Pet Pal: The Smart Pet Collar

Andres Graterol, Zachary Kirksey, Hanh Le, and Hunter Miller

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This paper presents a new form of smart technology in Pet Pal. Through the use of a variety of modern technologies, a new iteration of the classic pet collar was produced. This solution uses hardware components as well as an easily accessible mobile application to allow for pet owners to have peace of mind. The aim for this project was to both to improve quality of life as well as improve upon existing designs for this particular technology. This presented a unique challenge in making full use of electrical and computer knowledge to create a well functioning, yet still simple pet collar.**

I. Introduction

Developed with pet owners in mind, this project aims to bring a sense of security to individuals looking to keep a watchful eye on their pet's health. A physical collar will be responsible for getting readings such as heart rate, temperature, GPS location, outputting sound from the speaker, as well as making a bluetooth connection with the corresponding companion application we are developing on Android Studio. This companion application will be responsible for displaying the information outputted by the collar.

II. System Components

A. Microcontroller

The microcontroller we decided to use to develop this project is the ATmega328P. Specifically the 28-pin narrow dual in-line package (DIP-28N). We chose this microcontroller specifically because it has a Two-Wire serial interface (TWI) port, an SPI serial port, an I2C port, serial programmable USART, and 23 general purpose I/O lines. In addition, this microcontroller can easily be removed from the board via a 28-pin holder to allow for ease of access to the microcontroller. This number of pins and communication protocols are necessary for our project so that all of the peripherals necessary can effectively communicate with the microcontroller.

B. Vitality Sensors

The vitality sensors on this project are meant to read the pet's vitals upon commands issued by the companion application. They are made up of a MAX30102 pulse oximeter, which contains an internal temperature sensor, and a heart rate sensor. This pulse oximeter operates at 3.3V and communicates with the microcontroller through I2C.

C. GPS Module

For the GPS module our team chose to maintain an integrated circuit (IC) known as the Flora Wearable GPS. It has copper pads that allow for conductive threading to be looped through them. The use of conductive threading allows the IC to be threaded directly into the collar while still maintaining the flow of current. Although threading the IC directly to the collar was not the final method we chose for connecting it to the rest of our system, it was still advantageous to have this option during the planning stages of our project. Having the IC be external to the PCB allowed for more flexibility in terms of placement and also saved space on the PCB, allowing us to keep our slim design. An external coin cell battery is threaded to the IC to improve GPS start up time and help the GPS module remember which satellite it had a fix with.

D. Bluetooth Module

The bluetooth module acts as the API between the mobile application and the microcontroller. This MDBT40 bluetooth module is compatible with both bluetooth versions 4.0 and 4.1. The bluetooth communicates with the microcontroller and all the peripherals using the software USART characteristic developed by Nordic Semiconductors. On the hardware side, the bluetooth is backed by the SPI communication protocol.

E. Speakers

The speaker is responsible for outputting tones as commanded by the mobile application. The button used to send this command to the microcontroller is found on the locations page, as this functionality is mainly used to assist in the locating of the pet. The speaker used in the system is the Visaton K15S. This speaker was chosen since it is 1.5 cm in diameter and would fit our compact design. Despite its relatively small size, with the help of the audio amplifier circuit it can output up to 70 dB. At this rating, the speaker is not loud enough to hurt an animal's ears, but yet loud enough to be heard by someone in close vicinity. This speaker communicates to the microcontroller using PWM.

F. *Battery*

Beside trying to achieve the expected life span of the battery, one of the most important factors that we consider when we are choosing the battery is the size of it. The energizer L522 weighs 33.9 grams and its volume is 21.4 cubic centimeters. Although this is not a perfect size for a small system, it is good enough to use since it can provide a 9 volt output.

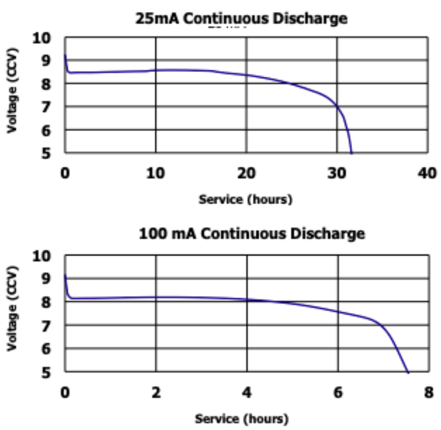

Fig. 1. Typical Discharge Performance of the Energizer L522

Our system will drain a current between 25 milliamps and 50 milliamps, for that reason, by using the Energizer L522, our system can last around 20 hours.

G. *Audio Circuit*

The main purpose of the audio circuit is to improve the efficiency of the speaker that we have chosen. When the speaker is connected directly to the audio signal, it can only produce a sound quality of 39 dB. Thus, an audio circuit is necessary to increase the speaker's efficiency from 39 dB to at least 50 dB. The audio amplifier used in our system is the LM386, which is one of the most common amplifiers used for building an audio system. The LM386 requires a DC input voltage between 7V to 12V. It is also small and powerful enough in order to ensure that our system achieves our desired dimensions.

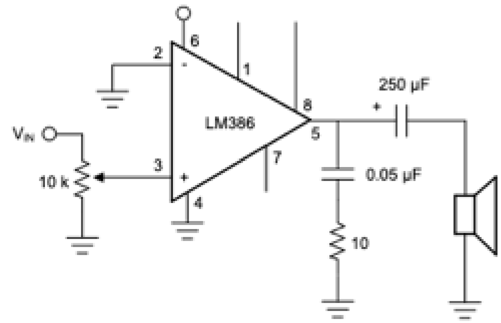

Fig. 2. The first schematic of the audio circuit

In the very first schematic, our group was simply trying to create an audio circuit that can increase the efficiency of the speaker. We barely paid attention to the sound quality, for that reason, there are only two capacitors used in the original circuit. In addition, there is also a volume controller in the first design. We realized that adding a volume controller will expand the size of the final product. Thus, we decided to remove the volume controller as well as add more capacitors in order to have a better sound quality.

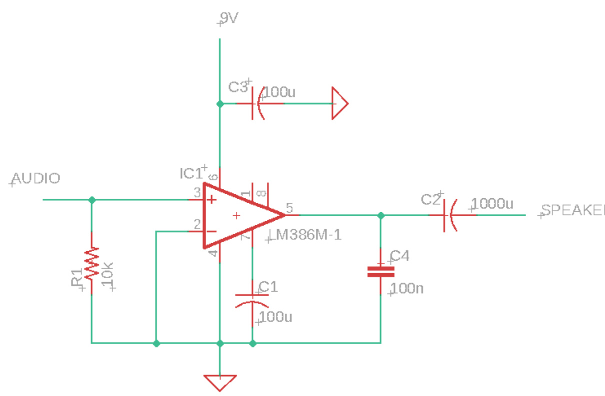

Fig. 3. The final design of the audio circuit.

In the final audio design, we still use the LM386 audio amplifier which has 8 pins. Pins 1 and 8 are the gain setting pins. Since we want to use the original gain of the LM386 amplifier, we will leave pins 1 and 8 open. Moreover, there are four capacitors used in the design. These capacitors will fill out the noise from both the amplifier and the audio signal so that we can have a much better sound quality.

H. *Power Circuit*

The power circuit will produce 3.3 volts and 5 volts which are used to supply power to the microcontroller and the components. The power circuit includes a polarity protection system, two different voltage regulators, and the audio circuit also plays a part in the power circuit. A battery of 9 volts will be used as an input voltage. This battery will be connected to the polarity protection system and from there, it supplies power to the voltage regulators and the audio amplifier.

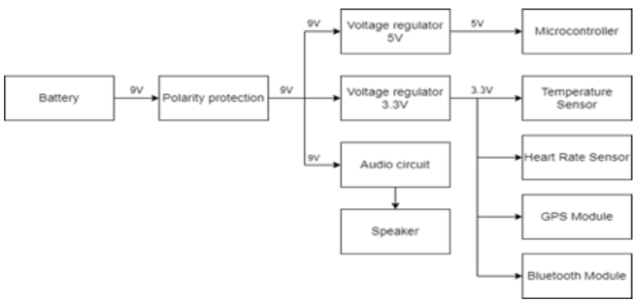

Fig. 4. Power Circuit Flowchart.

I. *Polarity Protection*

Using a replaceable battery can cause reverse polarity. Thus, a polarity protection system is included in our design. Using a P-channel MOSFET is one of the most efficient solutions which are used to prevent the reverse polarity. A battery of 9 volts will be connected to the drain terminal of the IRLML6402PBF P-channel MOSFET which has a low static drain to source on resistance and a threshold voltage between -0.4 volt and -1.2 volts.

J. *Voltage Regulator*

There are two different linear voltage regulators used in our design. The first voltage regulator is LM2936Z-3.3 and the linear voltage regulator is LM2936Z-5.

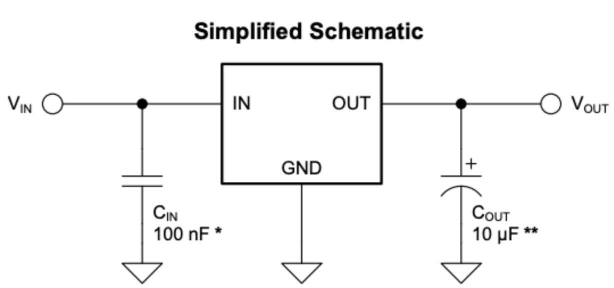

Fig. 5. Simplified Schematic of the LM2936 Linear Voltage Regulator.

The input voltage, Vin, of the voltage regulator is a DC voltage and the output voltages, Vout, are also DC voltages of 3.3 volts and 5 volts. We believe a linear voltage regulator with a low quiescent current such as the LM2936 will satisfy this simple requirement.

There are multiple components in our system including a heart rate and temperature sensor, a GPS module, a bluetooth module, and a speaker. All of the parts mentioned, except for the speaker, need an input of 3.3 volts in order to correctly operate. Thus, we use the LM2936Z-3.3 to create a 3.3 volts output. Similarly, the LM2936Z-5 is used to provide a 5 volts output which is required to run the microcontroller.

K. *Mobile Application*

The mobile application allows the user to control and display information from the Pet Pal collar system. It is developed for use on Android systems, with a requirement of an internet connection to register and login. We believe that creating a mobile app is the most convenient way to allow the user to interact with the collar, and we have made sure to be simplistic in our design for ease of use. The application communicates with the collar through the Bluetooth module, which is done by a simple press of a button. For functionalities other than the speaker, the Bluetooth module will then respond with the information requested by the user.

III. Collar System Overview

This pet collar system can be split into two main components: the hardware and software components. In this section, we will be discussing the physical collar system.

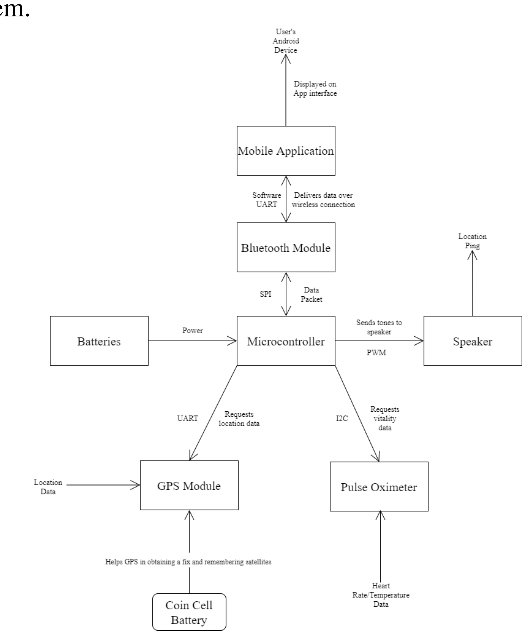

Fig. 6. Hardware Block Diagram shows how the hardware interacts with each other as well as the mobile application.

This hardware block diagram mainly depicts how the ATmega328P is going to interact with all of the peripherals. Next to the data line we show which communication protocol is being used to communicate with the stated peripherals. Knowing which communication protocols we needed to use was crucial to the success of this system, as the microcontroller has limited ports dedicated to communicating with each protocol. The only issue we came into contact with was the need to use multiple UARTs (one for the GPS module and one for the communication between the app and the bluetooth module). This is an issue because the ATmega328P only has one dedicated hardware UART port. We managed to sidestep this by implementing a software UART to communicate with the app and the bluetooth. This software implementation took up more space on the microcontroller's RAM, but we still had enough memory left over so as to not cause any stability issues.

Also worth mentioning is that if the speaker was plugged directly into the microcontroller's PWM pin, then it would not only produce a tone too quiet for our requirement specifications, but would also draw too much current from the microcontroller, possibly causing damage to the PWM pin. To counteract these pitfalls, we connected the microcontroller's PWM pin to an audio circuit which both provides the proper resistance to avoid damaging the pin and also acts as an amplifier so we achieve the proper decibel range.

Notice the coin cell battery that is connected to the GPS module. This coin cell battery is not for use as a power supply to the module, instead it is used to assist in acquiring a satellite fix. Without this coin cell battery, the GPS would not be able to get a satellite reading indoors as it would require the module to be in view of the sky. In addition, the coin cell battery also helps the GPS module remember which satellite it connected to last. This allows for faster start-up times for the GPS module.

IV. Hardware Details

A. *Printed Circuit Board*

The PCB is responsible for housing a majority of the peripheral sensors as well as the Microcontroller. The major constraints in the design process were size limitations and supply chain issues.

The width of the collar that was chosen for the design was one inch, as this was the largest that could be found on the market. In order for the system to maintain reasonable dimensions, the width of the PCB could not too far exceed the width of the collar. A similar problem arose for the length of the PCB as well. Too long of a board inside the casing would make it difficult for the sensors to sit closely enough to the skin to get accurate measurements. These limitations did not allow for large amounts of room to fit all of the components on the board. For these reasons, certain components were chosen to remain off board due to their size. These being the speaker, the battery, and the GPS module. With these constraints in mind, the final size of the PCB is 1.7 inches wide and 2.5 inches long.

Many manufacturers and, subsequently, distributors have been experiencing massive supply chain issues. This is due to a variety of factors. These being the recent pandemic and demand for electrical components being on the rise. During the course of this project, there were multiple components which were out of stock or had exorbitantly long lead times for all major distributors so alternatives were found or the package was changed to accommodate. As a result, multiple iterations of the board were needed to reflect the ever changing circumstances surrounding availability of components.

B. *MAX30102 Temperature and Heart Rate*

As stated previously, the MAX30102 module will be responsible for getting the temperature and heart rate data for our system.

The MAX30102 has an on-chip temperature sensor with an inherent resolution of 0.0625°C [3]. In order to prevent an inaccurate reading to the temperature sensor, we have to make sure to turn off the red on-board LED that is mainly used for heart rate readings. Turning off the LED prevents heat dissipation from affecting the temperature reading. The temperature sensor also has a range from -40°C to 85°C which will comfortably fit the needs of our system.

When the module is in Heart Rate mode, the red LED is turned on in order to capture optical data and determine the pet's heart rate through the process of photoplethysmogram (PPG).

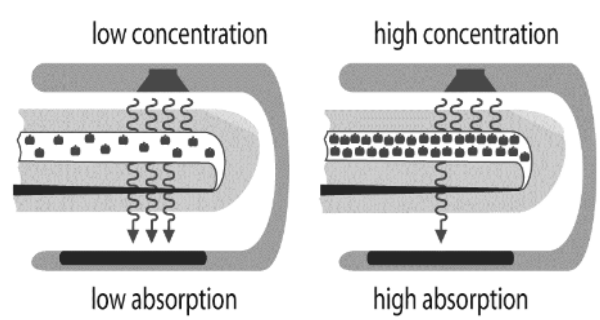

Fig. 7. A demonstration on photoplethysmography

PPG works by sending a light through the skin of the user. The red blood cells reflect the light back to the sensor which gets a different reading depending on the level of absorption that happens inside the skin of the user.

C. *PA6H GPS Module*

This module is what is located on the Flora GPS IC. It is a small low-power GPS module with built-in data-logging capability. It boasts -165 dBm sensitivity, 10 Hz updates, and 99 search channels [1]. It also has built-in datalogging and is RTC battery-compatible. In our system we are sewing (through the use of conductive threading) on an external coin cell battery not only to achieve real time clock capabilities but also to help the module obtain a satellite fix. More information on this can be found in section II of this paper.

In addition to all of this, the IC also has a u.FL connector to allow connection to an external active antenna which would boost the sensitivity of the module. In our design, we chose to opt against using an external antenna as fitting an external antenna inside of the casing would prove to be difficult. The extra sensitivity is not necessary for our design as the GPS signals can easily travel through the filament of the casing.

D. *nRF51822 BLE Module*

The MDBT40 is a Bluetooth low energy (BLE) module design based on the nRF51822 Nordic Solution. This solution incorporates GPIO, UART, I2C, and ADC interfaces for connecting peripherals. In our case, we are using a UART characteristic to communicate between the BLE module and the rest of the sensors. The MDBT40 series works off Bluetooth versions 4.0 and 4.1 [4].

To prototype the Bluetooth functionality on the application, we used the Bluefruit SPI Friend IC and the Bluefruit Connect Android Application both developed by Adafruit industries.

E. *Visaton K15S Speaker*

This speaker's small dimensions are the main reason why we chose to incorporate it into our design. It boasts a diameter of 1.5 cm and a height of 0.4 cm [2]. Despite its relatively small design, it is effective in reaching the decibel range that we set in the requirement specifications with the assistance of the audio circuit we have implemented. This speaker is also waterproof, so if any rain were to enter the casing's speaker holes, the speaker should continue to work without interference.

V. Software Details

The software portion of our project is split into two parts: the mobile application and the embedded programming. The embedded programming controls the data within our PCB, which involves the interactions between the Bluetooth module and each of the sensors. In addition, this code specifies how the Bluetooth module interacts with the mobile application. Meanwhile, the mobile application requests and displays this data.

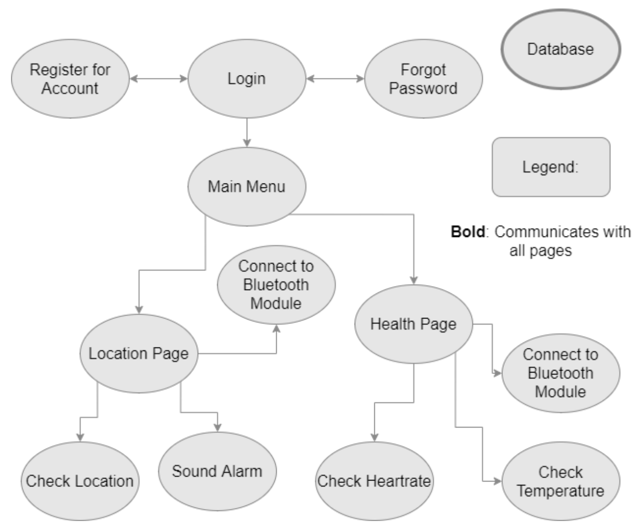

Fig. 8. Software Block Diagram shows all the pages on the mobile application as well as the flow between them.

A. *Mobile Application Development*

The mobile application is developed using the MERN stack: MongoDB, Express, React Native, and Node.js. MongoDB is a no-SQL database, which allows us to store abstract data easier than other databases. Express and Node.js work together to act as the API between the frontend and the database. Finally, React Native is the framework for our frontend, which allows for additional flexibility to create a clean user interface. In addition to the flexibility with this stack, we avoided a large learning curve since we have developed an Android app with these components.

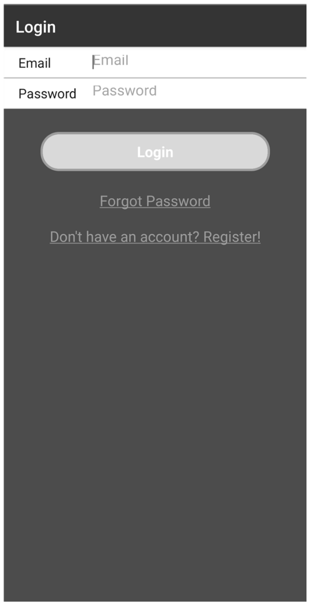

Fig. 9. Screenshot of the login page for the Pet Pal Companion Application.

The main functionalities of our application include retrieving temperature readings in Fahrenheit, heart rate measurements in beats per minute (BPM), and GPS location using latitude and longitude. In addition, the app is able to send a signal to the microcontroller to play an alarm. Specifically, the health readings are returned as a string, along with their unit of measurement, and displayed to the user. Meanwhile, the location is returned as a latitude and longitude string, which can then be copied and pasted into any maps application, such as Google Maps, to get an accurate location. In addition to communicating with the Bluetooth module, our application offers user customization for their Pet Pal Companion account. For example, a user can customize their name, which is displayed in various places across the application. Also, the user can securely verify their account and change their password using a confirmation code sent to their email address. A few screenshots of the app can be seen in Figures 9-11.

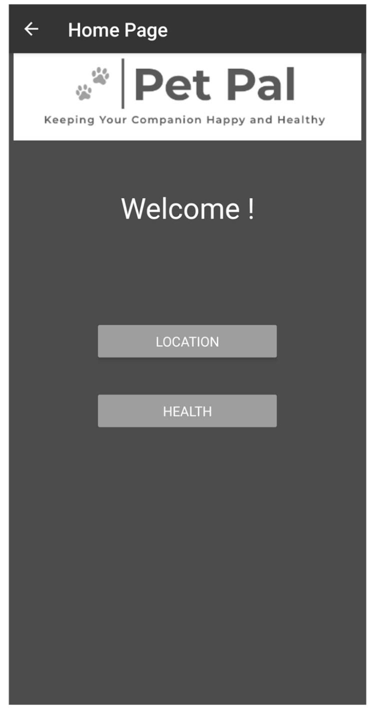

Fig. 10. Screenshot of the home page for the Pet Pal Companion Application.

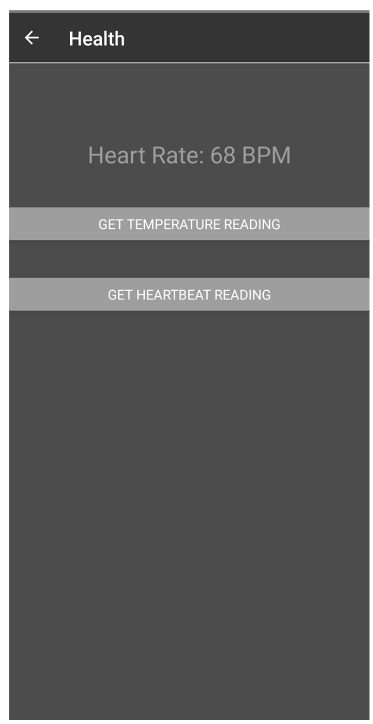

Fig. 11. Screenshot of example data being received by the Pet Pal Companion Application from the physical collar system.

B. Embedded Programming

For our system, Embedded programming is done using the Arduino IDE for our Arduino Uno microcontroller. This controls the flow of data in our Pet Pal Collar, so that the correct sensor interacts with the Bluetooth module when requested by the application.

The Arduino IDE makes it simple for us to use all the necessary communication protocols by providing libraries and easy functionality in the C-based Arduino language. In fact, this advantage over other IDEs such as Code Composer Studio, Atmel Studio, and MPLab, made us completely switch microcontrollers to one that was Arduino compatible. There is also an abundance of community support for the peripherals we are using for this system. Community libraries were available for the Speaker, GPS module, the pulse oximeter, and the bluetooth module. On Atmel Studio IDE, none of these community libraries exist, so all the sensor functionality would have to have been coded from scratch.

On top of this simplicity in coding, the Arduino IDE also allows for burning a bootloader onto a Arduino based microcontroller on a breadboard. The ATmega328P Dual in-line package model allows us to do this almost effortlessly without the use of solder. Once the bootloader has been burned into the microcontroller, we can upload the code directly to the PCB with the microcontroller still socketized. This is done by leaving access open to the microcontroller's TX, RX, and reset pins.

VI. System Casing

The system casing is responsible for holding the PCB as well as containing the GPS module. The casing is constructed using Fused Deposition Modelling (FDM) which uses filaments. We chose to use filament over resin due to its cheaper cost, sturdier material, and faster printing times. The only real benefit we saw that resin had over filament was higher quality prints, but since our casing is on the smaller side and doesn't require much detail, we decided to stick with the filament implementation. For our final design, we printed the casing with gray PLA filament.

The casing includes speaker holes to allow noise to travel through the casing without being dampened. It also extends past the length of the PCB to allow for the GPS module to be placed underneath the covering, thus protecting it from the elements. The covering does not weaken the signal of the GPS fix, as we have a coin cell battery attachment increasing its sensitivity and also because GPS signals can travel through the filament the casing is made up of.

Loops were made on either side of the casing to allow for the collar to be looped through securing the casing in place.

Lastly, holes were made to allow the battery pack to connect to the power pins on the PCB. Detailed views of the 3D printing model can be seen in Figures 12-15.

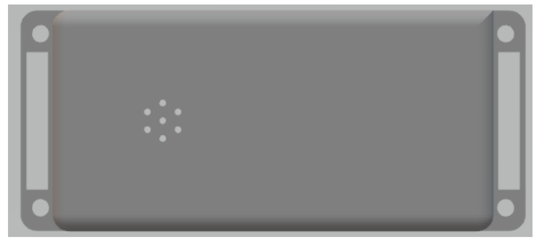

Fig. 12. Top side of the system casing. Speaker holes are shown.

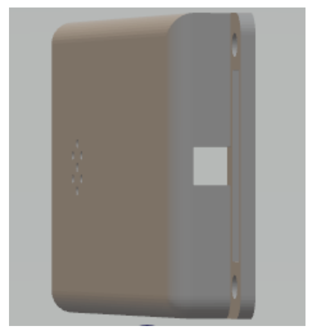

Fig. 13. Angled view of the top of the system casing. Shown is the hole in which the power system will be connected.

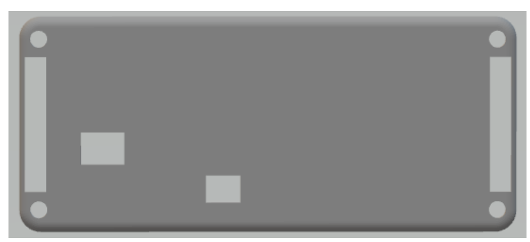

Fig. 14. View of the bottom part of the system casing. Shown are the holes for the pulse oximeter and sensitive parts of the PCB that cannot be closed off.

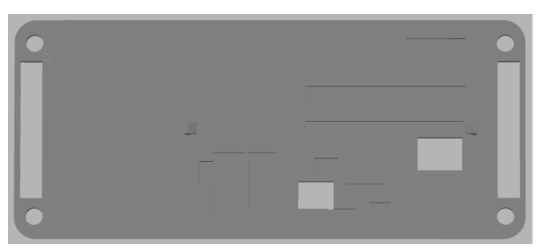

Fig. 15. View of the inside of the bottom of the casing. It has ridges to allow for the components to sit comfortably on the inside.

VIII. Conclusion

This project provided the group with a complex engineering problem which required all of the group member's expertise to accomplish. It also allowed the group to have a space to think critically, work well as a unit, and manage time in an engineering setting. It is also important to note how classes within the CECS prepared the group for this project, specifically in the areas of circuitry, embedded programming, and overall application development. Moving forward into professional careers, the group will take the lessons learned through this experience and use them to provide the world with better engineering solutions for the future.

Biography

Andres Graterol is a senior at the University of Central Florida who is planning on graduating with a Bachelor of Science in Computer Engineering and a minor in mathematics. He has an internship with Robins AFB Software Group in Summer 2022 and is planning on attending Graduate School in Fall 2022.

Zachary Kirksey is a senior at the University of Central Florida who is planning on graduating with a Bachelor of Science in Computer Engineering with a minor in Mathematics. Since summer 2020, he has been a Student Developer Co-op with the Naval Surface Warfare Center Panama City Division in Panama City, Florida. This includes working full-time in between school semesters, and part-time during each semester. After graduating, he plans to work as a Software Engineer for Lockheed-Martin in Marietta, Georgia.

Hanh Le is a senior at the University of Central Florida who is planning on graduating with a Bachelor of Science in Electrical Engineering. She is planning on taking the FE exam after her graduation.

Hunter Miller is a senior at the University of Central Florida pursuing a Bachelors of Science in Electrical Engineering. He has worked for Lockheed-Martin at RMS on the Electrical Hardware Team since the Summer of 2020 while taking classes full-time. After graduation, he plans to work full time for Lockheed-Martin with the Electrical Hardware team.

Acknowledgement

The Pet Pal team would like to acknowledge the assistance and support of our Senior Design supervisors, Dr. Richie and Dr. Wei. We would also like to thank Dr. Abichar for providing advice and assistance on embedded programming during rough spots in the making of this project. We have deep appreciation for our peers who helped us along the way, namely Mick Logtenberg, Bhupin Marwah, and Wyatt Vining.

References

[1] "GlobalTop-FGPMMOPA6H-Datasheet-V0A." GlobalTop Technology Inc.

[2] "K 15 S - 8 Ohm." *Visaton*, https://www.visaton.de/en/products/drivers/miniature-speakers/k-15-s-8-ohm.

[3] *MAX30102: High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*, https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf.

[4] "MDBT40-P256R." Raytac, 29 Apr. 2015.

Pet Pal

Keeping Your Companion Happy and Healthy