base	No wear shown
4.5	Top	No wear shown
6	Corner	No wear shown
6	Edge	No wear shown
6	Large Side	No wear shown
6	Small Side	No wear shown
6	Base	No wear shown
6	Top	No wear shown

Appendix D

Code Programmed in the Device Circuit

```
// Audio Tone Input
// Adapted from code by Tony DiCola (tony@tonydicola.com) which is part of the guide at
http://learn.adafruit.com/fft-fun-with-fourier-transforms/
// Code written for WPI MQP Cough Counter for Canines

#define ARM_MATH_CM4
#include <arm_math.h>
#include <Adafruit_NeoPixel.h>

////////////////////////////////////////////////////////////////////////////////
#include <SoftwareSerial.h>

// These are the Arduino pins required to create a software seiral
//  instance. We'll actually only use the TX pin.
const int softwareTx = 8;
const int softwareRx = 7;

SoftwareSerial s7s(softwareRx, softwareTx);

unsigned int counter = 1;  // This variable will count up to
65k//////////////////////////////////////////////////////////////////////////
char tempString[10];  // Will be used with sprintf to create strings

////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////////////////////
// CONIFIGURATION
// These values can be changed to alter the behavior of the spectrum display.
////////////////////////////////////////////////////////////////////////////
```

```
int SAMPLE_RATE_HZ = 4000;            // Sample rate of the audio in hertz.
const int TONE_LOWS[] = {             // Lower bound (in hz) of each tone in the input sequence.
  450
};
const int TONE_HIGHS[] = {            // Upper bound (in hz) of each tone in the input sequence.
  750
};
int TONE_ERROR_MARGIN_HZ = 10;        // Allowed fudge factor above and below the
bounds for each tone input.
int TONE_WINDOW_MS = 4000;            // Maximum amount of milliseconds allowed to enter
the full sequence.
float TONE_THRESHOLD_DB = 10.0;       // Threshold (in decibels) each tone must be above
other frequencies to count.
const int FFT_SIZE = 256;             // Size of the FFT.  Realistically can only be at most 256
                                      // without running out of memory for buffers and other state.
const int AUDIO_INPUT_PIN = 14;       // Input ADC pin for audio data.
const int ANALOG_READ_RESOLUTION = 10; // Bits of resolution for the ADC.
const int ANALOG_READ_AVERAGING = 16;  // Number of samples to average with each
ADC reading.
const int POWER_LED_PIN = 13;         // Output pin for power LED (pin 13 to use Teensy 3.0's
onboard LED).
const int NEO_PIXEL_PIN = 3;          // Output pin for neo pixels.
const int NEO_PIXEL_COUNT = 4;        // Number of neo pixels.  You should be able to
increase this without
                                      // any other changes to the program.
const int MAX_CHARS = 65;             // Max size of the input command buffer


////////////////////////////////////////////////////////////////////////////////
// INTERNAL STATE
// These shouldn't be modified unless you know what you're doing.
```

```
////////////////////////////////////////////////////////////////////////////////

IntervalTimer samplingTimer;
float samples[FFT_SIZE*2];
float magnitudes[FFT_SIZE];
int sampleCounter = 0;
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NEO_PIXEL_COUNT, NEO_PIXEL_PIN,
NEO_GRB + NEO_KHZ800);
char commandBuffer[MAX_CHARS];
int tonePosition = 0;
unsigned long toneStart = 0;


////////////////////////////////////////////////////////////////////////////////
// MAIN SKETCH FUNCTIONS
////////////////////////////////////////////////////////////////////////////////

void setup()
  {
  // Set up serial port.
  Serial.begin(38400);

  // Set up ADC and audio input.
  pinMode(AUDIO_INPUT_PIN, INPUT);
  analogReadResolution(ANALOG_READ_RESOLUTION);
  analogReadAveraging(ANALOG_READ_AVERAGING);

  // Turn on the power indicator LED.
  pinMode(POWER_LED_PIN, OUTPUT);
  digitalWrite(POWER_LED_PIN, HIGH);
```

```
  // Initialize neo pixel library and turn off the LEDs
  pixels.begin();
  pixels.show();

  // Clear the input command buffer
  memset(commandBuffer, 0, sizeof(commandBuffer));

  // Begin sampling audio
  samplingBegin();

  ////////////////////////////////////////////////////////////////////////////////
  // Must begin s7s software serial at the correct baud rate.
  //  The default of the s7s is 9600.
  s7s.begin(9600);

  // Clear the display, and then turn on all segments and decimals
  clearDisplay();  // Clears display, resets cursor
  s7s.print("-ON-");  // Displays -ON- on all digits

  setDecimals(0b111111);  // Turn on all decimals, colon, apos
//
//  // Flash brightness values at the beginning
  setBrightness(0);  // Lowest brightness
  delay(1500);
  setBrightness(127);  // Medium brightness
  delay(1500);
  setBrightness(255);  // High brightness
  delay(1500);

  // Clear the display before jumping into loop
  clearDisplay();
```

```
  //////////////////////////////////////////////////////////////////////////////
}

void loop() {

    ////////////////////////////////////////////////////////////////////////////////
   // Magical sprintf creates a string for us to send to the s7s.
  //  The %4d option creates a 4-digit integer.
  //sprintf(tempString, "%4d", counter);

  // This will output the tempString to the S7S
  //s7s.print(tempString);

  //counter++;  // Increment the counter
  //delay(100);  // This will make the display update at 10Hz.
  ////////////////////////////////////////////////////////////////////////////////

  // Calculate FFT if a full sample is available.
  if (samplingIsDone()) {
    // Run FFT on sample data.
    arm_cfft_radix4_instance_f32 fft_inst;
    arm_cfft_radix4_init_f32(&fft_inst, FFT_SIZE, 0, 1);
    arm_cfft_radix4_f32(&fft_inst, samples);
    // Calculate magnitude of complex numbers output by the FFT.
    arm_cmplx_mag_f32(samples, magnitudes, FFT_SIZE);

    // Detect tone sequence.
    toneLoop();

    // Restart audio sampling.
    samplingBegin();
```

```
  }

  // Parse any pending commands.
  parserLoop();

}

////////////////////////////////////////////////////////////////////////////////
// UTILITY FUNCTIONS
////////////////////////////////////////////////////////////////////////////////

// Compute the average magnitude of a target frequency window vs. all other frequencies.
void windowMean(float* magnitudes, int lowBin, int highBin, float* windowMean, float*
otherMean) {
    *windowMean = 0;
    *otherMean = 0;
    // Notice the first magnitude bin is skipped because it represents the
    // average power of the signal.
    for (int i = 1; i < FFT_SIZE/2; ++i) {
      if (i >= lowBin && i <= highBin) {
        *windowMean += magnitudes[i];
      }
      else {
        *otherMean += magnitudes[i];
      }
    }
    *windowMean /= (highBin - lowBin) + 1;
    *otherMean /= (FFT_SIZE / 2 - (highBin - lowBin));
}

// Convert a frequency to the appropriate FFT bin it will fall within.
```

```
int frequencyToBin(float frequency) {
  float binFrequency = float(SAMPLE_RATE_HZ) / float(FFT_SIZE);
  return int(frequency / binFrequency);
}

// Convert intensity to decibels
float intensityDb(float intensity) {
  return 20.0*log10(intensity);
}


////////////////////////////////////////////////////////////////////////////////
// SPECTRUM DISPLAY FUNCTIONS
///////////////////////////////////////////////////////////////////////////////

void toneLoop() {
  // Calculate the low and high frequency bins for the currently expected tone.
  int lowBin = frequencyToBin(TONE_LOWS[tonePosition] - TONE_ERROR_MARGIN_HZ);
  int highBin = frequencyToBin(TONE_HIGHS[tonePosition] +
TONE_ERROR_MARGIN_HZ);
  // Get the average intensity of frequencies inside and outside the tone window.
  float window, other;
  windowMean(magnitudes, lowBin, highBin, &window, &other);
  window = intensityDb(window);
  other = intensityDb(other);
  // Check if tone intensity is above the threshold to detect a step in the sequence.
  if ((window - other) >= TONE_THRESHOLD_DB) {
    // Start timing the window if this is the first in the sequence.
    unsigned long time = millis();
    if (tonePosition == 0) {
      toneStart = time;
```

```
    }
    // Increment key position if still within the window of key input time.
    if (toneStart + TONE_WINDOW_MS > time) {
      tonePosition += 1;
    }
    else {
      // Outside the window of key input time, reset back to the beginning key.
      tonePosition = 0;
    }
  }
  // Check if the entire sequence was passed through.
  if (tonePosition >= sizeof(TONE_LOWS)/sizeof(int)) {
    toneDetected();
    tonePosition = 0;
  }
}

void toneDetected() {
  // Flash the LEDs one time.
  int pause = 250;
  for (int i = 0; i < 1; ++i) {
    for (int j = 0; j < NEO_PIXEL_COUNT; ++j) {
      pixels.setPixelColor(j, pixels.Color(255, 0, 0));
    }
    pixels.show();
    delay(pause);
    for (int j = 0; j < NEO_PIXEL_COUNT; ++j) {
      pixels.setPixelColor(j, 0);
    }
    pixels.show();
    delay(pause);
```

```
    //////////////////////////////////////////////////////////////////////////////////
      // Magical sprintf creates a string for us to send to the s7s.
  //  The %4d option creates a 4-digit integer.
    sprintf(tempString, "%4d", counter);

  // This will output the tempString to the S7S
    s7s.print(tempString);

    counter++;  // Increment the counter
    delay(100);  // This will make the display update at 10Hz.
    //////////////////////////////////////////////////////////////////////////////////
  }
}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Send the clear display command (0x76)
//  This will clear the display and reset the cursor
void clearDisplay()
{
  s7s.write(0x76);  // Clear display command
}

// Set the displays brightness. Should receive byte with the value
//  to set the brightness to
//  dimmest------------->brightest
//     0--------127--------255
void setBrightness(byte value)
{
  s7s.write(0x7A);  // Set brightness command byte
  s7s.write(value);  // brightness data byte
```

```
}

// Turn on any, none, or all of the decimals.
//  The six lowest bits in the decimals parameter sets a decimal
//  (or colon, or apostrophe) on or off. A 1 indicates on, 0 off.
//  [MSB] (X)(X)(Apos)(Colon)(Digit 4)(Digit 3)(Digit2)(Digit1)
void setDecimals(byte decimals)
{
  s7s.write(0x77);
  s7s.write(decimals);
}
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////

////////////////////////////////////////////////////////////////////////////////
// SAMPLING FUNCTIONS
////////////////////////////////////////////////////////////////////////////////

void samplingCallback() {
  // Read from the ADC and store the sample data
  samples[sampleCounter] = (float32_t)analogRead(AUDIO_INPUT_PIN);
  // Complex FFT functions require a coefficient for the imaginary part of the input.
  // Since we only have real data, set this coefficient to zero.
  samples[sampleCounter+1] = 0.0;
  // Update sample buffer position and stop after the buffer is filled
  sampleCounter += 2;
  if (sampleCounter >= FFT_SIZE*2) {
    samplingTimer.end();
  }
}
```

```
void samplingBegin() {
  // Reset sample buffer position and start callback at necessary rate.
  sampleCounter = 0;
  samplingTimer.begin(samplingCallback, 1000000/SAMPLE_RATE_HZ);
}

boolean samplingIsDone() {
  return sampleCounter >= FFT_SIZE*2;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void parserLoop() {
  // Process any incoming characters from the serial port
  while (Serial.available() > 0) {
    char c = Serial.read();
    // Add any characters that aren't the end of a command (semicolon) to the input buffer.
    if (c != ';') {
      c = toupper(c);
      strncat(commandBuffer, &c, 1);
    }
    else
    {
      // Parse the command because an end of command token was encountered.
      parseCommand(commandBuffer);
      // Clear the input buffer
      memset(commandBuffer, 0, sizeof(commandBuffer));
    }
  }
}
```

```
// Macro used in parseCommand function to simplify parsing get and set commands for a
variable
#define GET_AND_SET(variableName) \
  else if (strcmp(command, "GET " #variableName) == 0) { \
    Serial.println(variableName); \
  } \
  else if (strstr(command, "SET " #variableName " ") != NULL) { \
    variableName = (typeof(variableName)) atof(command+(sizeof("SET " #variableName " ")-
1)); \
  }

void parseCommand(char* command) {
  if (strcmp(command, "GET MAGNITUDES") == 0) {
    for (int i = 0; i < FFT_SIZE; ++i) {
      Serial.println(magnitudes[i]);
    }
  }
  else if (strcmp(command, "GET SAMPLES") == 0) {
    for (int i = 0; i < FFT_SIZE*2; i+=2) {
      Serial.println(samples[i]);
    }
  }
  else if (strcmp(command, "GET FFT_SIZE") == 0) {
    Serial.println(FFT_SIZE);
  }
  GET_AND_SET(SAMPLE_RATE_HZ)
  GET_AND_SET(TONE_ERROR_MARGIN_HZ)
  GET_AND_SET(TONE_WINDOW_MS)
  GET_AND_SET(TONE_THRESHOLD_DB)
}
```

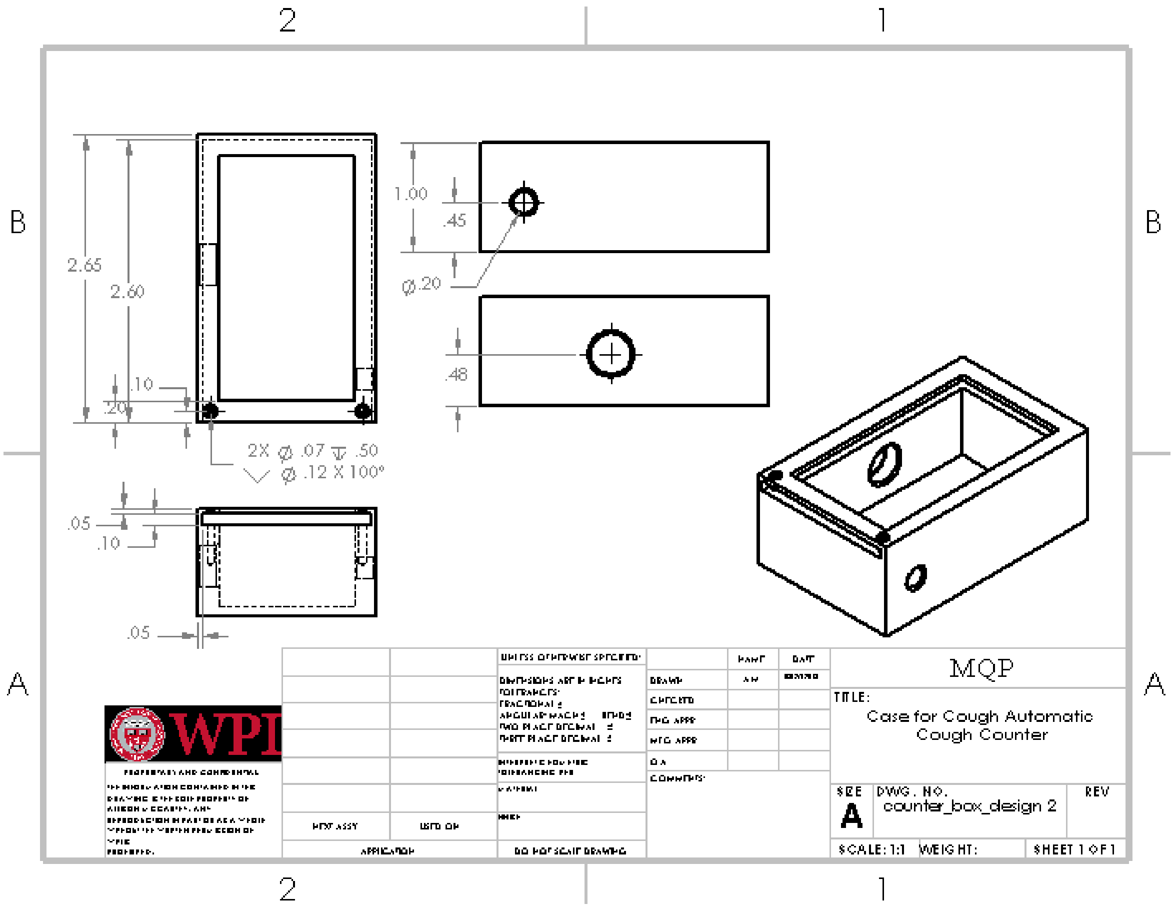
Appendix E

Bill of Materials for Final Device Design

Item	**Cost (USD)**
Teensy 3.2 Development Board	19.95
Adafruit Microphone	6.95
SparkFun 7-segment Display	12.95
Flora RGB Neopixel LED (pack of 4)	7.95
3 x AAA Battery Pack	1.95
Custom Printed Case	8.46
LEXAN Plastic Sheet 10 in x 8 in	4.98
Mesh Dog Harness	13.99
Total	77.18

Appendix F

Solidworks Drawings of Cough Counter Plastic Case



Appendix G

Solidworks Drawings of Cough Counter Plastic Case Lid