Embedded System for Recording Vocalizations of a Waterbird

A Thesis
Presented in Partial Fulfillment of the Requirements for the
Degree of Master of Science
with a
Major in Computer Engineering
in the
College of Graduate Studies
University of Idaho
by
Nikolai Tiong

Approved by:
Major Professor: John C. Shovic, Ph.D.
Committee Members: John C. Shovic, Ph.D.; Courtney Conway, Ph.D.; James Frenzel, Ph.D.
Department Administrator: Joseph Law, Ph.D.

August 2023

Abstract

Using animal vocalizations to determine the population of wildlife is a developing field. Finding out what the population is and whether it is increasing or decreasing helps government agencies determine what actions need to be taken to help conservation of endangered species. Surveys can be done by actively playing animal vocalizations and listening for responses but relies on researcher experience to correctly identify the species and whether it has been counted already. Statically placed recording devices in animal habitats can record nearby sounds but will only record animals that vocalize nearby. Placing a recording device onto the animals will record all the sounds it makes but also those of conspecifics it interacts with. This method has challenges such as a lower battery life, limited processing power and needs to minimize interfering with the animals' everyday activities. This thesis aims to design a device addressing these challenges that can be attached to the back of the Ridgway's Rail, a water bird that lives along the Southern Californian coastline and record the vocalizations it makes to help with population counting.

Acknowledgments

I would like to thank Dr John Shovic for many things throughout my journey with the University of Idaho: being the client during my senior capstone design project, teaching some of my most enjoyable classes in Real Time Operating Systems, Advanced Robotics I and II, being my major professor, obtaining a teaching assistantship and 2 summer research jobs for me.

Dr Courtney Conway for bringing this project to the attention of the engineering college and having it find its way to me and being on my committee. This has been a very challenging project and I had to learn many new topics that I had not come across throughout all the classes I have taken.

Dr Jim Frenzel for allowing me to take Digital Systems Engineering via Engineering Outreach when I asked about adding it to the schedule and being on my committee. That class was one of the most difficult ones I took, but also an enjoyable one working with Field Programmable Gate Arrays.

The Joe & Susan Rumble Endowment for providing the funding for the project.

Everyone who's helped or listened to both the highs and lows of this project in our lab including Marz, Hunter, Garrett, Lacey, and Jacob.

Table of Contents

Abstract ii

Acknowledgments iii

List of Tables vi

List of Figures vii

Chapter 1: Introduction 1

Chapter 2: Literature Review 4

Chapter 3: Design 14

3.1 Requirements 14

3.1.1 Expected Use Case 15

3.1.2 Audio Analysis 16

3.2 Hardware 17

3.2.1 Hardware Options 18

3.2.2 Hardware Selected 20

3.2.3 Hardware Architecture 22

3.2.4 Hardware Design and Assembly 27

3.2.5 Base Station 28

3.3 Software 29

3.3.1 Bird Backpack 29

3.3.2 Base Station 34

Chapter 4: Results 36

4.1 Functionality 36

4.2 Radio Range Testing 37

4.3 Power and Current Usage 40

4.4 Audio Capture ... 41
4.5 Size and Weight ... 42
4.6 Solar Generation ... 44
4.7 Conclusion ... 44
Chapter 5: Future Work ... 46
5.1 PCB Design ... 46
5.2 Fix Radio/Audio Recording Issue and Audio Glitch ... 46
5.3 Waterproofing the Device ... 47
5.4 Using the EFM32WG and SI446X ... 47
5.5 Use a Software Defined Radio ... 47
5.6 Add a Solar Panel and Charger ... 47
5.7 Researching Flexible PCB Manufacturing ... 48
5.8 Adding Additional Message Types Using the Radio ... 48
5.9 Additional Signal Processing ... 48
Chapter 6: Conclusion ... 49
References ... 51
Appendix A: Schematics ... 53
Appendix B: PCB Layout ... 57
Appendix C: Component List and Prices ... 59
Appendix D: EZR32WG Pinouts ... 63

List of Tables

Table 2.1: Power Consumption of the AudioMoth When Recording at Various Sample Rates. [14] 13
Table 3.1: Comparison of Embedded System Options. 19
Table 3.2: Comparison of EZR32WG Variants Used in This Project. 21
Table 3.3: Operational Amplifier Gain Options Available. 23
Table 3.4: Cascaded Operational Amplifier Gain Options Available. 24
Table 3.5: Receive (RX) Circuit Component Values. 27
Table 3.6: Transmit (TX) Circuit Component Values. 27
Table 3.7: Layout and Contents of Radio Message Packet Excluding Headers. 32
Table 4.1: Maximum Range of Radio With Various Setups. 39
Table 4.2: Current Draw of Primary Functions of the EZR32WG Microcontroller. 41
Table 4.3: Dimensions and Weight of Bird Backpack Versions, AudioMoth, µMoth and the Development Kit. 43
Table 4.4: Current Generated by Solar Panels. 44
Table 4.5: Comparison of AudioMoth/µMoth and the Bird Backpack. 45
Table C.1: Component List for Bird Backpack. 59
Table C.2: Component List of Supporting Hardware. 62
Table D.1: Pinouts of EZR32WG230 Used in Bird Backpack. 63

List of Figures

Figure 2.1: Rail Population Counted for Each Annual Survey Performed 1980-2018. Adapted From [3] 5

Figure 2.2: Rail Movement Around Tijuana Slough. [8]..... 6

Figure 2.3: Spectral Analysis of the Focal and Non-Focal Recording of a Distance Call. [9]. 7

Figure 2.4: Vocal Communication in Noise. (a) Sonograms of a Loud Syllable in the ‘No-Noise’ (Left) and ‘Noise’ (Right, 80 dB) Condition. [9] 8

Figure 2.5: Illustration of Calls Emitted by two Different Individuals of Cape Gannets in the Four Behavioral Contexts. [12]..... 9

Figure 2.6: Block Diagram of Embedded System to Record Cattle. [13] 10

Figure 2.7: Printed Circuit Board Design of the Cattle Recording Embedded System. [13] . 11

Figure 2.8: Component and Silkscreen View of the AudioMoth. [14]..... 12

Figure 2.9: AudioMoth Hardware Architecture. [14]..... 12

Figure 3.1: Spectrogram of a Rail Audio Recording. 16

Figure 3.2: Waveform of the Same Audio Recording From Figure 3.1. 17

Figure 3.3: Block Diagram of Hardware Architecture. 22

Figure 3.4: Operational Amplifier Pinouts for the EZR32WG 230 Model (Left) and 330 Model (Right). 23

Figure 3.5: Operational Amplifier Configuration for the Bird Backpack. 24

Figure 3.6: High Power Direct Tie Layout +20dBm. 26

Figure 3.7: Low Power Direct Tie Layout +13dBm. 26

Figure 3.8: The BRD4503B Board (Left) and BRD4001A Board (Right) Combined Make up the SLWSTK6222A Development Kit. 29

Figure 3.9: Block Diagram of Bird Backpack Operation. 30

Figure 3.10: State Machine of Bird Backpack. 31

Figure 3.11: Recording Audio Operation. 32

Figure 3.12: Process of Bird Backpack Requesting the Time From the Base Station via Radio. 34

Figure 3.13: State Machine of Base Station. 35

Figure 3.14: Block Diagram of Base Station Interactions With the AudioMoth Time Application and Bird Backpack. 35
Figure 4.1: AudioMoth Time App Setting the Time of the Development Kit Behaving as the AudioMoth. 36
Figure 4.2: The Development Kit Printing the Time on the LCD Screen After Having the Time set. 37
Figure 4.3: Map of Tijuana Slough With Projected Radio Range of Rails, Modified From [8]. 40
Figure 4.4: Comparison of Waveform From the μMoth (Left) and Bird Backpack (Right). . 42
Figure 4.5: Comparison of Spectrograms From the μMoth (Left) and Bird Backpack (Right). 42
Figure 4.6: Top of Version 0.4 of the Bird Backpack. 43
Figure A.1: Schematic of EZR32WG and Decoupling and Filtering Capacitors. 53
Figure A.2: Schematic of Linear Regulator, Battery, and USB Connector. 53
Figure A.3: Schematic of Microphone Controlled by P-Channel MOSFET. 54
Figure A.4: Schematic of Crystal Oscillators for the low Frequency, High Frequency, and Radio Clocks. 54
Figure A.5: Schematic for the MicroSD Card Controlled by P-Channel MOSFET. 55
Figure A.6: Schematic for the Voltage Reference Used by the Operational Amplifiers and ADC. 55
Figure A.7: Schematic for the +20 dBm Radio. 56
Figure A.8: Schematic of Pins Brought out to Headers for Testing and Debugging. 56
Figure B.1: Top Layer (1) PCB Layout. 57
Figure B.2: Layer 2 PCB Layout. 57
Figure B.3: Layer 15 PCB Layout. 58
Figure B.4: Bottom Layer (16) PCB Layout. 58

Chapter 1: Introduction

The Light-Footed Ridgway's Rail (*Rallus obsoletus levipes*) is a federally endangered species of water bird [1] that lives in coastal marshland environments along Southern California and northern Baja California, Mexico. Human development from Santa Barbara County to San Diego County has reduced the habitat of the Ridgway's Rail by 75% since the late 19th century [2]. To aid conservation efforts, the population has been monitored by performing an annual survey since 1980 [1]. The purpose of these surveys is to determine how many adult rails persist, and whether the population is increasing or decreasing. A low of 142 breeding pairs in 1985 were counted. It reached a peak of 656 pairs in 2016 before dropping down to 356 pairs in 2018 [3].

There are multiple ways to perform bird surveys: visual observation, marking and recapturing birds, and acoustics. Visual observation requires a trained person to travel around the animal's habitat to identify the species being counted. However, there are difficulties with observation such as animals that may be small, camouflaged, nocturnal, underground, live in water or in thick foliage [4]. Another issue with observation is that there are false positives – counting animals as present when they are absent through double counting or misidentifying an animal, and false negatives – not detecting animals that are present [5]. Trapping and marking animals can help alleviate these issues but handling can induce stress in the animal and is extremely expensive and time-consuming. Moreover, some species are not easy to capture.

Acoustic observation is the most used method to survey for birds and surveys for rails typically consists of both passive and active surveys. A passive survey is like visual observation in that a trained person travels around the animal's habitat, but instead is listening for animal vocalizations that are unique to the focal species. This requires the observer to make instantaneous decisions about whether a sound heard is from the animal being surveyed, or from a different conspecific and to decide whether each sound is from an individual that has already been counted. It also relies on the observer's hearing ability and experience which can vary greatly among surveyors [1]. An active survey differs in that animal vocalizations are induced by playing a recording of a vocalization (call-broadcast survey) with the expectation that animals of the same species nearby will respond. Both methods have some of the same drawbacks as visual observation: false positives from double counting, misidentification and false negatives from animals that don't respond [4].

An alternative to having a trained observer perform a passive survey is to place devices with microphones around the animal's habitat to record audio i.e., Autonomous Recording Units (ARUs). ARUs can be superior to visual surveys because some animals produce sounds that are detectable at a greater distance than visual detections. They also operate in all light conditions, are less affected by weather, and the data captured can be processed and analyzed using automated processes [4]. However, audio intensity drops off the further away the audio capture device is from the source: doubling the distance reduces the intensity by 6 dB. This means statically placed acoustic devices may not be able to pick up far away sounds.

Placing a recording device on the animal provides a consistent distance from the animal it is attached to. This can also allow recording of other conspecifics that interact with it such as a breeding mate. The distance can then be calculated based on the relative intensities of the audio recorded. Placing devices on animals involves capturing them and then attaching the device in a manner that doesn't interfere with mobility or other activities. This type of audio capture is a developing field that can assist with better estimating populations and produce novel information on behavior of secretive animals that are sensitive to human presence.

This thesis aims to do the following:

- Design and assemble a prototype embedded system small enough to fit on a Ridgway's Rail that can record audio and has wireless connectivity. This device will be referred to as the "Bird Backpack".
- Design a second embedded system that can be used to communicate with the Bird Backpack while being carried around by researchers. This device will be referred to as the "Base Station".

The rest of this thesis is organized as follows. Chapter 2 will present a literature review of research related to these fields and key takeaways that may be applied. Chapter 3 will describe the requirements, use case, evaluation of potential hardware options, hardware design and software design for both devices. The results of the manufactured and assembled device's functionality and experiments performed will be presented in Chapter 4. Chapter 5 will discuss the future work that can be done to improve upon the design. Chapter will provide a conclusion to the entire project.

Chapter 2: Literature Review

The literature review for this thesis consisted of multiple topics including current research on the Ridgway's Rail, population estimation and surveys using acoustics, attaching tracking devices to wildlife to determine their range, attaching recording devices to wildlife, recording audio, and analyzing the different types of calls, and custom embedded devices that record audio.

[1], [3] and [6] are part of a series of annual counts of the Light-footed Ridgway's Rail from 1980-2018. Wetlands along Southern California were surveyed by between one and three observers. These consisted of walking through and listening for spontaneous calls by Rails. Surveys were generally conducted in the 2 hours before sunset but can also be done 2 hours after sunrise. This is when Rails are most vocal. Rails make a variety of sounds and are described as "clapper", "kek", "kik, "kek-burr", and "bup". They also perform "paired duets" where 2 members of a breeding pair give the "clapper" call in unison. Rails are very responsive to other rails' calls and, hence, vocalizations can be induced by performing a playback of a recording. However, research performed on the Yuma Ridgway's Rail, a closely related subspecies of Ridgway's Rail, indicates that they can become conditioned to the recordings if it's used too frequently in the same location.

A captive breeding program has also been producing rails for release into the wild to bolster the population since 1980 by [3]. Captive rails are bred in an aviary at 2 breeding facilities in southern California. Chicks are raised for one month before being moved to a conditioning pen in preparation for release into the wild. The Rails are marked with a metal band around their leg to individually identify each bird. 49 Rails were released in the 2018 season, split up into several breeding marshes to help bolster multiple breeding populations. Figure 2.1 shows the size of the wild Rail population over the entire survey period and is measured in breeding pairs. Decreases in population in 2008 have been attributed to weather events. Decreases in 2017-2018 are thought to have been caused by one or more of the following: reduced habitat from ocean inlet closure, predation, rising sea levels, and climate effects.

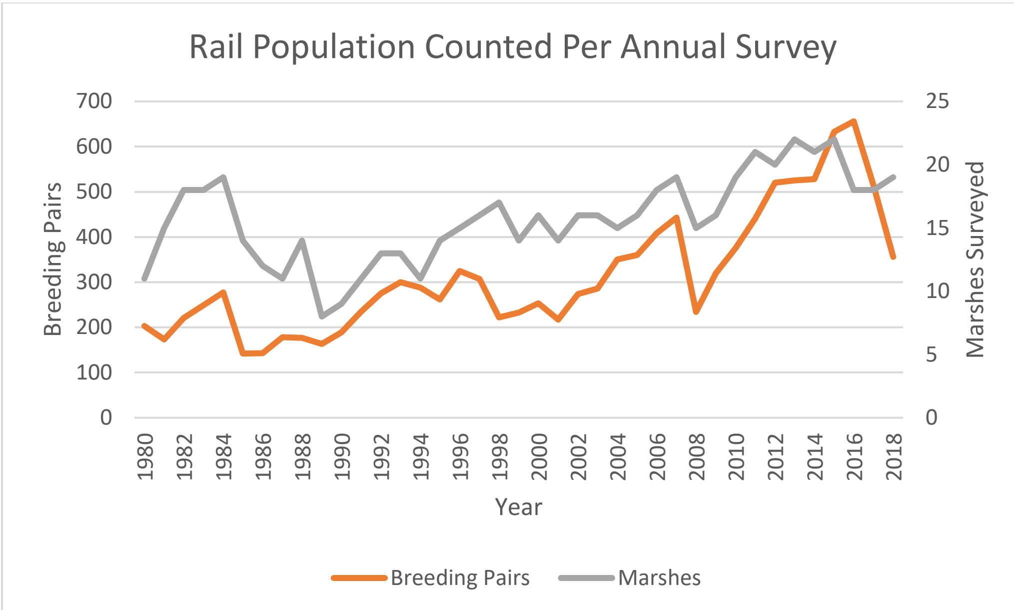

Figure 2.1: Rail Population Counted for Each Annual Survey Performed 1980-2018. Adapted From [3]

The movements of Light-footed Ridgway's Rails have been tracked around Tijuana Slough and Batiquitos Lagoon, San Diego, in [7] and [8]. Rails were lured out with audio playback, then captured using a carpet trap. A small GPS transmitter with a solar panel weighing 6-9 grams were attached to the backs of the Rails using a small backpack fashioned to fit them. Positional location was transmitted several times a day. Results from this study show that the Rails generally move to a specific area and then stay within a 100-meter radius of this location. It also demonstrates that Rails can be captured without injuring them and small devices can be attached. Figure 2.2 shows the movements of tracked Rails in Tijuana Slough.

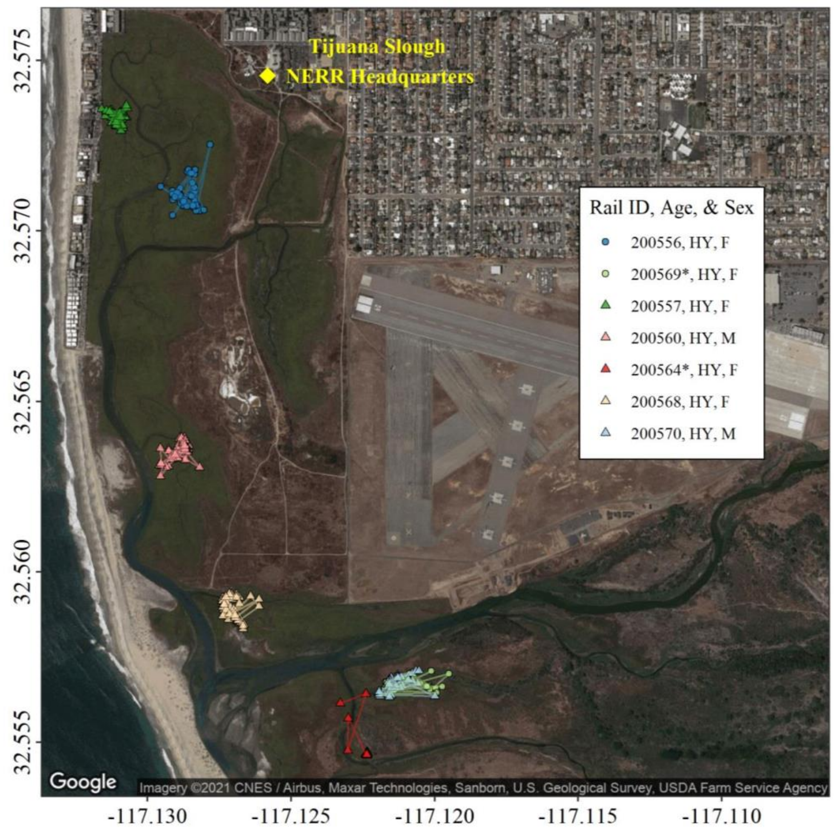

Figure 2.2: Rail Movement Around Tijuana Slough. [8]

Passive acoustic surveys were explored in [4]. The number of animals cannot be counted directly – the number of vocalizations can be, but not how many animals produced the vocalizations detected. A population density estimate was modelled by:

$$\widehat{D} = \frac{n(1-\hat{f})}{\hat{p}a\hat{r}}$$

Where:

- n is the number of detected vocalizations.
- $\hat{f}$ is the proportion of false positives.
- $\hat{p}$ is the probability of detecting an object within an area.
- a is the area.
- $\hat{r}$ is the multiplier that converts vocalization density to animal density.

Obtaining the values for f, p and r requires estimation from the data n collected in the region.

Passive sensors can be used both on fixed and mobile platforms. Mobile platforms are common for marine wildlife, while fixed ones can be used for terrestrial wildlife on objects such as trees, or on the animal itself as in [9] and [10]. A small condenser microphone (Knowles FB-23359) and wireless transmitter (unspecified) weighing 0.75 grams were attached to captive Australian zebra finches (*Taeniopygia guttata castanotis*) using a leg loop harness. Audio was transmitted using frequency modulation between 270-320 MHz with a channel dedicated to each finch. These were picked up with Yagi antennas, a radio receiver, and then digitally recorded.

In [9], the experiment was run over a 20-day period with the finches placed in the cage for the first 7 days without the microphone backpack attached to establish a baseline. The number of calls made were recorded with external microphones and counted, then the microphone backpack was attached. After a week, the battery was replaced, and the experiment continued. Results showed that both the movement and the number of calls produced dropped from a mean 1900 calls per day to almost 0. The number of calls increased each subsequent day before stabilizing around 1500 calls per day on the 5th day after attaching the microphone backpacks. A smaller drop to about 900 calls per day was seen with the battery swap but reached the baseline within 2 days. The audio recordings were also analyzed with sonograms, and power spectrum. Analysis of the same audio recorded from two different finches showed that it was possible to identify the sound-emitting individual based on the normalized power spectrum and is shown in Figure 2.3.

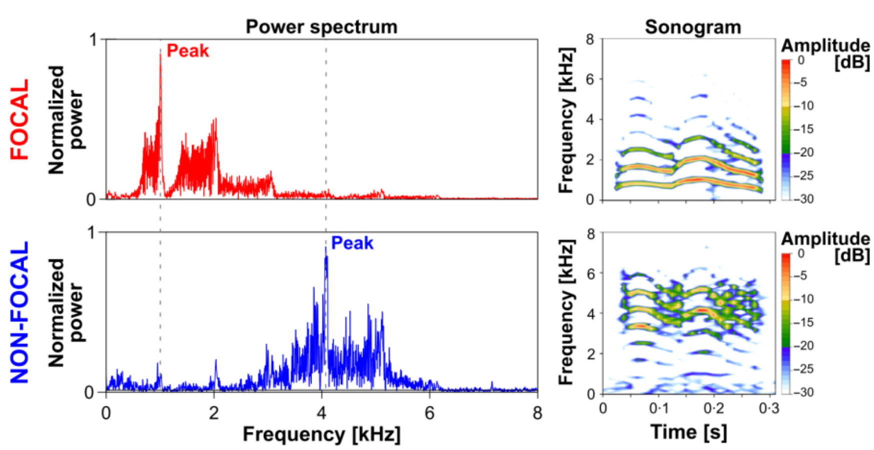

Figure 2.3: Spectral Analysis of the Focal and Non-Focal Recording of a Distance Call. [9]

Noisy conditions were also tested where 80 dB white noise was played 1 meter away from the external speaker. A comparison of the backpack and external microphone recordings shows that the backpack was unaffected by the white noise, while the external microphone completely masked the calls and is shown in Figure 2.4.

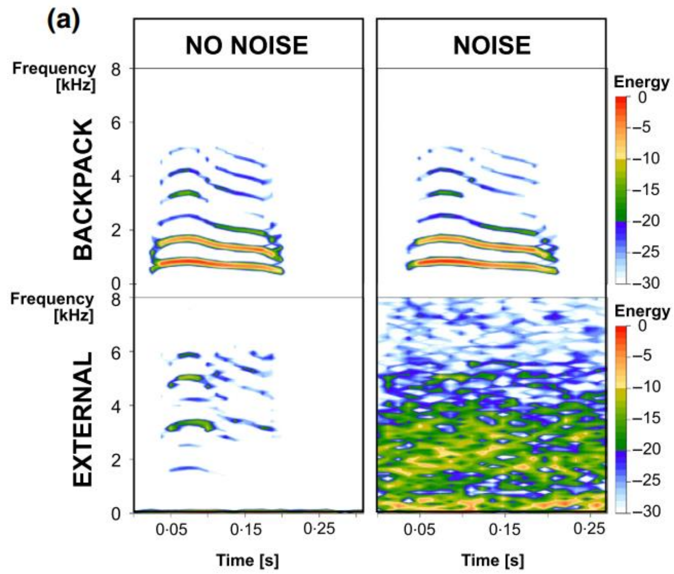

Figure 2.4: Vocal Communication in Noise. (a) Sonograms of a Loud Syllable in the 'No-Noise' (Left) and 'Noise' (Right, 80 dB) Condition. [9]

This shows that attaching a microphone onto the animal directly can pick up vocalizations much better than a statically placed microphone in their habitat. It is also essential for the separate recordings to be synchronized correctly. This is to determine if a call recorded on one device appears in the recording of another device.

Recording devices have been attached to other birds such as jackdaws (*Coloeus* spp.) in [11]. A digital voice recorder (Edic Mini Tiny A31), battery (ICP481323PA) and radio transmitter used for relocating the device (Holohil BD-2), weighing a combined 11.5 grams. The device was attached as a backpack using a harness to both captive and wild jackdaws and recorded continuously for several hours in the morning at 22,050 Hz sampling frequency and saved to an uncompressed WAV file. The recordings were then used to train and test a classifier and probabilistic latent component analysis (PLCA) event detection system for the different types of calls produced. Types of sounds recorded and classified include walking, movement, self-maintenance, non-focal calls, focal calls, looking around and flying. The classifier performed best for the sounds that had more audio recorded and showed that the model can be used for automatically classifying animal activities.

The Cape gannet (*Morus capensis*) is a seabird that has also been studied with recording devices in [12]. An audio recorder (Edic mini Tiny+ B80) recording at 22050 Hz, along with a battery, GPS, video camera and time-depth recorder weighing between 50-90 grams in various configurations were attached to Cape gannets. Since this bird dives underwater, the devices were waterproofed using nitrile gloves which attenuated the amplitude by 3 dB in laboratory testing. Cape gannets were captured at their nests to attach the recording device. The bird would then leave their nest to go on a foraging trip ranging from 4-25 hours before being recaptured to retrieve the device. Four types of calls were identified and classified using the random forest algorithm – on the water, take off, flying, and diving. Sonograms show that flying and diving have similar waveforms and that calls of different individuals appear distinct as shown in Figure 2.5.

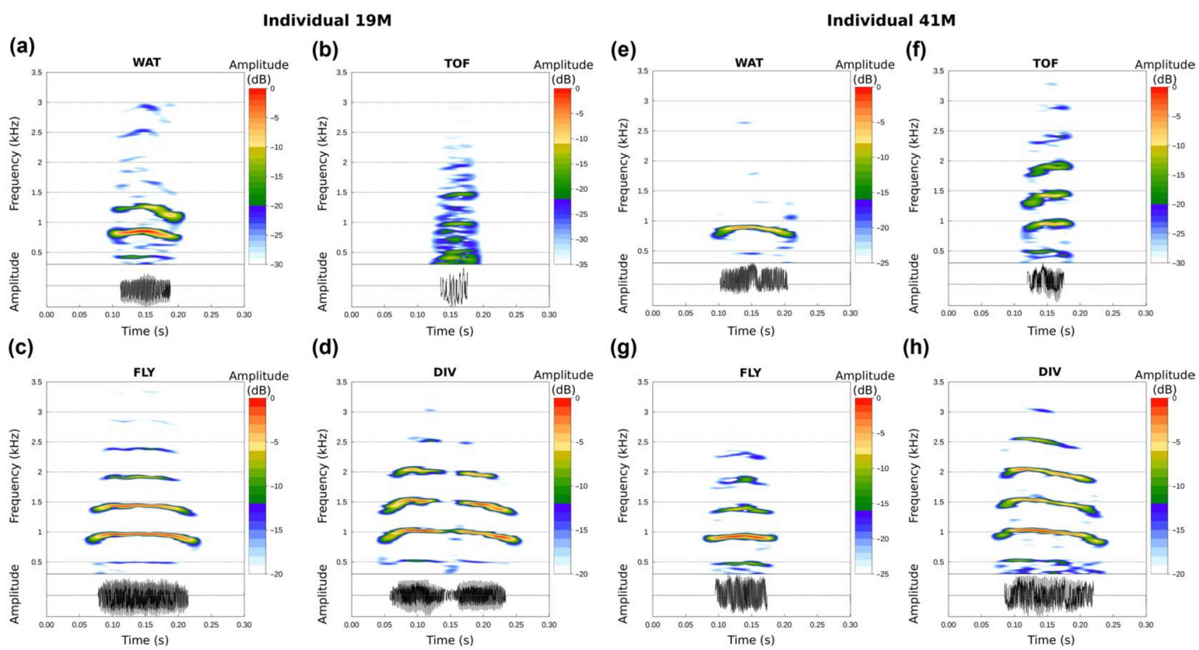

Figure 2.5: Illustration of Calls Emitted by two Different Individuals of Cape Gannets in the Four Behavioral Contexts. [12]

The classification algorithm was able to classify the four categories with accuracy of 88, 62, 56, and 66% for water, take off, flying and diving respectively. The Cape gannets gained weight during their foraging trips indicating that the devices did not hinder their ability to obtain food.

Customized devices have also been designed and attached to cattle to record vocalizations in [13]. A printed circuit board (PCB) was designed with a MCF51JM128 32-bit microcontroller. An electret microphone was used and filtered using a cascade of TLV2784 operational amplifiers. The signal was then amplified with a MAX9814 microphone amplifier. A TLV2781 level shifter was used to shift the mean of the signal from 0 to 1.65 V so that the range can have both positive and negative values for the analog-to-digital conversion (ADC). An 8-bit ADC on the microcontroller was used giving a resolution of 12.9 mV per step. Audio was recorded to W25Q128FV FLASH memory chips via a Serial Peripheral Interface (SPI) bus and then transmitted to a Base Station via an XBee XB24 radio at 2.4 GHz and has a range of up to 500 m. A voltage regulator - TPS7a4901, solar charger – BQ24230, 1 W solar panel and 2500 mAh battery were used to provide power to the device and keep it charged. A GPS receiver was used to provide the time to the real time clock and capture position of the cattle. The hardware architecture is also represented in Figure 2.6 and the assembled circuit board is shown in Figure 2.7.

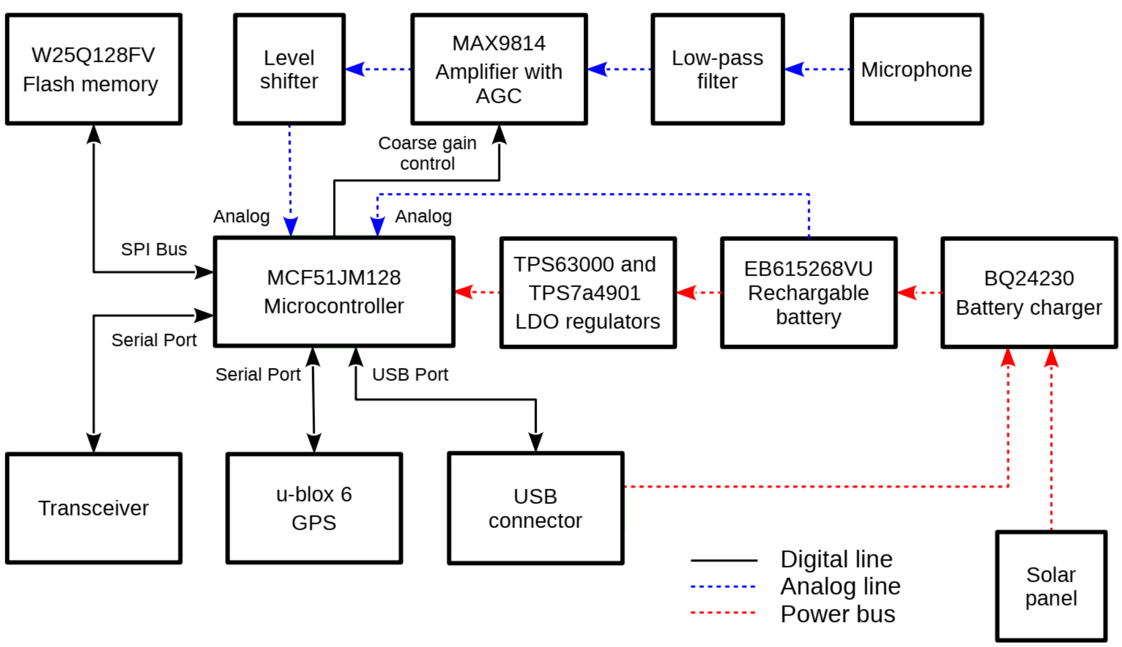

Figure 2.6: Block Diagram of Embedded System to Record Cattle. [13]

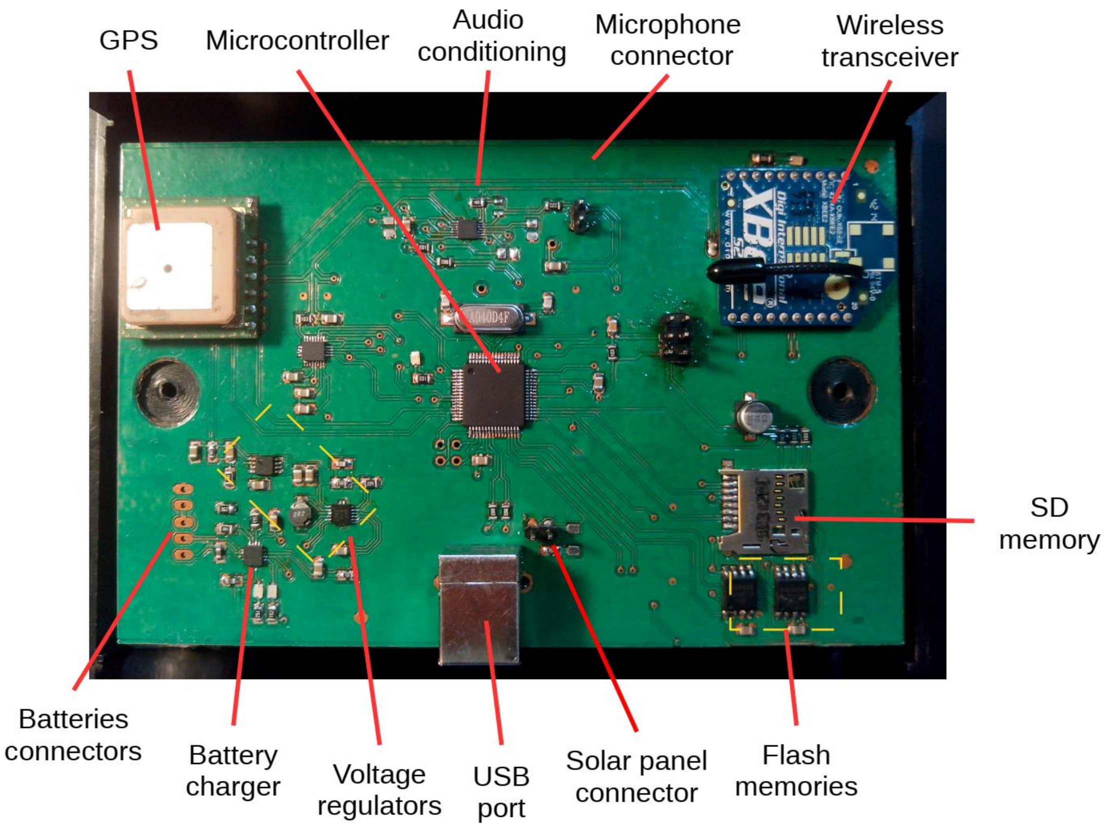

Figure 2.7: Printed Circuit Board Design of the Cattle Recording Embedded System. [13]

The purpose of this device was to record audio and process the audio to identify the foraging behavior of the cattle. Audio was sampled at 44.1KHz frequency. The ADC was sampled every 500 us and analyzed to determine whether the sound record was a “chew”, “bite” or “chew-bite” and record the length of time, amplitude, time, and position that the sound occurred. This information was then assembled into a data packet every 5 minutes. The Base Station would then request a packet transmission from the device periodically and all the data packets would be sent using the wireless transmitter. To save power, the device would go into sleep mode when it wasn’t performing any actions. The system was able to correctly detect 92% of events as feeding ones and then correctly classify 78% of these events into chew, bite, or chew-bite. This study showed that creating a custom embedded system is possible, same with on-board processing of data, but it is limited by power and computational cost.

The AudioMoth is a customized embedded system that functions as an acoustic logging device [14]. The purpose of it is to provide a cheaper alternative to commercial solutions that cost close to USD $1,000, coming in at around $50. It uses the EFM32 Wonder Gecko microcontroller which is based on the ARM Cortex-M4 processor. It can run at very low power while in sleep mode and still maintains some functionality. The AudioMoth measures 58x48x15mm and contains a USB port for connecting to a PC and MicroSD slot for storing recordings. The EFM32WG contains most of the functionality including operational amplifiers, 12-bit ADC, SPI for the MicroSD card, Direct Memory Access (DMA) for transferring audio samples to SRAM. An external SRAM module is used to store the samples using an External Bus Interface (EBI). This allows for recording audio at high samples rates – up to 384 KHz with oversampling. The PCB and enclosure are shown in Figure 2.8, while the hardware overview is shown in Figure 2.9.

Figure 2.8: Component and Silkscreen View of the AudioMoth. [14]

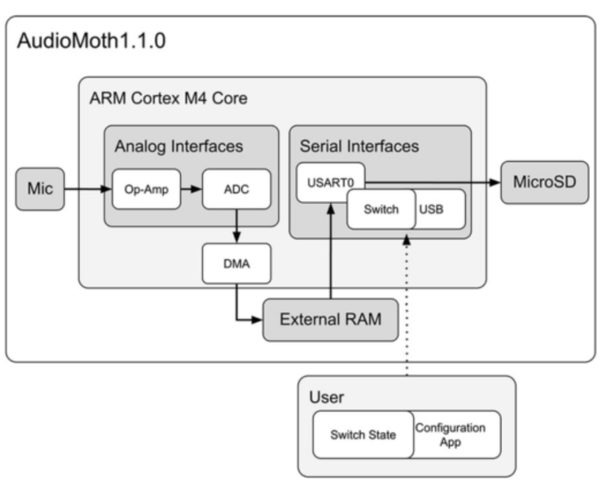

Figure 2.9: AudioMoth Hardware Architecture. [14]

The AudioMoth hardware and software is open source with schematics, layout, and the software projects available on GitHub [15]. It is configured using a suite of Windows applications that are used to set the time, update the firmware, or upload custom firmware and change settings such as filters, sample rate, recording schedule, recording duration, and gain. Sleep modes are utilized when not recording as well as during the recording process by using DMA. The current draw for different sample rates is shown in Table 2.1.

Sample Rate (Hz)	**Recording Current (mA)**
8,000	10
16,000	12
32,000	13
48,000	14
96,000	17
192,000	25
256,000	30
384,000	40

Table 2.1: Power Consumption of the AudioMoth When Recording at Various Sample Rates. [14]

A smaller version of the AudioMoth has also been made by the authors called the μMoth (micro moth). This device maintains the functionality of the AudioMoth but in a much smaller size: measuring 32 mm x 24 mm and weighing 5 grams without a battery.

Chapter 3: Design

This chapter covers the design of the Bird Backpack and Base Station. The requirements of the entire system will be described as well as the expected use cases for the devices. Audio recordings are provided and analyzed to determine dominant frequencies of the Rail's calls for microphone selection. Different hardware solutions are researched and evaluated based on metrics such as features, RAM, and power usage. The hardware and software design of the entire solution are then covered in detail.

3.1 Requirements

The following design requirements were specified for the Bird Backpack:

- Maximum weight of 9 grams. The Ridgway's Rail weighs between 248-301 grams for females and 300-350 grams for males. The maximum weight that can be attached to an animal for testing is 5% of its body weight [15].
- Waterproof/resistant. The Ridgway's Rail lives in marshland environments, and they will occasionally dive underwater. The device must be able to withstand being submerged temporarily and still operate.
- Shaped to fit a Rail's back better than a flat device.
- Battery to last at least 2 weeks, the longer the better. The Rail primarily vocalizes during two daily 2-hour windows coinciding with sunrise and sunset. The device should be inactive outside of these times to conserve power. Adding a solar panel to recharge the battery would also be very useful.
- Record audio to a storage device. The Rail's vocalizations are 1-3 seconds long and should be triggered so that it doesn't record all the time. The storage device should be large enough to store as many recordings as possible.
- The recordings need to be timestamped with the date and time to the second.
- The recordings are analyzed and information about the recordings transmitted. If possible, transmit the entire recording.

3.1.1 Expected Use Case

The Bird Backpack and Base Station are expected to be used in the following manner:

- Attach fully charged batteries to the Bird Backpack devices and seal them to prevent water intrusion.
- Get the time from a host device – laptop computer or another device – a Base Station.
- When the Bird Backpacks are powered on, they will request the time wirelessly.
- The Bird Backpack will then determine the state based on the time of day.
- Researchers will place traps to capture Rails in their habitat.
- Once a Rail is captured, place the Bird Backpack on the Rail and release it.
- The Bird Backpack will then record audio when the Rails make calls during the prescribed 2-4 hour sampling windows.
- Come back to the Rail's habitat every few days with the Base Station/laptop and a large antenna.
- The Bird Backpacks will send out messages periodically indicating that they have recordings and metadata about the recordings – time, number, frequencies, amplitude.
- When the Base Station receives these messages, reply to let the Bird Backpack know that they don't need to send messages about the most recent recordings anymore.
- When the batteries in the Bird Backpacks die, set traps to capture the Rails and replace the Bird Backpack with a new one.

3.1.2 Audio Analysis

Some audio recordings of Rails were provided in mono wav, stereo wav and MP3 format. There were examples of the “kek”, “kek-burr”, “bup” and the paired “clapper” duets. These were analyzed by using a spectrogram from the software Sonic Visualizer to identify the frequencies of the calls made by the Rails. An example of a recording is shown in Figure 3.1.

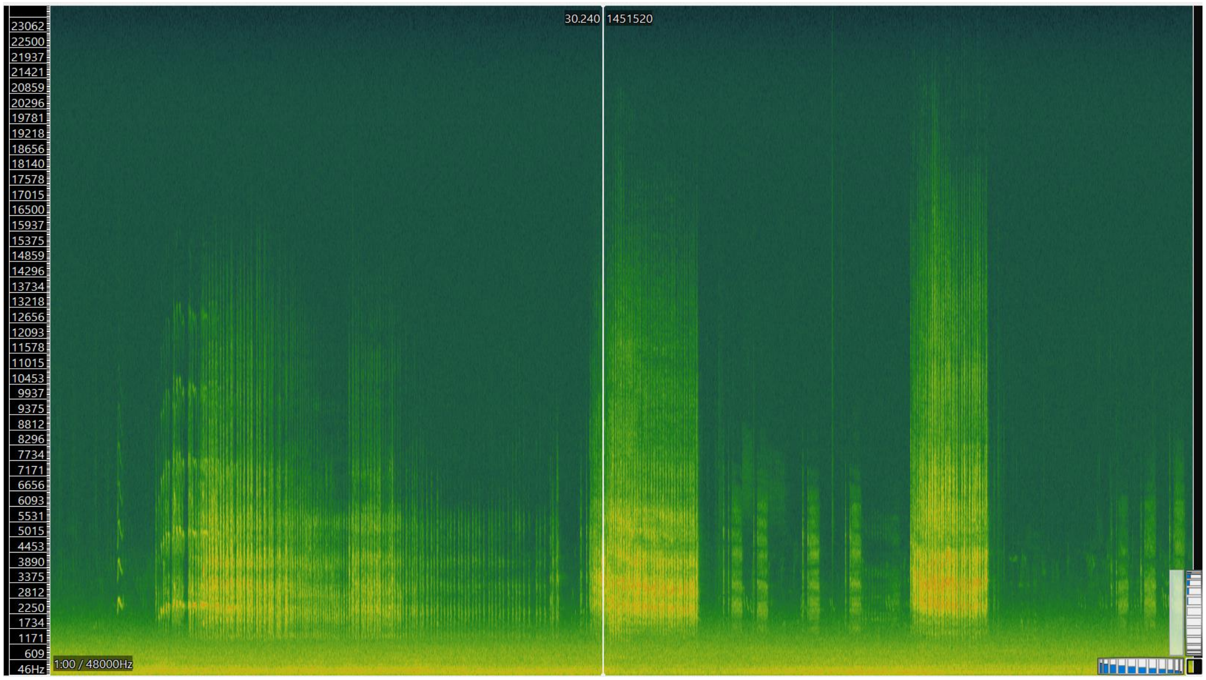

Figure 3.1: Spectrogram of a Rail Audio Recording.

There is a consistent background noise in the recording that is around the 0-1000 Hz range. The Rail’s produce calls that have the greatest magnitude from 2,000-6,000 Hz. Some of their calls have frequencies up to and above 20,000 Hz, however these typically cannot be heard by humans. The waveform of the recording is shown in Figure 3.2 and shows the relative intensities.

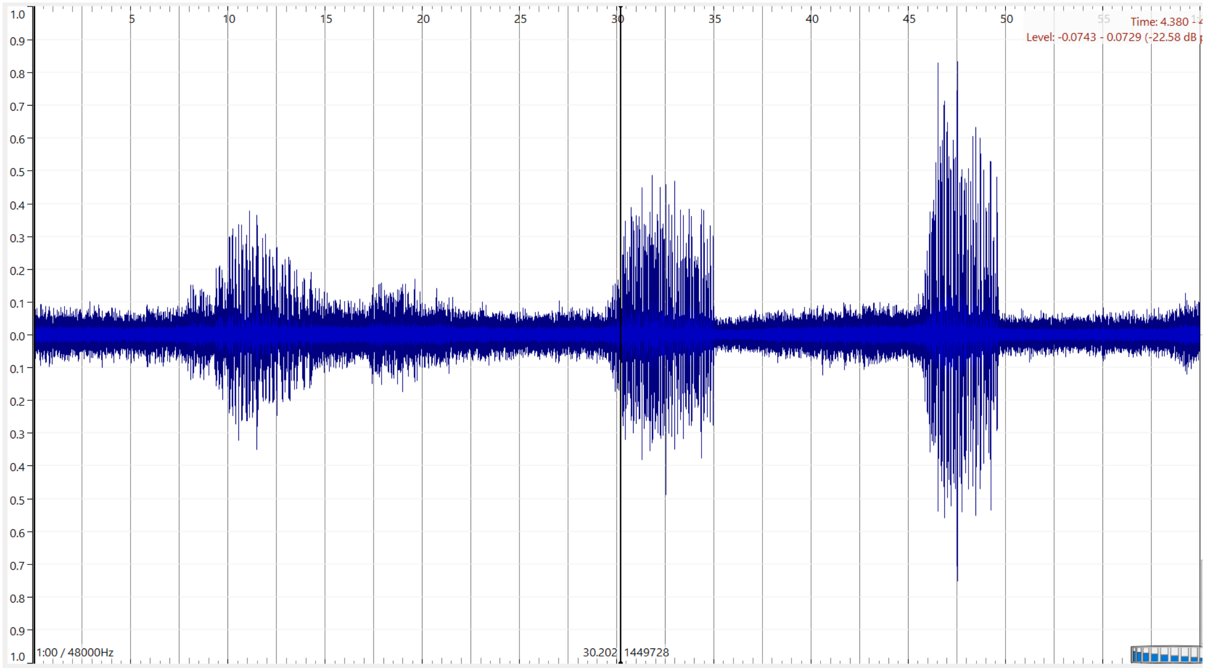

Figure 3.2: Waveform of the Same Audio Recording From Figure 3.1.

3.2 Hardware

There are four viable hardware device solutions that have been identified: use a ready-made microcontroller board such as an Arduino or ESP32; design a custom microcontroller like the AudioMoth; design a solution using a Field Programmable Gate Array (FPGA); design a custom integrated circuit as part of a custom microcontroller. With the requirements listed in 3.1, the following hardware requirements should be met:

- The microcontroller must be fast enough to perform all its operations in a timely manner, in particular recording audio so that samples are not lost.
- The microcontroller must be low power and have sleep options to reduce the power consumption when not in use.
- Interrupts must be available to use to wake the device from sleep mode to perform functions.
- Interfaces for microphones must be present – operational amplifiers and an analog-to-digital converter (ADC) for an analog microphone, or I2C/I2S for a digital microphone.
- The microphone should be capable of picking up frequencies in the range of 0-20 KHz.

- SPI must be present to use a MicroSD card. MicroSD cards are the smallest commercially available storage medium that are readily usable on PCs.
- A wireless radio to transmit data. The radio is ideally low power and can have a large range.
- A real time clock (RTC) must be present to allow the device to know what time of day it is. It must be accurate enough to reliably have the correct time over the battery's lifetime.

3.2.1 Hardware Options

The ESP32 is a system on a chip that consists of an Xtensa dual-core 32-bit microprocessor and operates at up to 240 MHz. It has wireless connectivity with Wi-Fi and Bluetooth, a 12-bit analog-to-digital converter, SPI, I2S, I2C, up to 34 GPIO pins, 320 KB RAM, 448 KB ROM. With these interfaces available, an analog microphone would require an external operational amplifier for the incoming signal. A digital microphone would use the I2S interface. The typical range of Wi-Fi is 50 meters and Bluetooth is 10 meters. This would require the wildlife researchers to be very close to the Rails to pick up any transmissions. A MicroSD card would be able to be used over SPI to store audio recordings. However, the current draw of the ESP 32 is up to 68mA when running. Several sleep modes are available which reduce the current to 1 uA in the lowest power state.

The Arduino is an open-source hardware and software project that designs microcontrollers using the Atmel AVR processor running at 16 MHz. The Arduino Nano contains a 10-bit ADC, 32 KB FLASH, 2KB SRAM, 1KB EEPROM, I2C, SPI and up to 14 GPIO pins. The low resolution of the ADC, lack of any radio and not having any operational amplifiers would require additional integrated circuits to be added to a design. The current draw of the Arduino Nanoo is 35mA when running. Sleep modes can reduce this amount to 30 uA.

The AudioMoth and µMoth are custom microcontroller designs using the Silicon Labs EFM32 Wonder Gecko microcontroller. This is based on the ARM Cortex M4 processor. It runs at up to 48 MHz, has up to 256 KB of FLASH, 32 KB of SRAM, a 12-bit ADC, up to 4 operational amplifiers, SPI, I2C, I2S, UART, USART and up to 93 GPIO pins. It has 5 run modes which draws between 5mA idling in run mode and 3nA in the deepest sleep mode. The EFM32WG does not have a radio, but Silicon Labs has designed an accompanying sub gigahertz radio module called the SI 446x (x ranging from 0-9), using a proprietary protocol called EzRadio Pro. This radio can be configured to transmit between 142-1050 MHz. The EZR32WG combines the EFM32WG and an SI446x radio onto the same integrated circuit while removing some of the GPIO pins, operational amplifiers, and the EBI.

A Field Programmable Gate Array (FPGA) can be programmed with VHDL or Verilog/SystemVerilog to record audio if it has a microphone and MicroSD card connected. However, it would need to be small enough and have enough I/O pins for them and a radio. It would be likely that a custom solution like the AudioMoth/μMoth would have to be made.

An alternative to finding a microcontroller that has all the required features is to design a custom IC with everything on board – an Application Specific Integrated Circuit (ASIC). This would also allow the IC to be optimized for operations such as recording the audio to MicroSD, processing it, keeping time, and controlling sleep modes. However, the time required to design this could be years and have a very high cost associated with fabrication - >$10,000 using a 0.6um process. Any kind of fault in the design would likely require a revision and extra cost to fabricate the new version. Additionally, a custom PCB would still have to be designed and assembled with peripherals such as the microphone, radio, and MicroSD card. Due to the cost and time requirements, this was not considered a viable option.

Table 3.1 summarizes the differences between the ESP32, Arduino Nano, the EFM32WG and EZR32WG.

	ESP32	**Arduino Nano**	**Custom PCB with EFM32WG/μMoth**	**Custom PCB with EZR32WG**
Processor	Xtensa dual core 32-bit, 240 MHz	AVR Atmega 328 8 bit, 16 MHz	ARM Cortex M4, 48 MHz	ARM Cortex M4, 48 MHz
SRAM	320 KB	2 KB	32 KB	32 KB
FLASH	448 KB	32 KB	64/128/256 KB	64/128/256 KB

Table 3.1: Comparison of Embedded System Options.

	ESP32	Arduino Nano	Custom PCB with EFM32WG/µMoth	Custom PCB with EZR32WG
Power Modes	5	6	5	5
Current Draw (typical)	68mA/0.8mA/150uA/5uA/1uA	19mA/12mA//1.65mA/1.62mA/0.84mA/0.36mA	10mA/3mA/0.95uA/0.65uA/20nA	10mA/3mA/0.95uA/0.65uA/20nA
GPIO Pins	Up to 34	Up to 14	Up to 93	Up to 41
ADC	18 channel, 12-bit SAR	6 channel, 10-bit SAR	8 channel, 12-bit SAR	8 channel, 12-bit SAR
Radio	Wi-Fi, Bluetooth	No	No – Can add SI 446x radio module	Yes – EzRadio Pro sub gigahertz
Op Amp	No	No	Yes - 4	Yes - 2
Communication Interfaces	SPI, I2S, I2C	SPI, I2C, UART	SPI, I2S, I2C, UART, USART, USB (330 model), EBI	SPI, I2S, I2C, UART, USART, USB (330 model)
Dimensions	18x25x3mm	18x45mm	Varies / µMoth 32x24 mm	Varies
Weight	9g	5g	Varies / µMoth 5g	Varies
Approx Cost Per Unit	$10	$30	~$50-100	~$50-100

Table 3.1: Comparison of Embedded System Options, Continued.

3.2.2 Hardware Selected

Since the ESP32 and Arduino options don't have all the features required on board such as the operational amplifier, a solution using them would require a customized PCB using the Xtensa or AVR microprocessors and using additional integrated circuits to provide the additional functionality. The AudioMoth project is open source and has provided schematic designs and source code. This gave the option of using the AudioMoth design as a base and modifying both the schematics and code to better fit the requirements for the Bird Backpack. However, due to shortages in obtaining hardware throughout 2021 and 2022, the

EFM32WG microcontroller and SI446x radio were not in stock at suppliers such as Digikey and Mouser. There was a limited stock of the EZR32WG microcontroller available. Table 3 shows the microcontrollers that were purchased at different stages for designing and testing when stock was available.

Model	**Flash (KB)**	**Radio Power (dBm)**	**Max Sensitivity (dBm)**	**USB**	**GPIO**
EZR32WG330F64R60G	64	+13	-129	Yes	38
EZR32WG230F256R63G	256	+20	-129	No	41
EZR32WG330F256R63G	256	+20	-129	Yes	38
EZR32WG230F256R69G	256	+13 and +20	-133	No	41

Table 3.2: Comparison of EZR32WG Variants Used in This Project.

The SLWSTK6222A development kit was also purchased which contained an EZR32WG330F256R63G. The development kit consists of a debugger for programming custom boards using the EZR32WG, an LCD screen, 2 push buttons, 2 LEDs, current and voltage monitoring, breakout headers for the GPIO pins, and the radio configured to use the 915 MHz frequency. 915 MHz frequency is the center of the 902-928 MHz band and is available for unlicensed use in the United States of America. The development kit was also used as the Base Station.

To keep the size of the custom PCB size down, surface mount components were primarily used: resistor, capacitor and inductor sizes were 0402 with some higher value capacitors using 0603. Hardware was selected based on whether it was in stock at the time of purchase. Other components used include the following:

- Linear regulator at 3.3V.
- Crystal oscillators.
    - Low frequency clock: 32.768 KHz.
    - High frequency clock: 48 MHz .
    - Radio: 30 MHz.
- MicroSD card slot.
- 20 KHz microphone.
- USB connector.

- Voltage reference for the amplifier.
- uFL connector.
- P-channel MOSFETs for the microSD card and microphone.
- JST connector for the battery.

The complete list of components with part numbers and prices is in Appendix C.

3.2.3 Hardware Architecture

Figure 3.3 shows the hardware architecture of all the external components and how they interact with the internal components of the EZR32WG microcontroller.

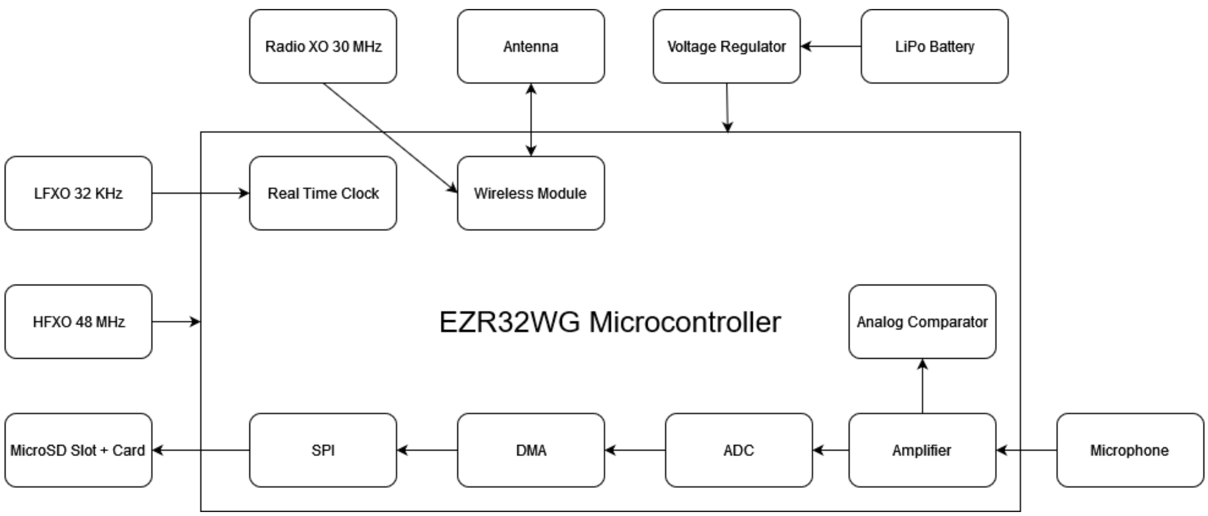

Figure 3.3: Block Diagram of Hardware Architecture.

Power is provided to the device by a 3.7V LiPo battery. Potential capacities are 100mAh and 150mAh weighing 3g and 4.5g respectively. A 3.3V linear voltage regulator was selected instead of a switched unit due to its simplicity, cost, and low noise. However, this comes at the cost of a higher quiescent current and lower efficiency. Decoupling capacitors were placed close to power pins of the microcontroller - +3.3V, VDD, AVDD (analog), VRF (radio) according to specifications provided by Silicon Labs.

The EZR32WG contains 2 operational amplifiers (op amps 1 and 2) compared to 4 on the EFM32WG. The op amps share pins with the analog-to-digital converter or analog comparator for their inputs and outputs. Op amp 1 has 5 possible output pins on the 230 model, but only 1 on the 330 model. This is due to the USB power and data pins replacing an analog comparator on the 330 model (PC12-15). Figure 3.4 shows the pinouts for both versions of the EZR32WG.

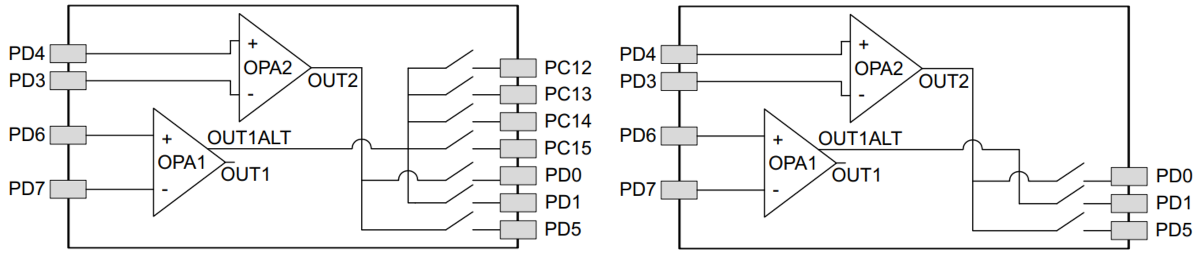

Figure 3.4: Operational Amplifier Pinouts for the EZR32WG 230 Model (Left) and 330 Model (Right).

The op amps can be configured in follower, inverting and non-inverting modes. A resistor ladder is present to allow configuration of different gains. Table 2.1 shows the gains that are available for the op amps.

R2/R1	**Non-Inverting Gain (1+R2/R1)**	**Non-Inverting Gain (dB)**	**Inverting Gain (-R2/R1)**	**Inverting Gain (dB)**
1/3	1.33	2.5	-0.33	-9.5
1	2.00	6	-1.00	0
5/3	2.67	8.5	-1.67	4.45
11/5	3.20	10.1	-2.20	6.8
3	4.00	12	-3.00	9.5
13/3	5.33	14.5	-4.33	12.7
7	8.00	18	-7.00	16.9
15	16.00	24	-15.00	23.5

Table 3.3: Operational Amplifier Gain Options Available.

The op amps can be cascaded to increase the gain further. They should both be configured as inverting amplifiers to minimize the noise due to its low input impedance compared to a non-inverting amplifier. Since both op amps are inverting, the output signal is not inverted, and no additional processing of the signal must be done at this stage.

Table 3.4 shows some of the gains available with cascaded inverting amplifiers.

Op Amp 1 Gain	Op Amp 2 gain	Total Gain Multiplier	Total Gain (dB)
-7	-4.33	30.3	29.6
-15	-2.2	33	30.3
-15	-3	45	33
-7	-7	49	33.8
-15	-4.33	65	36.2
-15	-7	105	40.4
-15	-15	225	47

Table 3.4: Cascaded Operational Amplifier Gain Options Available.

The microphone used is the AMM-3742-T-R by PUI Audio. It is an omnidirectional analog MEMS microphone and operates at 50 Hz-20 KHz. Power to the microphone is controlled by a P-channel MOSFET using a GPIO pin on the microcontroller.

The EZR32WG operates at 3.3V and a negative voltage is not available for the positive input of the op amp. Thus, a MAX-6070 1.8 V voltage reference was used to bias the input at 1.8V. This allows a voltage swing between 0.3 and 3.3V for the microphone.

The output of op amp 2 then goes into the ADC as well as an analog comparator to allow the device to awaken from sleep when sound is detected.

Figure 3.5 shows the entire microphone and op amp setup.

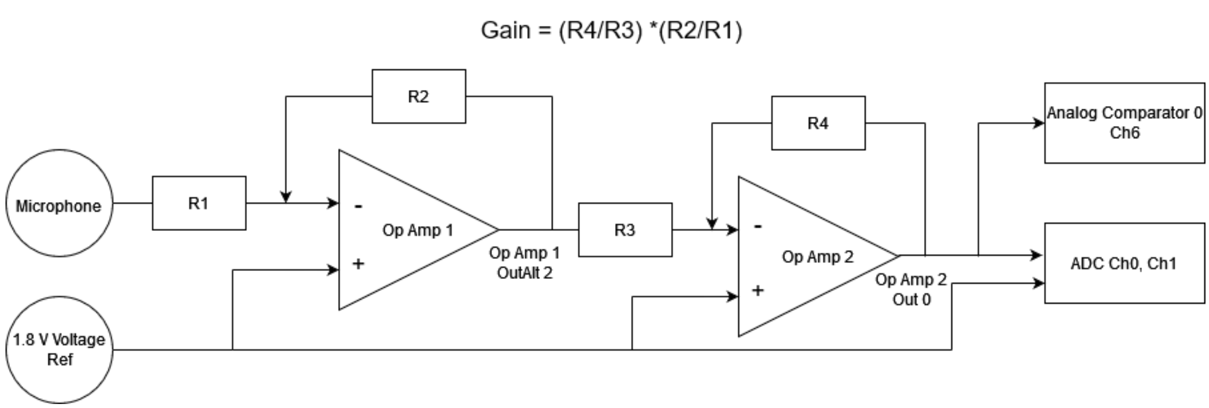

Figure 3.5: Operational Amplifier Configuration for the Bird Backpack.

There are two different radio powers that were used on the EZR32WG depending on the model. The +13dBm model uses the SI4460 EZRadio Module while the +20dBm model uses the SI4463 module. Silicon Labs provides application notes for the possible implementations of the radio depending on user needs. AN627 – "Si4X6X and EZR32 Low-Power PA Matching" and AN648 "Si4X6X and EZR32 High-Power PA Matching" describe the following implementations:

- Class E Split – Transmit and receive each have an antenna, matching network and filter.
- Class E Direct Tie – Transmit and receive share an antenna, matching network and filter.
- Class E TX/RX Switch – Transmit and receive share 1 antenna, matching network, and filter. A GPIO controlled switch determines which is active.

The goal of these application notes is to provide designers with schematics and components that have been verified, as well as try to optimize the following:

- Target the nominal output power of +10/+13/+16/+20 dBm.
- Minimize power consumption.
- Constrain peak voltage at the drain of the output devices.
- Comply with ETSI and FCC specifications.
- Get close to immunity against termination impedance variations.
- Low variation over temperature and supply voltage.

To minimize the area of the PCB, the Direct Tie layout was selected. There is a slight layout difference between the +13dBm and +20dBm radios as shown in Figure 3.6 and Figure 3.7. There is an additional inductor (LM3) in the low pass filter for the +20dBm layout. The labeling for the capacitors tied to ground in the low pass filter are also different: CM2 and CM3 vs CM1 and CM2. Components were provided for both wire-wound and multilayer inductors. Due to the higher Q value, wire-wound were selected despite being slightly more expensive: $0.13-0.25 vs $0.10-0.15 per unit depending on inductance value.

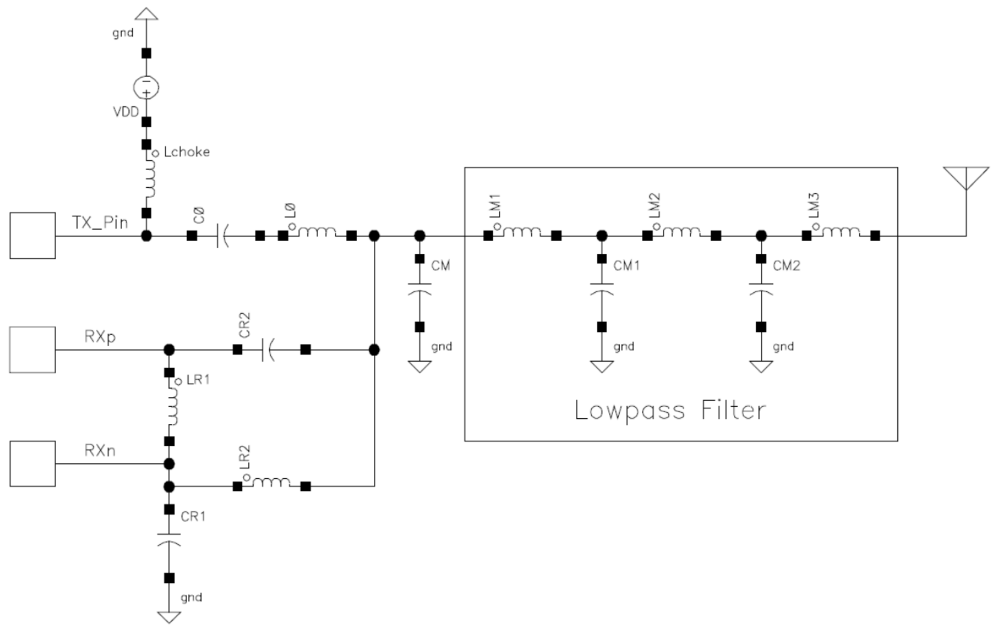

Figure 3.6: High Power Direct Tie Layout +20dBm.

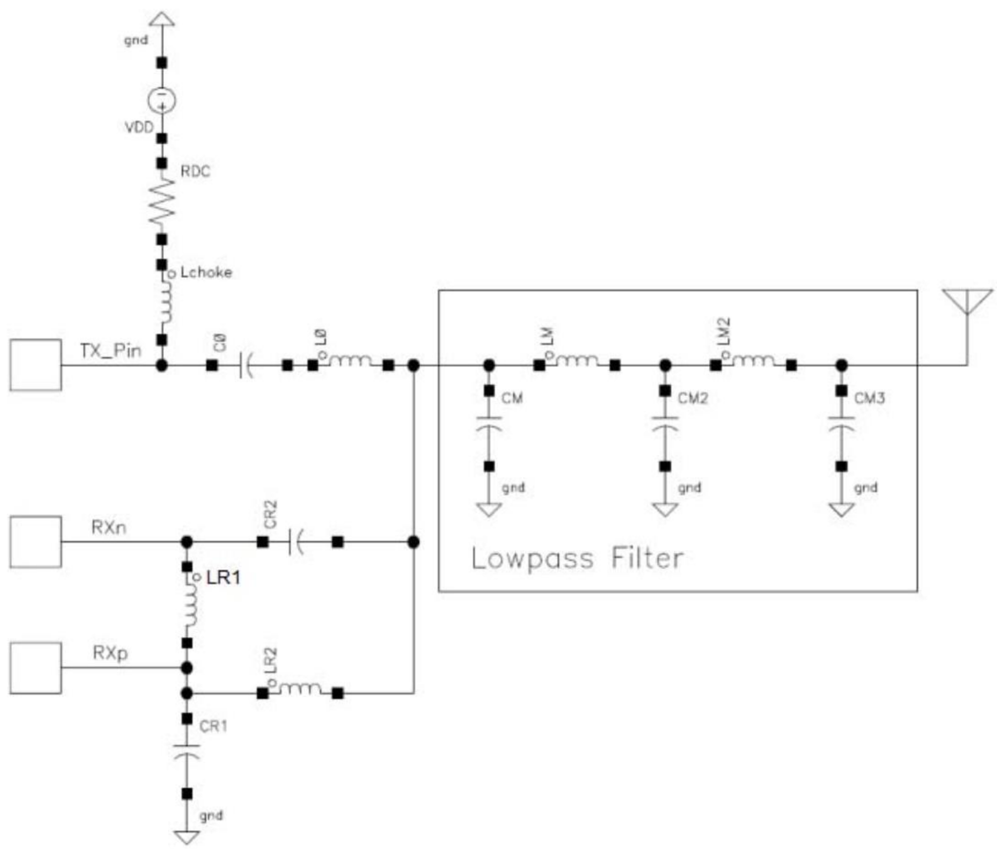

Figure 3.7: Low Power Direct Tie Layout +13dBm.

Table 3.5 shows the component values for each circuit for 915MHz for the receive portion.

Power (dBm)	LR1 (nH)	LR2 (nH)	CR1 (pF)	CR2 (pF)
+13	20	24	3.0	1.0
+20	18	22	3.0	1.0

Table 3.5: Receive (RX) Circuit Component Values.

Table 3.6 shows the component values for each circuit for 915MHz for the transmit portion and filter. Note that CM is an inductor instead for the +20dBm radio.

Power (dBm)	LC (nH)	C0 (pF)	L0 (nH)	CM (pF)	LM (nH)	CM2 (pF)	LM2 (nH)	CM3 (pF)	LM3 (nH)	Rdc
+13	120	3.6	19	2.7	9.1	5.1	9.1	2.7	0	0
+20	100	4.7	3.3	6.2nH	11	3.3	22	3.3	11	0

Table 3.6: Transmit (TX) Circuit Component Values.

The MicroSD Card is controlled in the same manner as the microphone: using a P-channel MOSFET to enable power to the device. A ferrite bead is placed in series with the power leading into the MicroSD card to eliminate high frequency noise. The MicroSD card is connected to an SPI interface on the microcontroller.

Due to time constraints and hardware sourcing difficulties, solar power was not implemented in the current design of the Bird Backpack.

3.2.4 Hardware Design and Assembly

Schematics and board layout were made using Autodesk Fusion 360 with an Education License. The final version of Fusion 360 used was 2.0.15509. Component footprints used include ones provided with Fusion 360 along with SnapEDA, Octopart and Ultralibrarian. The schematics follow design rules provided by Silicon Labs. For example, the radio schematic and component values were provided for different power levels and frequencies. This also included the physical layout of the components with rules such as nearby inductors must be at 90 degrees to one another, and stitching vias must be placed along the edges of the board. The microphone, voltage reference and MicroSD card schematics follow the design of the Audiomoth. The full schematics are shown in Appendix A and are divided into logical sections such as the radio, MicroSD card, power, etc.

The PCB design uses 4 layers to facilitate routing and is 1.2mm thick. The EZR32 microcontroller is placed roughly in the center with components placed as close as possible to their relevant pins. Both sides of the PCB are used for placing components. Vias use a drill size of 9.84252 mil (0.25mm) and diameter of 17.84252 mil (0.45mm). Vias are placed along the perimeter of the board and near any component that has one end connected to ground. Ground planes are maximized for each layer. Traces for GPIO and components are 7 mil wide, traces carrying power are 12 mil, and the trace for the radio is 24 mil. Each version of the board designed brought out GPIO pins to headers for testing purposes. The headers are placed along the perimeter of the board. The PCB was extended horizontally to add stencil holes of 98.4252 mil (2.5mm) diameter for alignment to allow applying solder paste.

The PCB was manufactured using JLCPCB and it took approximately 10 days to receive the completed PCB. The board was then assembled by hand using tweezers to place components. The top layer was done first, and a hot plate was used to reflow the solder. The bottom layer components were then placed, and a hot air gun used to reflow solder. The completed board was then visually inspected using a microscope for bridges. Any bridges identified were removed using a soldering iron and solder wick.

Once the board was deemed clear of bridges, power was connected, and a multimeter used to verify that the microcontroller powered on and 3.3V was detected across VDD and ground. Headers were then soldered on, and the development kit debugger connected. An LED was connected to a GPIO pin, and the board was programmed with a blink program. Subsequent programs were then used to test various features such as the crystal oscillators, radio, MicroSD card, microphone, and amplifier.

3.2.5 Base Station

The Base Station hardware is the SLWSTK6222A development kit. This consists of BRD4001A which is the mainboard that contains all the GPIO pins, LCD screen, debugger pins, push buttons, battery holder, USB port and network port. BRD4503B contains the EZR32WG microcontroller and the radio circuit and plugs into BRD4001A. An SMA connector is available to attach a radio antenna along with a USB connector for interfacing with the USB interface of the microcontroller. Both components are required for the Base Station to power on. The devices are shown in Figure 3.8.

Figure 3.8: The BRD4503B Board (Left) and BRD4001A Board (Right) Combined Make up the SLWSTK6222A Development Kit.

3.3 Software

The software for the Bird Backpack is developed using an IDE provided by Silicon Labs' called Simplicity Studio. It provides the libraries for all the various functions of the microcontroller such as the Clock Management Unit (em_cmu), Energy Management Unit (em_emu), and Operational Amplifier (em_opamp). Application Notes are available on Silicon Labs' site with example programs for these libraries. Example programs for the development kits are also available within Simplicity Studio to demonstrate functions such as transmitting and receiving radio messages.

3.3.1 Bird Backpack

The Bird Backpack operates as a fully wireless device. Upon powering on, it goes through an initialization process and starts up the oscillators, clocks, GPIO pins and the real time counter (RTC). At this point, the RTC does not have a time set. The radio is then initialized, and the device sends out a message requesting the time. If the Base Station is powered on and has its time set, it will receive the message containing the request for the time and send a reply with the time. If there is no reply received, the Bird Backpack will go to sleep for 5 minutes before waking up with an interrupt and trying again.

If the reply is received, the time is then set, and the device will go into the appropriate state. There are 2 primary states: deep sleep with the microphone, op amps, and analog comparator active waiting to wake it up, and deep sleep with a timer running. The Ridgway's Rail is most vocal 2-4 hour windows associated with sunrise and sunset. To allow for variance in the sunrise and sunset time throughout the summer in southern California, the microphone is active between 5am-8am and 5pm-8pm. When the timer ticks over to one of these times, the state changes and the microphone is enabled/disabled.

When the device is in S2 or S4, the microphone, op amps and analog comparator are enabled. The device is asleep in EM2 but will wake up via the analog comparator interrupt. The bias point is 1.8V and the comparator value can be set to 1.25 V, 2.5 V or scaled to VDD (3.3V) in increments of VDD/63. For testing, the voltage level was set to 2.5 V which represents a +0.7 V differential. The op amps were set to provide a 105x gain (+40.4 dB). If the analog comparator input detects a voltage above 2.5 V, the device will wake up and begin recording audio to the MicroSD card for 10 seconds. Once this is complete, the header is written, and the file is timestamped and named with the date. The device then goes back to sleep. The block diagram showing the general operation is shown in Figure 3.9, the state machine is shown in Figure 3.10.

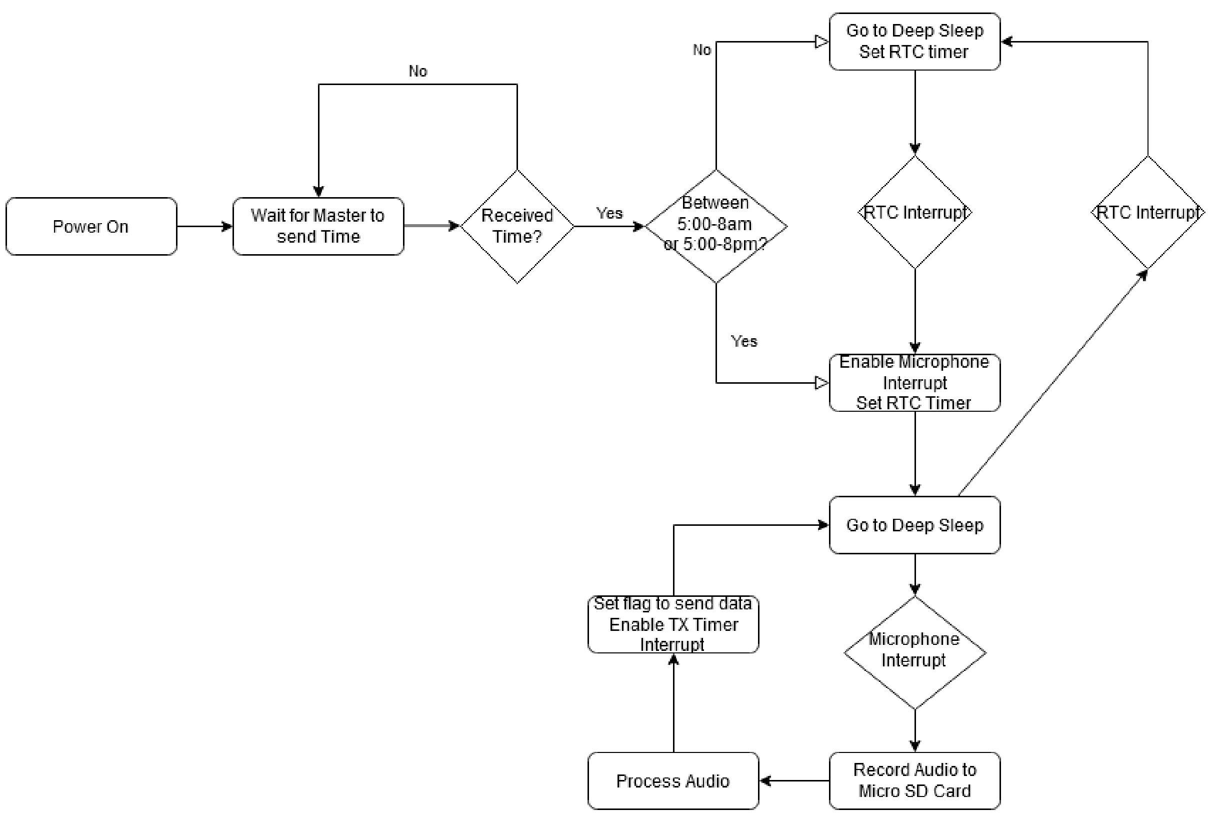

Figure 3.9: Block Diagram of Bird Backpack Operation.

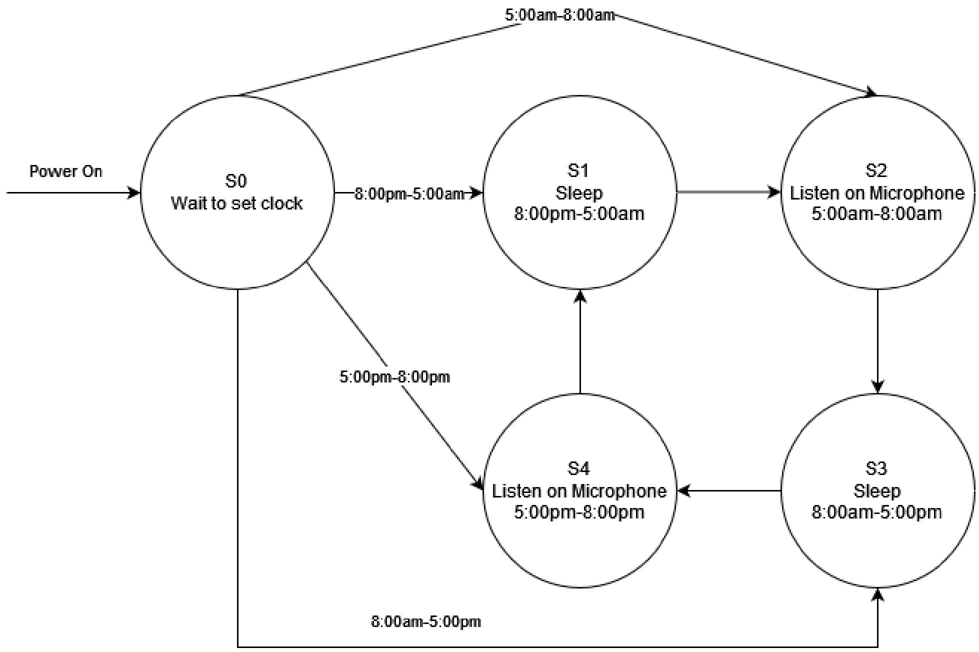

Figure 3.10: State Machine of Bird Backpack.

Audio recording occurs when the analog comparator detects a voltage above the 2.5 V threshold. A blank .WAV file is created, the ADC is initialized with the sample rate, acquisition cycles, clock divider and oversample rate. These values are set to 32 KHz, 16, 4, and 1 respectively. The ADC operates in differential mode using the 1.8 V reference and output of op amp 2.

Direct Memory Access (DMA) is used to move samples from the ADC to RAM. 2 RAM buffers of size 12 KB each are used. The ADC has 12-bit resolution for a total of 4096 discrete values. The effective range of the op amps and ADC are 0.3-3.3V at best, this gives increments of 0.732 mV per step. A WAV sample is 16-bit, each buffer can store 6144 samples, or 0.192 seconds of audio each. The DMA is operated in Ping Pong mode, the DMA transfers samples from the ADC to one buffer. When the buffer is full, the microcontroller wakes up and copies the buffer over to the MicroSD card, while the DMA transfers samples to the other buffer. The microcontroller then goes back to sleep. This continues until the requisite samples are collected and written to the MicroSD card: in this case, a 32 KHz sample rate recording for 10 seconds is 320,000 samples recorded. Using DMA to transfer the samples allows the microcontroller to run in EM1 instead of EM0 and save power. Figure 3.11 shows the operation of the audio recording.

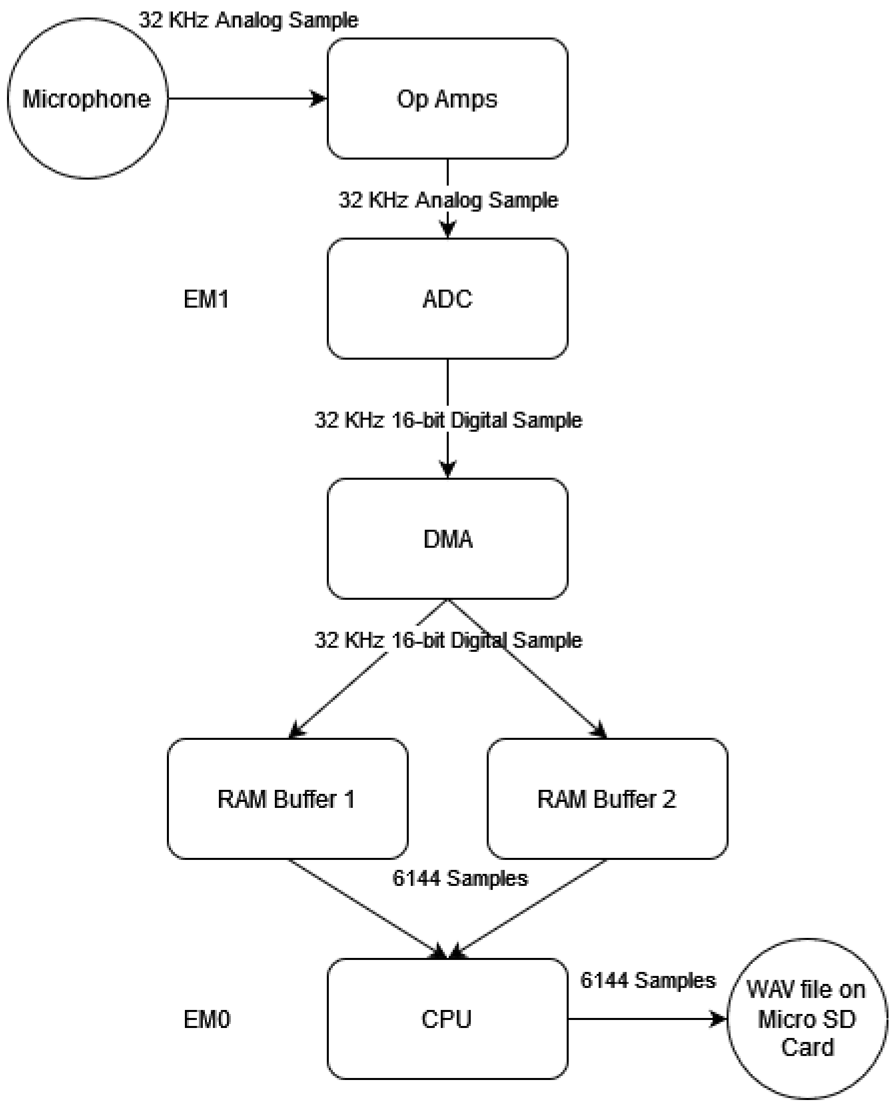

Figure 3.11: Recording Audio Operation.

The radio is configured within a framework in Simplicity Studio. The same settings were used as in the example programs for the SLWSTK6222A development kit board. The radio operates at 915 MHz and uses 2-Frequency Shift Keying (2FSK) modulation. Message packets are encoded as character arrays. Thus, it is possible to send messages in plain text. A simple messaging scheme has been set up as shown in Table 3.7.

Packet[0]	**Packet[1]**	**Packet[2]**	**Packet[3+]**
Destination (1 byte)	Source (1 byte)	Message Type (1 byte)	Message (variable)

Table 3.7: Layout and Contents of Radio Message Packet Excluding Headers.

Each device is assigned an ID and is used in the Source and Destination fields. The Message Type is the type of request/response being sent, while the Message is the contents associated with the message type.

Two types of messages have been defined:

- The request for time has a message type of 1 and does not have any message content.
- The reply for time has a message type of 2 and its contents consists of the time variable.

Time is represented as a 32-bit signed integer. To fit the time variable into 8-bit values, the 32-bit value is right shifted and cast into an 8-bit signed integer. Thus, 24 bits for Packet[3], 16 bits for Packet[4], 8 bits for Packet[5] and 0 bits for Packet[6]. When the reply is received, the time is calculated from the message by left shifting the message the same number of bits and summing up the values.

For example, the Base Station is assigned the ID of 1 and the Bird Backpack has ID 2. The Bird Backpack is powered on and needs to request the time. It sends a packet with contents "121" using the radio. The Base Station receives this message and decodes it as a message for it: Packet[0] == 1, it came from ID 2: Packet[1] == 2 and the message is a request for the time: Packet[2] == 1. The current time of the Base Station is 2:00:00pm, March 20, 2023. This gives a timestamp of 1679346000 in base 10 and 0110 0100 0001 1000 1100 1001 0101 0000 in base 2. The Base Station creates a reply message with Packet[0] = 2, Packet[1] = 1, Packet[2] = 2. Packet[3] is the timestamp right-shifted 24 bits to get 0110 0100 (100 decimal), packet[4] is right-shifted 16 bits to get 0001 1000 (24 decimal), packet[5] is right shifted 8 bits to get 1100 1001 (201 decimal) and packet[6] is the last 8 bits for 0101 0000 (80). The entire message is then sent and can be represented as "212d╔`". 24 encoded as a character is the cancel character.

The Bird Backpack then receives the messages and sees that the message is intended for it from the first 3 bytes. The time is then reconstructed from Packet[3-6]. Packet[3] is left-shifted 24 bits to get 1,677,721,600 in decimal, Packet[4] is left-shifted 16 bits to get 1,572,864, Packet[5] is left-shifted 8 bits to get 51,456. These are then summed up with 80 to get the timestamp of 1,679,346,000 which is used to set the date and time.

This entire process is shown in Figure 3.12.

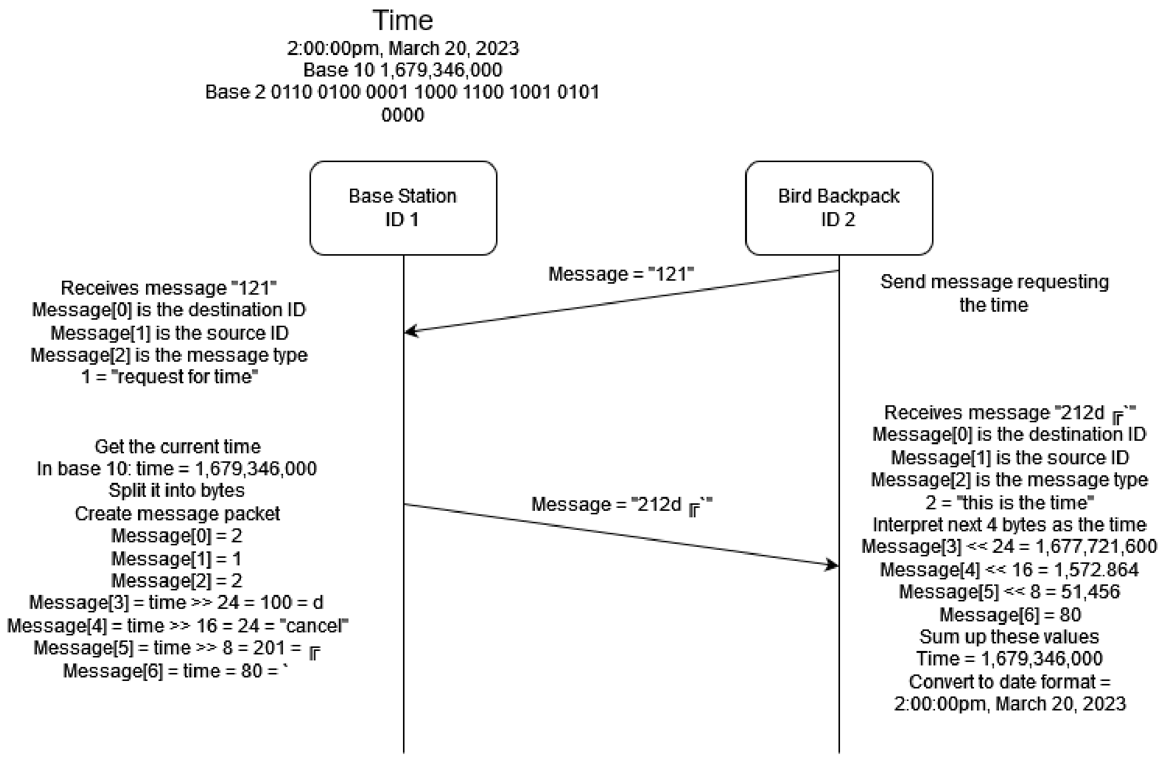

Figure 3.12: Process of Bird Backpack Requesting the Time From the Base Station via Radio.

3.3.2 Base Station

The Base Station makes use of the AudioMoth code that deals with interfacing with the Windows applications via USB. This code is being used because it provides all the functionality for communications sent between the device and the Windows application. Also, by using this code, the Base Station will appear as an AudioMoth device that can be configured with the AudioMoth Time App.

Upon powering on, the Base Station will wait for the time to be set. Once the time has been set, the Base Station then initializes the radio and listens for incoming messages to process as shown in Section 3.3.1. Incoming messages and replies are printed out on the LCD screen for feedback. Figure 3.13 shows the state machine of the Base Station while Figure 3.14 shows the interaction of the Base Station with the Windows Time Application and Bird Backpack.

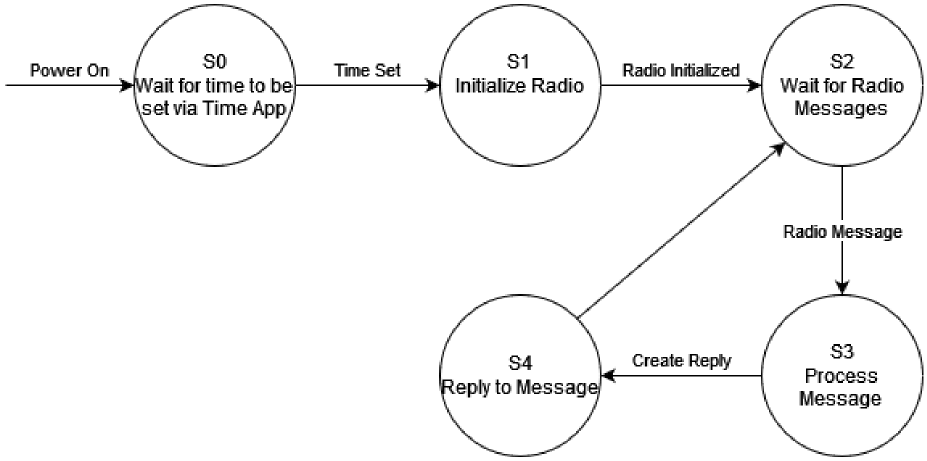

Figure 3.13: State Machine of Base Station.

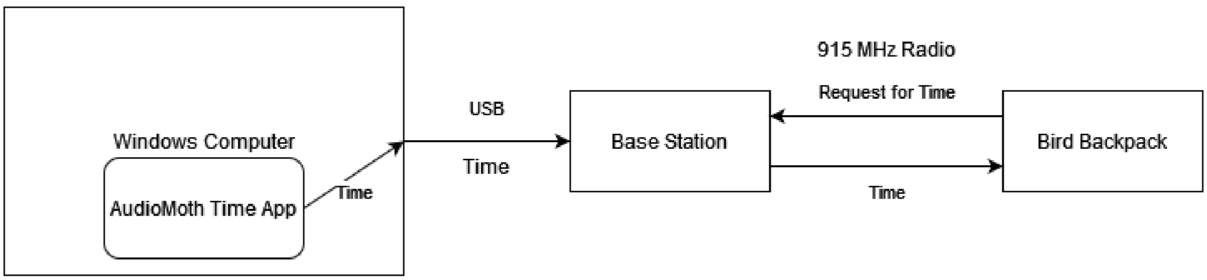

Figure 3.14: Block Diagram of Base Station Interactions With the AudioMoth Time Application and Bird Backpack.

To summarize, the design for the Bird Backpack consists of a customized PCB to have greater control over the hardware, layout, and dimensions of the device. The EZR32WG microcontroller was selected because it is close in functionality to the EFM32WG used on the AudioMoth while also having a radio and was also in stock. This allows leveraging the AudioMoth software for the audio recording, as well as the USB interface for the Windows applications. Modifications to the AudioMoth software were required due to hardware differences in pinouts and the lack of the EBI in the EZR32WG. The radio circuit was configured for 915 MHz and a simple communication scheme was designed to send messages to and from the Base Station which made use of the SLWSTK6222A development kit.

Chapter 4: Results

This chapter goes over the results and findings of various aspects of the Bird Backpack's design and functionality. These include the general functionality of the software, radio range, power usage, audio capture, the size and weight of the device, and testing small solar panel power generation.

4.1 Functionality

The Base Station can communicate with the AudioMoth Time App successfully. It appears as if it is an AudioMoth device and the time can be set by clicking on "Set Time". The time on the Base Station gets updated and its current time is shown on the AudioMoth Time App. The USB cable connected to BRD4503B can be disconnected and plugged back in later. The time showing on the AudioMoth Time App will update showing the Base Station's time operating as expected and is shown in Figure 4.1.

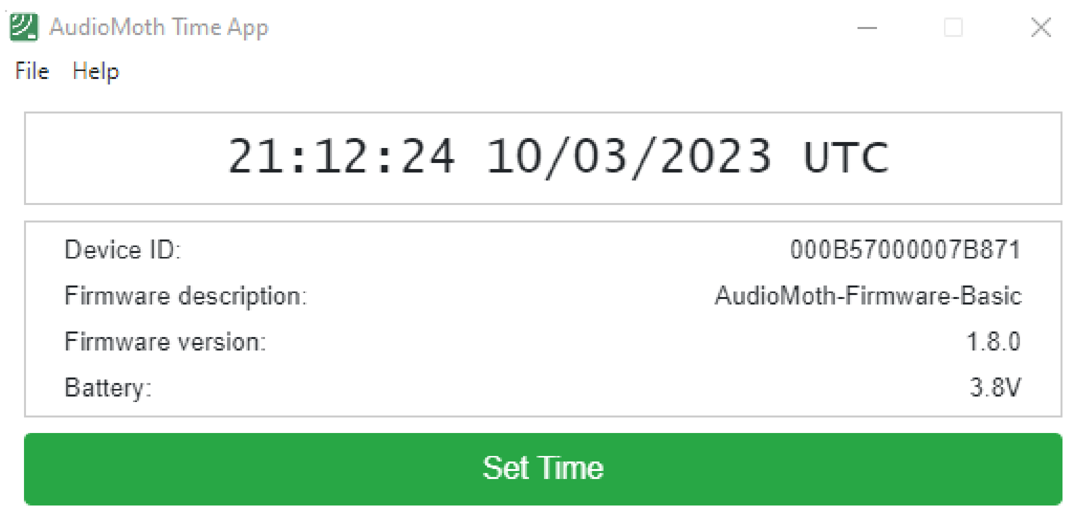

Figure 4.1: AudioMoth Time App Setting the Time of the Development Kit Behaving as the AudioMoth.

The AudioMoth interfaces with the Windows Applications with a switch on its board. The switch needs to be changed from USB to Default to disable the USB interface and enter the audio recording state. A latching switch is not present on either BRD4503B or BRD4001A. Pressing the switch can be simulated by using a jumper wire to set the USB detect pin high when setting the time, and then connecting it to ground to enter the radio initialization state. The time and date are then printed on the LCD screen as shown in Figure 4.2.

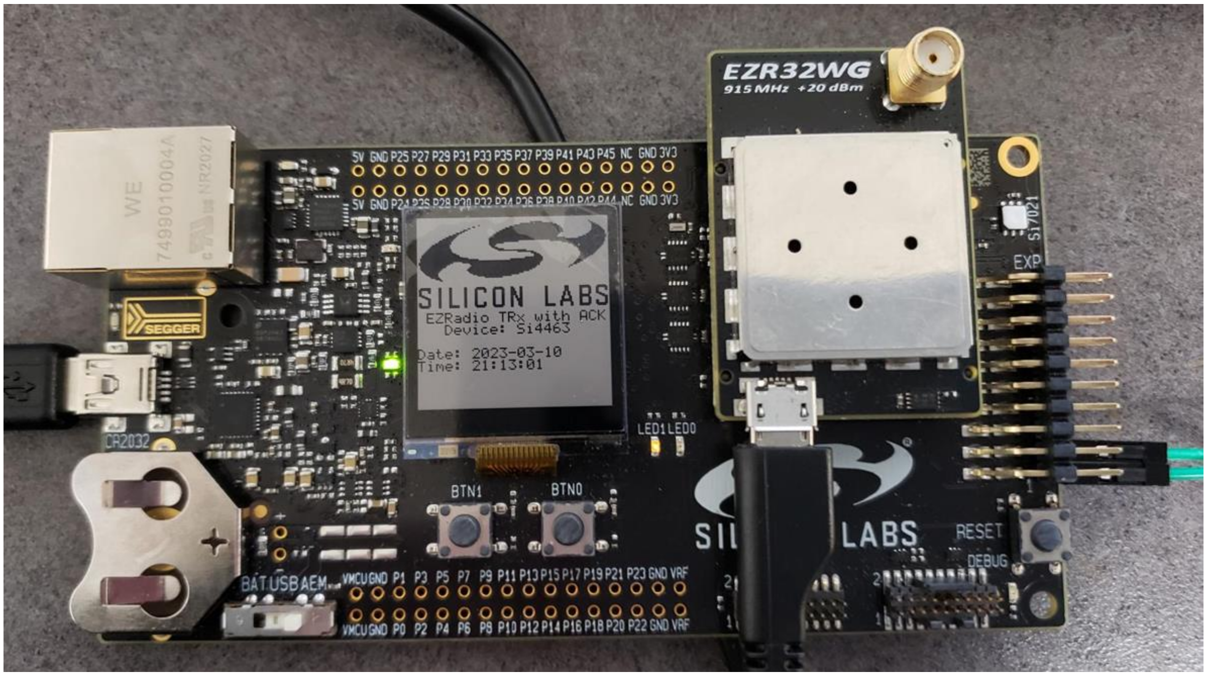

Figure 4.2: The Development Kit Printing the Time on the LCD Screen After Having the Time set.

The Bird Backpack powers on and then sits in a loop sending a message via radio to request time. When the time is received, the Bird Backpack goes into its appropriate state based on the time. If the state is S2 or S4 (5:00am-8:00am or 5:00m-8:00pm), the op amps, microphone and analog comparator are enabled, and the Bird Backpack goes into EM2. When a sound is detected that goes above the voltage amplitude threshold, the Bird Backpack goes into EM1 and initializes the DMA to record audio. The audio records for 10 seconds, creates a WAV file, and then goes back to EM2. When S1 or S3 are reached, the microphone and amplifiers are disabled, and no recordings take place until the time causes the state to go back to S2 or S4. There is an issue with the radio and audio recording in the current implementation. Both make use of DMA and when the radio on the Bird Backpack is active, the DMA does not trigger for the audio recording.

4.2 Radio Range Testing

The EZR32WG microcontroller comes with 4 different transmit power levels for the radio: +10dBm, +13dBm, +16dBm and +20dBm. Due to supply chain issues, only the +13dBm and +20dBm radios were purchased and tested. The range experiments were performed in the Rathdrum prairie in North Idaho: a large, flat, mostly undeveloped, grassland area that is near an airport. In comparison, Tijuana Slough in southern California where the Rails are most abundant has a similar flatness, is marshy and also has an airport nearby. This provided the closest conditions to where the Bird Backpack will be deployed.

The software used for these tests was the SLWSTK6222A_ezradio_trx_ack example program provided with Simplicity Studio. This program configures the development kit to send a single packet upon pressing one of the buttons, or repeated packets upon pressing another button. A second development kit or custom device is also programmed to do the same. In this scenario, one device sends out a message. The other device receives it and then sends an ACK (acknowledgement) message which indicates that it received the message. The sending device then receives it and prints out the message on its LCD display. The program can still be used in the same manner on the custom device as long as it is the receiver sending the ACK.

The tests were performed as follows:

- Set up the sender and receiver with the trx_ack program.
- Power the devices with a battery or USB power bank.
- Connect an antenna of varying size and gain to both devices.
- Drive out to the Rathdrum prairie with someone accompanying.
- Get dropped off by the side of the road with one device, while the other is in the vehicle.
- Call up the person on the phone.
- Point the antenna (if directional) at the vehicle as it drives away.
- Press the button to send a single packet approximately every 2 seconds and wait for a response to be received and printed on the LCD screen.
- Let the driver know when CRC errors and/or no more responses are being received.
- The driver takes note of the number of utility poles passed and any landmarks at the point where the responses stopped.
- The driver turns around and packets are sent again and noted when they are received.
- Use Google Maps' measurement tool to approximate the range of the radio.

The following configurations were tested:

- 2x +20dBm radios with +3dBi omnidirectional antennas on both.

- 2x +20dBm radios, one with a +10dBi 434 MHz 32” long Yagi antenna and the other with a 78 mm long wire antenna (this was stranded 16AWG copper wire with insulation, unknown dBi, 78 mm is the quarter wavelength for 915 MHz).
- 1x +20dBm radio with a +10dBi Yagi antenna, 1x +13dBm radio with +3dBi omnidirectional antenna.

Table 4.1 shows the results of each of these tests.

Radio and Antenna Configuration	**Maximum Range (meters)**
2x +20dBm radio, +3dBi omni	500
2x +20dBm radio, 1 +10dBi Yagi, 78 mm wire	750
+20dBm radio with +10dBi Yagi, +13dBm radio with +3dBi omni	340

Table 4.1: Maximum Range of Radio With Various Setups.

Figure 4.3 shows the range of 750m overlaid over the map of Rail locations in Tijuana Slough. This indicates the maximum range to receive messages using the 32” Yagi antenna. This can be used to aid researchers when they go out into the field with the Base Station and determine walking paths to travel to maximize the likelihood of collecting transmissions from Bird Backpacks.

Figure 4.3: Map of Tijuana Slough With Projected Radio Range of Rails, Modified From [8].

4.3 Power and Current Usage

There are multiple operations that need to have power and current consumption measured. These include recording audio, transmitting over radio, receiving over radio, as well as the current consumed in each of the 5 power states – EM0 through EM4. The lifetime of the device can then be approximated based on a 100mAh battery and various amounts of usage. A 100mAh battery weighs approximately 3g [17]. The measurements were taken for both the development kit as well as the version 0.4 of the Bird Backpack. Table 4.2 shows the average current draw and potential battery life if the action was performed continuously.

Mode/Action	Current (μA)	Battery Life With 100 mAh battery (days)
TX +20dBm 1 packet	70,000 (14 ms long)	0.06
RX	29,000	0.14
Recording 10s Audio at 32 KHz	14,000	0.30
EM0 – Inside a loop doing nothing	5,100	0.82
EM1	1,770	2.34
EM2	**1**	**4776**
EM3	0.15	27,365
EM4	0.00273	1.5 million

Table 4.2: Current Draw of Primary Functions of the EZR32WG Microcontroller.

As the energy mode increases (goes into a deeper sleep), fewer peripherals are available. EM2 is the lowest mode that can be used while maintaining the real time counter for timekeeping. Transmitting over the +20dBm radio consumes a large amount of current, but its duration is short and is not going to be performed frequently.

Recording audio is the most current intensive operation due to its expected frequency and duration. The battery is expected to last a little over 7 hours of total recording. This length can be increased by several parameters such as reducing the recording length, reducing the sample rate, and increasing the amplitude threshold of the analog comparator before triggering a recording. All of these will result in lower audio quality and/or missed vocalizations and will thus have to be balanced according to future requirements. Another way to increase battery life is by adding a solar panel to charge the battery during the day.

4.4 Audio Capture

A μMoth was obtained and the same audio recording of the Rails from Figures 10 and 11 was played back to the μMoth and Bird Backpack. Approximately 25 seconds of audio was recorded at 32 KHz. The speaker used came from a laptop computer at 50% volume and the microphones were placed about 15 cm away from the source. The μMoth was set to "medium" gain (30.6 dB). The test was performed in a residential bedroom with additional ambient noise. Figure 4.4 shows the waveforms recorded from each device. There is a "glitch" that is intermittently recorded on the Bird Backpack and appears as the vertical lines that mainly reach the maximum magnitude. This is suspected to be when DMA transfers the audio samples to the onboard SRAM as these "glitches" don't appear on an oscilloscope when measured at the output of op amp 2. Figure 4.5 shows the spectrogram of both recordings. It appears that the µMoth microphone (Knowles SPM0408LE5H) has a greater sensitivity and gain than the AMM-3742-T-R used on the Bird Backpack.

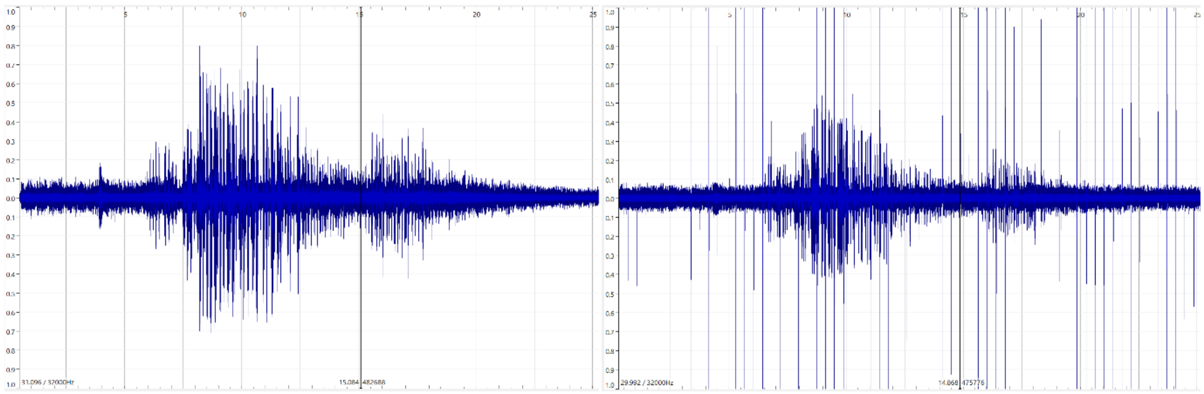

Figure 4.4: Comparison of Waveform From the µMoth (Left) and Bird Backpack (Right).

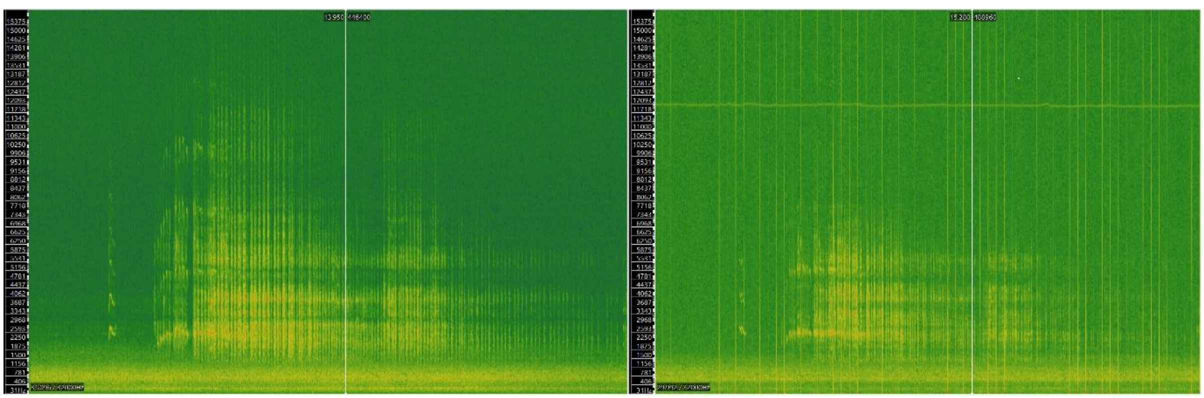

Figure 4.5: Comparison of Spectrograms From the µMoth (Left) and Bird Backpack (Right).

4.5 Size and Weight

Four versions of the Bird Backpack were designed, manufactured, and assembled throughout development. The dimensions and weights are larger than what a production model would be due to the addition of headers to allow for ease of testing I/O of the microphone and op amps, programming via the debugger and GPIO pins for feedback via LEDs. Additionally, a USB connector was added on each version to power the device. Figure 4.6 shows the top of version 0.4 of the Bird Backpack.

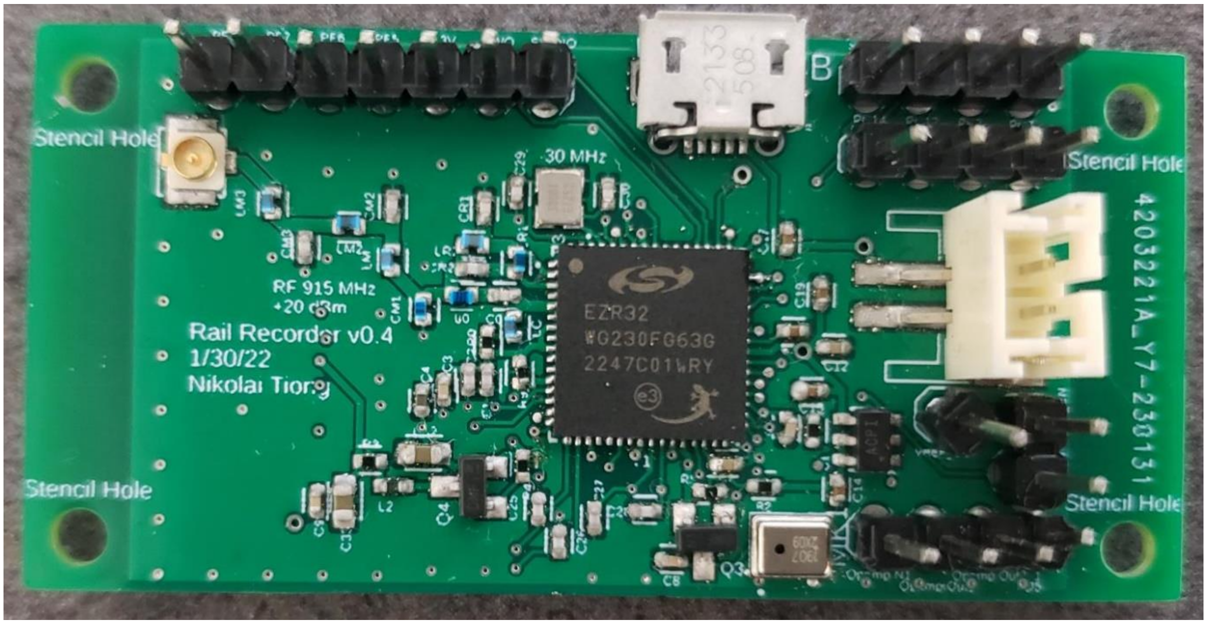

Figure 4.6: Top of Version 0.4 of the Bird Backpack.

Table 4.3 lists the unassembled and assembled weight of the devices without the antenna or battery along with dimensions of the AudioMoth, µMoth and development board.

Version	**Note**	**Dimensions (mm x mm)**	**Board Weight (g)**	**Weight Assembled (g)**
0.1	First prototype	62.79 x 42.5	7.12	8.88
0.2	Changed coin cell holder to JST, brought more GPIO pins to headers	67.56 x 38.64	7	10
0.3	Brought all GPIO pins to headers	67.56 x 38.64	7	10
0.4	Removed most GPIO pins, compacted all components	56.29 x 27.38	4.1	7
AudioMoth		58 x 48	N/A	80
µMoth		32 x 24	N/A	5
Dev Kit		43 x 30	N/A	6

Table 4.3: Dimensions and Weight of Bird Backpack Versions, AudioMoth, µMoth and the Development Kit.

4.6 Solar Generation

The final prototype designed did not incorporate the solar panel or charger. Testing with some small solar panels was performed with an Adafruit BQ24074 solar charger. A multimeter was used in series with the solar panel and the charger to measure the current. The solar panels were tested outside on a sunny day around 12pm during the summer. 3 different positions were tested – direct sunlight, indirect sunlight and facing away from the sun. Table 4.4 summarizes the results.

Solar Panel Model	Specs	Light	Avg Current (mA)	Charging?
KXOB25-02X8F-TB-ND	26.3mW, 5.53 VOC, 6.3 mA ISC	Direct	2.9	Yes
		Indirect	1.85	Yes
		No sunlight	0.02	No
KXOB25-01X8F-TB	24.5mW, 5.53 VOC, 5.9 mA ISC	Direct	2.5	Yes
		Indirect	1.85	Yes
		No sunlight	0.09	No

Table 4.4: Current Generated by Solar Panels.

Due to the Bird Backpack being attached to a bird that is not likely to be standing still, the solar generation will be difficult to estimate since it will be moving around, and the bird (and hence the Bird Backpack) will rarely be exposed to direct sunlight due to their secretive habits.

4.7 Conclusion

This chapter went over the results of various experiments testing the functionality of the Bird Backpack. The Base Station successfully behaves as the AudioMoth and can interface with the Windows AudioMoth Time Application to set the time and send the time out via the radio. The Bird Backpack can receive the time via the radio and go into the appropriate state. It can record audio to the MicroSD card but has an audio glitch as well as radio and DMA issues that need fixing. Incorporating the AudioMoth code for the Bird Backpack was much more difficult than expected with multiple changes/additions/omissions required including setting the pin mode for the MicroSD card enable, the RAM buffers for the DMA, and ensuring that the correct filters were applied.

Power usage testing shows that it is comparable to the AudioMoth and adding a small solar panel to recharge the battery can be viable. The size of the Bird Backpack is currently slightly larger and heavier than the μMoth, but this is partially due to the radio circuit and header pins. Table 4.5 compares the features of the AudioMoth and the Bird Backpack.

Feature	**AudioMoth/ μMoth**	**Bird Backpack v0.4**
Devices	1	2 – Bird Backpack and Base Station
Dimensions (mm x mm)	58 x 48 / 32 x 24	56.29 x 27.38
Area (mm^2)	27.64 / 7.68	15.4
Weight without battery (g)	80 / 5	7
Microphone	10 KHz MEMS microphone with 20 dB gain	20 KHz MEMS microphone
Amplifier Gain (dB) – can be adjusted in the program	27.2/28.7/30.6/31.6/32	40.4
SRAM	32 KB onboard + 256 KB External	32 KB onboard
Radio	None	915 MHz
USB	Yes	Bird Backpack – no Base Station - yes
Available GPIO Pins	15	23

Table 4.5: Comparison of AudioMoth/μMoth and the Bird Backpack.

Chapter 5: Future Work

There is still significant scope for future work that can be done on the design for both hardware and software. This chapter is a list of the improvements that can be made.

5.1 PCB Design

The latest design of the PCB: version 0.4, has several components that are not ideally suited for the final product. This was due to the difficulty of obtaining components. A 20 KHz microphone is currently being used, a better value would be 10 KHz. The JST connector faces straight up when placed on the PCB. This would cause the battery connector to poke into the Rail's back or upwards depending on the side of the board on which it is placed. The connector should be at 90 degrees so that it points outwards along the Rail's back. The MicroSD card slot is a hinge type which wears out after multiple insertions and removals of the card. A spring-loaded holder would be more durable. The micro-USB connector is used to provide an alternative source of power since the 230 model of the EZR32 doesn't have a USB interface. This can be removed in the final design when it only needs to be powered by the battery. All the pins brought out to headers can be removed in the final design or brought out to test pads. The only ones that must remain are +3.3V, SWO, SWDIO, SWCLK, RESETN and GND to program the device initially. A possible alternative is to omit these pins as well and program the microcontroller on another board, desolder it and place it onto the production board during assembly. With the previous changes, the board layout will also need adjustment, but will result in a much smaller footprint and weight. Making these adjustments to version 0.4 reduces the dimensions from 56.29 x 27.38 mm to 42.95 x 23.26 mm without major layout changes. This reduces the area from 15.4mm^2 to 10mm^2 and would reduce the weight of the PCB from 4.1g to approximately 2.7g.

5.2 Fix Radio/Audio Recording Issue and Audio Glitch

There is the issue with the radio and audio recording. Both make use of DMA. When the radio is initialized, the DMA interrupt for the audio recording will not trigger. The radio uses a library called "dmadrv" while the audio recording uses the base library "em_dma". The radio is initialized first using the function ezradioInit, while the audio recording function reinitializes the DMA every time a recording occurs. A possible solution to this is to rework the audio recording function to use dmadrv and its callbacks. This may also fix the audio glitches that occur periodically in the recordings.

5.3 Waterproofing the Device

There are several ways that can be utilized to waterproof the electronics and prevent damage when the Rails go into water. These include applying a protective coating on the device. However, this will need to be reapplied when retrieving and changing the battery as the seal will be broken at the JST connector. Another way would be to use a heat shrink wrap around the device once the battery is connected. The wrap would be a one-time use and add some weight to the overall backpack.

5.4 Using the EFM32WG and SI446X

Use the EFM32WG microcontroller and the SI446x RF transceiver instead of the EZR32WG. This would allow the addition of the external SRAM module via the EBI. There would also be better compatibility with the AudioMoth software. The downside to this is that there will be an increase of the PCB board footprint leading to a weight increase as well.

5.5 Use a Software Defined Radio

A Software Defined Radio (SDR) with a device such as a Raspberry Pi for communication with the Bird Backpack could be used instead of the development kit and reduce reliance on AudioMoth software. This would require research on how to implement the modulation of the EzRadio and packet structure such as the preamble, sync word, and CRC.

5.6 Add a Solar Panel and Charger

Assuming an average of 1mA generation with 10 hours of sunlight during the summer would provide 10mAh, or 10% of the battery capacity per day. This would allow the device to be deployed for longer periods before having to capture the Rail and retrieve the MicroSD card. The drawbacks to this are the addition of size and weight to accommodate the additional hardware as well as the cost for the components.

5.7 Researching Flexible PCB Manufacturing

The current PCB design is a standard flat rectangular one. A flexible PCB would potentially reduce the weight of the board, as well as fit the back of the Rail better. However, pricing for a flexible PCB is much greater than for a standard PCB. Appendix shows the cost of PCBs. A quantity of 5 standard PCBs from JLCPCB costs $8.00, while 5 flexible PCBs from PCBWAY costs $422. Due to this cost difference, the flexible PCB should only be used for the final design.

5.8 Adding Additional Message Types Using the Radio

The toolchain for sending and receiving messages between the Bird Backpack and the Base Station has been established. It is possible to add additional types of messages such as recording parameters – gain, frequency and recording length. The encoding scheme can be modified to allow for more types of messages, or fewer to reduce the length of the transmissions and power used. Currently, the transmissions are not encrypted, and this should be done to prevent eavesdropping on the messages since the messages can be sent in plaintext characters.

5.9 Additional Signal Processing

Currently, the Bird Backpack does not do any additional signal processing to the audio recordings. The ARM Cortex M4 contains a Digital Signal Processing (DSP) unit that can perform operations such as Fast Fourier Transforms (FFTs). This would allow identification of dominant frequencies in the audio recordings. These could then be sent to the Base Stations in a message. There would be an additional power cost to this operation, which would need to be measured and weighed against the benefit of knowing this information prior to retrieval of the MicroSD card.

Chapter 6: Conclusion

The Light-Footed Ridgway's Rail is an endangered water bird whose population has been monitored by aural surveys and relies on the surveyor's hearing and call identification expertise to count the number of Rails in a marshland. This type of counting is problematic due to double counting, an unknown number of false negatives (i.e., not all birds vocalize), and the birds habituate to call playbacks. A device capable of recording their vocalizations, a Bird Backpack, that can be attached to their backs and would allow researchers to directly estimate the proportion of false negatives (i.e., detection probability) during rangewide survey efforts. GPS transmitters have already been used successfully on the Rails and can successfully track their movements. On-board audio recording devices for animals is a developing field with small recording devices being tested, as well as customized embedded systems being developed such as the AudioMoth. Key requirements for the Bird Backpack are as small as possible, has a total weight of less than 9g and uses as little power as possible. The researchers also want to be able to learn more about the functions of different types of calls based on the recordings from Bird Backpacks.

A prototype wireless audio recorder small enough to fit on the back of the Ridgway's Rail was designed and tested. A customized solution was selected due to the ability to control the size, layout, and weight of the entire device. The solution also made use of a microcontroller in the same family as the AudioMoth to reuse and adapt the code that the authors have open-sourced. The semiconductor shortage throughout 2021 and 2022 limited the microcontroller choice for the Bird Backpack to the EZR32WG230 model, which has similarities to the EFM32WG and contains a radio module capable of transmitting at 915 MHz. The development kit for the EZR32WG was designed as a Base Station which can accompany researchers while they go out into the Rail's habitat.

A hardware and software toolchain was developed making use of part of the AudioMoth software library and wireless capabilities of the EZR32WG microcontroller. The Base Station made use of the code that interfaces with a Windows application developed by the authors to set the time. The Bird Backpack modifies the audio recording code due to the EZR32WG lacking the EBI that the EFM32WG has.

The Bird Backpack is capable of recording audio triggered by an amplitude threshold, without requiring the SRAM module the AudioMoth uses, while using a comparable amount of current when recording at 32 KHz: 14 mA vs 13 mA. The Bird Backpack operates on a 24-hour clock cycle and only records during sunrise and sunset hours to reduce current draw to 1 μA outside of the recording times. There is an ongoing issue with the audio recording having periodic glitches that can be heard and seen when visualizing the recordings' waveforms and should be the first issue to investigate and fix.

A simple wireless communication protocol was developed to allow the Base Station to set the time of the Bird Backpack. The message format was designed in a manner to allow additional types of commands to be created. The range of the radio is up to 750 meters which will allow researchers to cover a larger area when collecting data in the field. There is also an ongoing issue with the radio initialization causing an issue with the audio recording to not occur and should be the second issue to investigate and fix. It is suspected that both issues are related to DMA since both the radio and audio recording make use of it.

In comparison with the μMoth, the Bird Backpack is currently twice the area – 15.4mm^2 vs 7.68mm^2 and weighs 7g vs 5g without a battery and antenna. This is above the desired weight when including a battery weighing 3g and antenna weighing 1g giving a total weight of 11g. However, there is potential to reduce the Bird Backpack's size and weight further in future revisions by removing headers and the USB connector.

Once the radio and audio recording issues are sorted out and the size is reduced on a future version, the Bird Backpack will have similar capabilities to the AudioMoth/μMoth while having the flexibility of a wireless radio for communication.

References

[1] R. Zembal, S. M. Hoffman, C. Gailband and J. Konecny, "Light-footed Ridgway's (Clapper) Rail Management, Study, and Zoological Breeding in California 2016 Season," California Department of Fish and Wildlife, Sacramento, 2016.

[2] E. D. Stein, K. Cayce, M. Salomon, D. L. Bram, D. De Mello, R. Grossinger and S. Dark, "Wetlands of the Southern California Coast – Historical Extent and Change Over Time. Technical Report 826," Southern California Coastal Water Research Project Authority, Costa Mesa, CA, 2014.

[3] R. Zembal, S. M. Hoffman, J. Konecny and B. Sabiston, "Light-footed Ridgway's (Clapper) Rail in California 2018 Season," US Fish and Wildlife Service and California Department of Fish and Wildlife, Sacramento, 2018.

[4] T. A. Marques, L. Thomas, S. W. Martin, D. K. Mellinger, J. A. Ward, D. J. Moretti, D. Harris and P. L. Tyack, "Estimating animal population density using passive acoustics," *Biological Reviews*, vol. 88, no. 2, pp. 287-309, 2013.

[5] K. M. Strickfaden, D. A. Fagre, J. D. Golding, A. H. Harrington, K. M. Reintsma, J. D. Tack and V. J. Dreitz, "Dependent double-observer method reduces false-positive errors in auditory avian data," *Ecological Applications,* vol. 30, no. e02026, 2020.

[6] R. Zembal, S. M. Hoffman and J. Konecny, "Light-footed Ridgway's (Clapper) Rail in California 2017 Season," California Department of Fish and Wildlife, Sacramento, 2017.

[7] K. A. Sawyer, C. J. Conway, T. Katzner and E. J. Harrity, "Survival and Movement of Light-footed Ridgway's Rails. 2020 Annual Report," Idaho Cooperative Fish and Wildlife Research Unit, Moscow, ID, 2020.

[8] C. J. Conway, T. Katzner, E. Harrity and K. Sawyer, "Movement and Survival of Light-footed Ridgway's Rails, Project Update, January 2021," Moscow, ID, 2021.

[9] L. F. Gill, P. B. D'Amelio, N. M. Adreani, H. Sagunsky, M. C. Gahr and A. ter Maat, "A minimum-impact, flexible tool to study vocal communication of small animals with precise individual-level resolution," *Methods in Ecology and Evolution,* vol. 7, no. 11, pp. 1349-1358, 2016.

[10] A. Ter Maat, L. Trost, H. Sagunsky, S. Seltmann and M. Gahr, "Zebra Finch Mates Use Their Forebrain Song System in Unlearned Call Communication," *PLos ONE,* vol. 9, no. 10, 2014.

[11] D. Stowell, E. Benetos and L. F. Gill, "On-Bird Sound Recordings: Automatic Acoustic Recognition of Activities and Contexts," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 25, no. 6, pp. 1193-1206, 2017.

[12] A. Thiebault, I. Charrier, P. Pistorius and T. Aubin, "At sea vocal repertoire of a foraging seabird," *Journal of Avian Biology,* vol. 50, no. 5, 2019.

[13] N. N. Deniz, J. O. Chelotti, J. R. Galli, A. M. Planisich, M. J. Larripa, H. L. Rufiner and L. L. Giovanini, "Embedded system for real-time monitoring of foraging behavior of grazing cattle using acoustic signals," *Computers and Electronics in Agriculture,* vol. 138, pp. 167-174, 2017.

[14] A. P. Hill, P. Prince, J. L. Snaddon, C. P. Doncaster and A. Rogers, "AudioMoth: A low-cost acoustic device for monitoring biodiversity and the environment," *HardwareX,* vol. 6, 2019.

[15] OpenAcousticDevices, "OpenAcousticDevices," 2023. [Online]. Available: https://github.com/OpenAcousticDevices. [Accessed 12 4 2023].

[16] D. F. Caccamise and R. S. Hedin, "An aerodynamic basis for selecting transmitter loads in birds," *Wilson Bulletin,* vol. 97, no. 3, pp. 306-318, 1985.

[17] Adafruit, "Lithium Ion Polymer Battery - 3.7v 100mAh : ID 1570 : $5.95 : Adafruit Industries, Unique & fun DIY electronics and kits," 2023. [Online]. Available: https://www.adafruit.com/product/1570. [Accessed 12 4 2023].

Appendix A: Schematics

Figures A.1-A.8 are the schematics for version 0.4 of the Bird Backpack. The schematics are designed in Fusion 360.

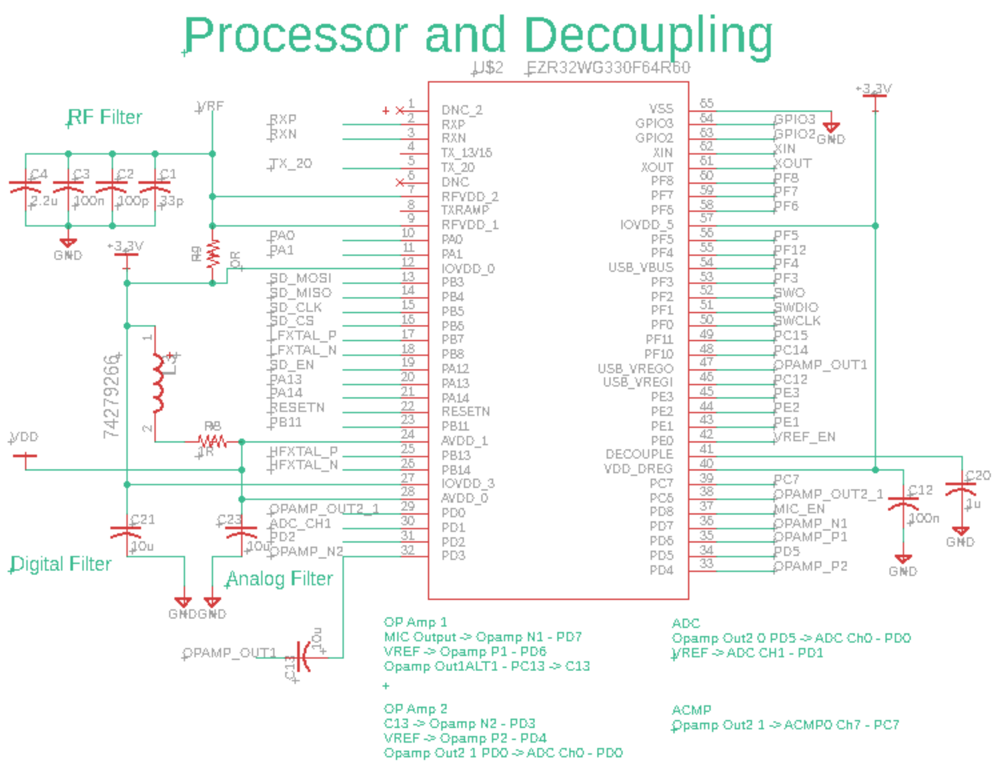

Figure A.1: Schematic of EZR32WG and Decoupling and Filtering Capacitors.

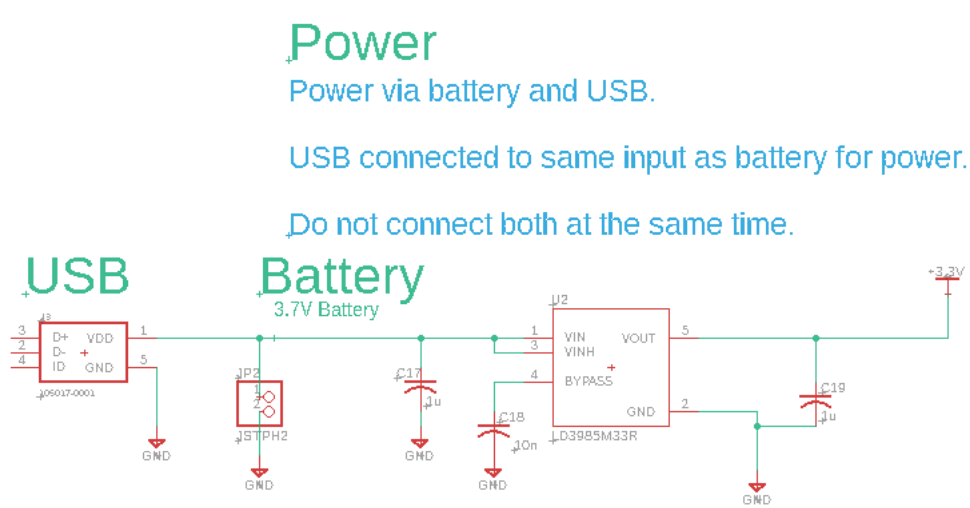

Figure A.2: Schematic of Linear Regulator, Battery, and USB Connector.

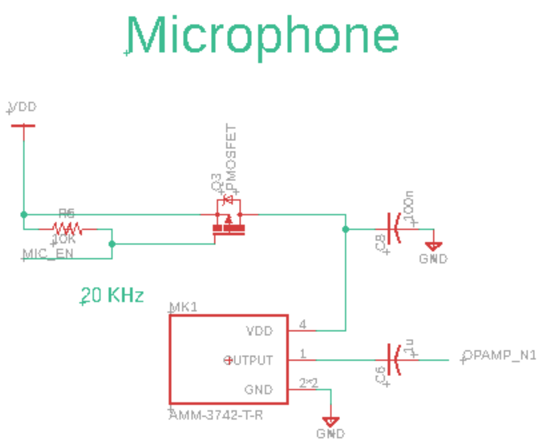

Figure A.3: Schematic of Microphone Controlled by P-Channel MOSFET.

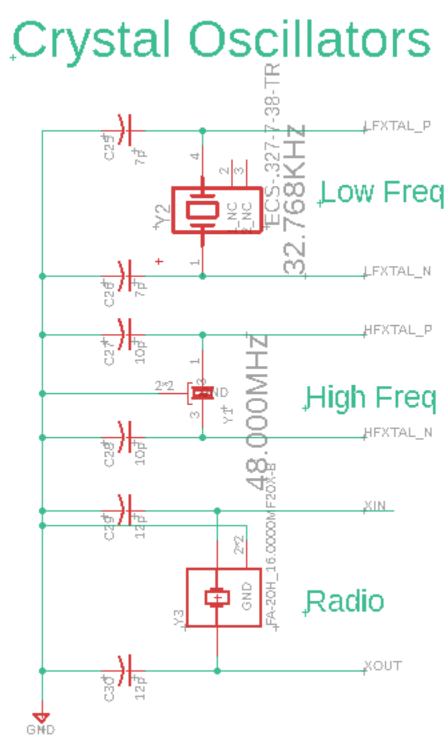

Figure A.4: Schematic of Crystal Oscillators for the low Frequency, High Frequency, and Radio Clocks.

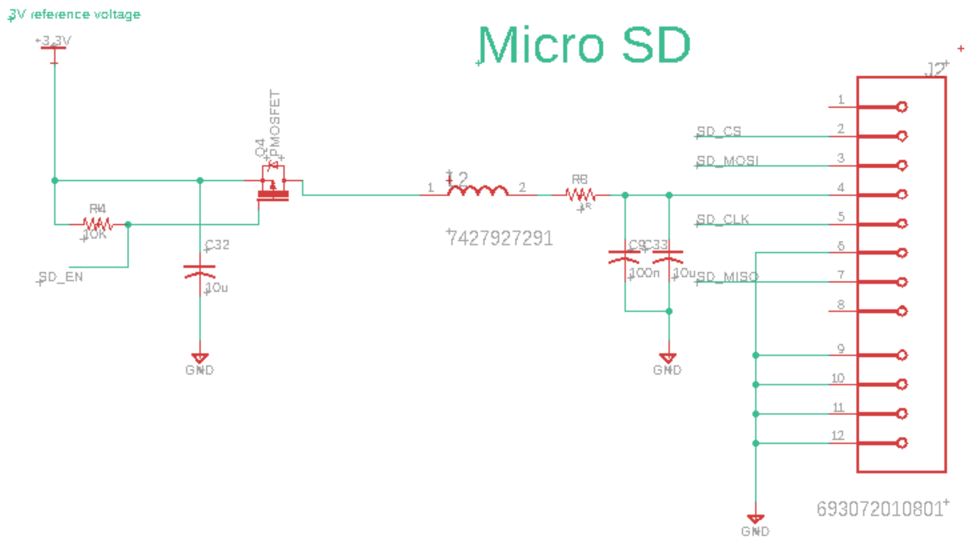

Figure A.5: Schematic for the MicroSD Card Controlled by P-Channel MOSFET.

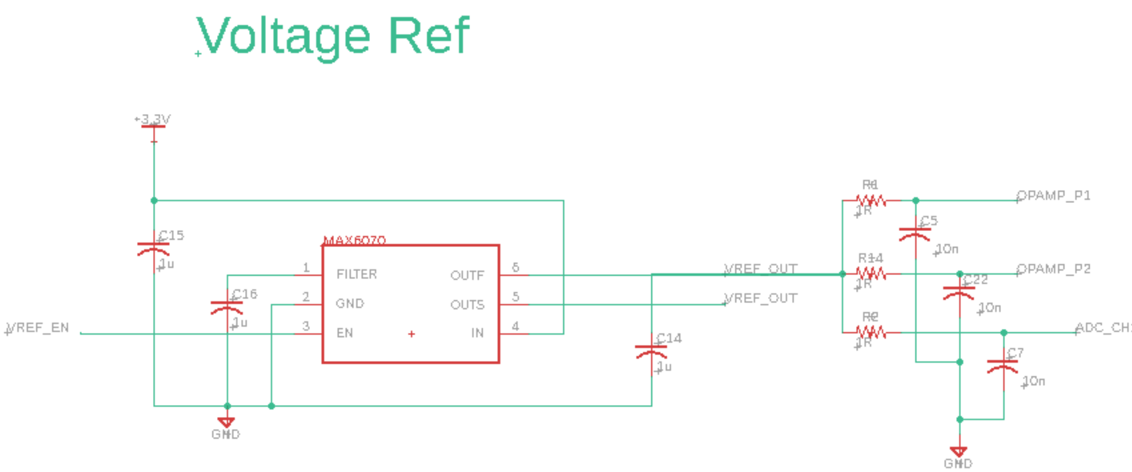

Figure A.6: Schematic for the Voltage Reference Used by the Operational Amplifiers and ADC.

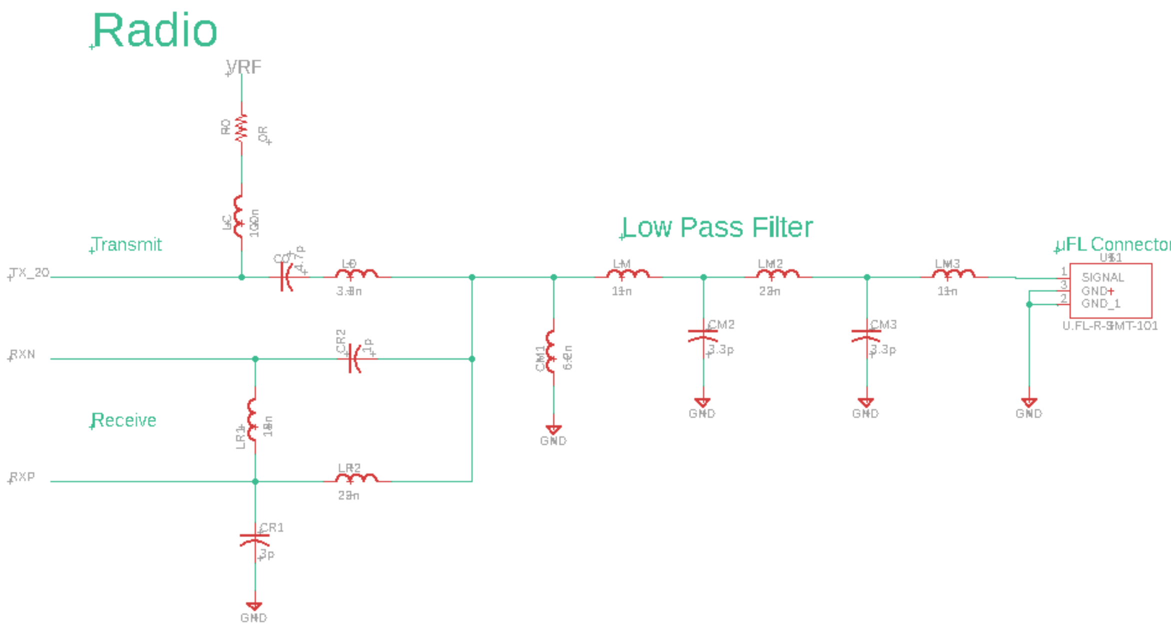

Figure A.7: Schematic for the +20 dBm Radio.

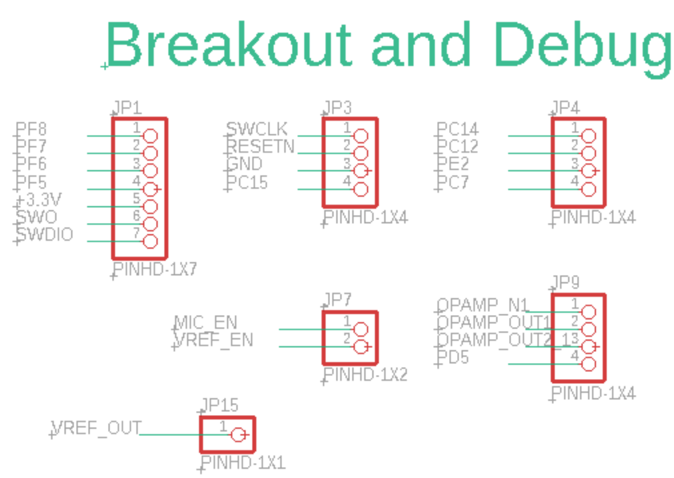

Figure A.8: Schematic of Pins Brought out to Headers for Testing and Debugging.

Appendix B: PCB Layout

Figures B.1-B.4 in this appendix show the board layout for each layer of the PCB. The bottom layer is mirrored.

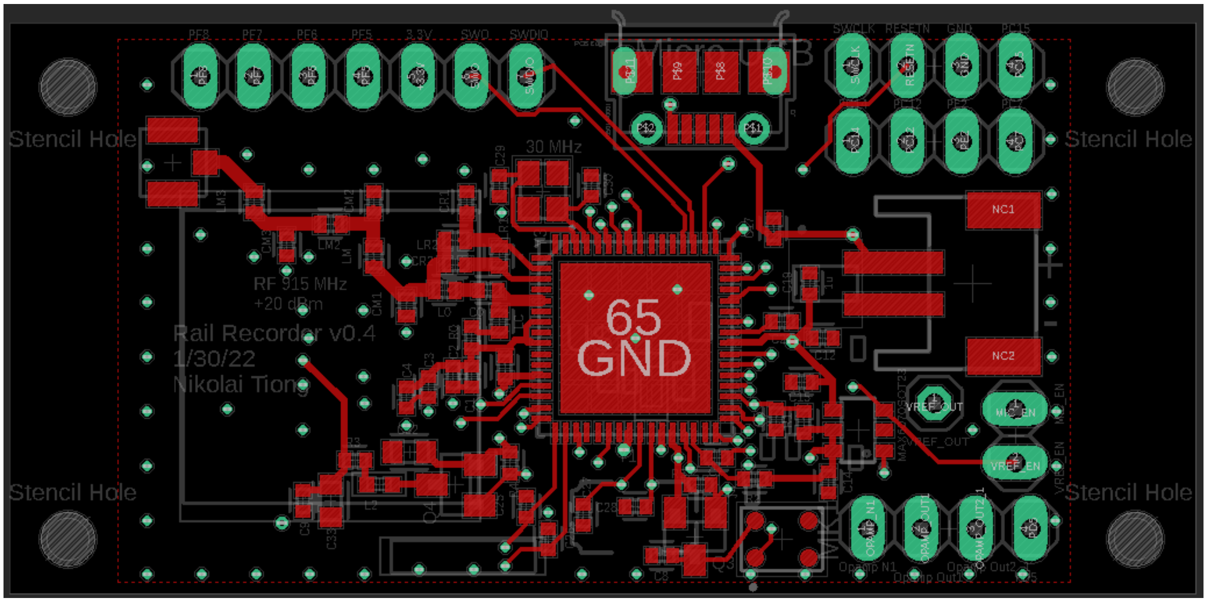

Figure B.1: Top Layer (1) PCB Layout.

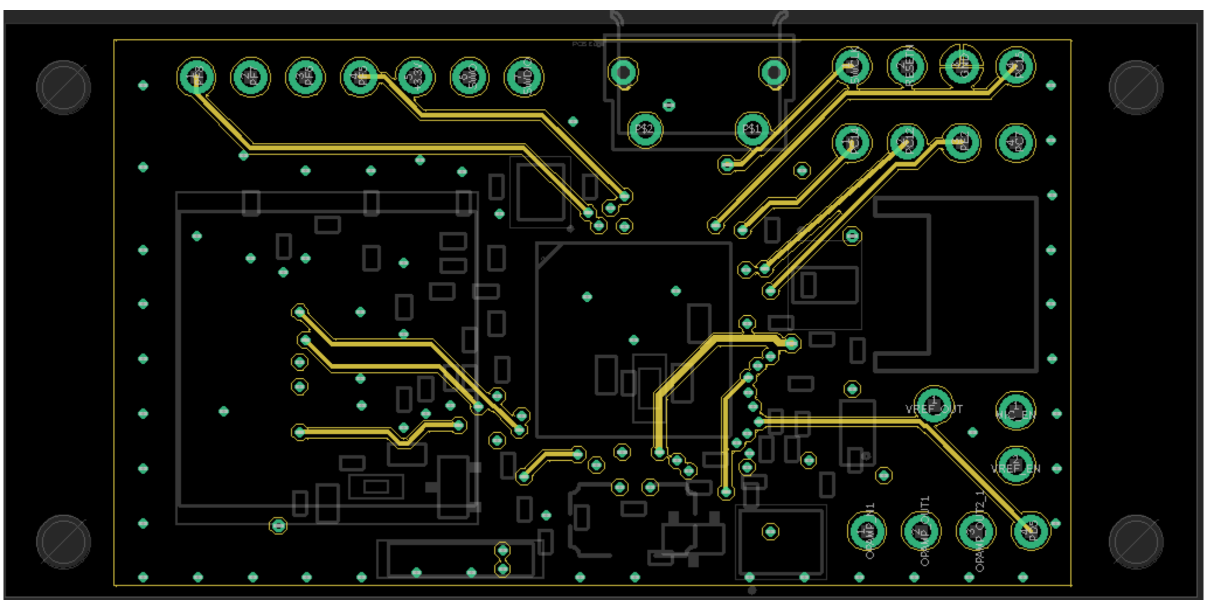

Figure B.2: Layer 2 PCB Layout.

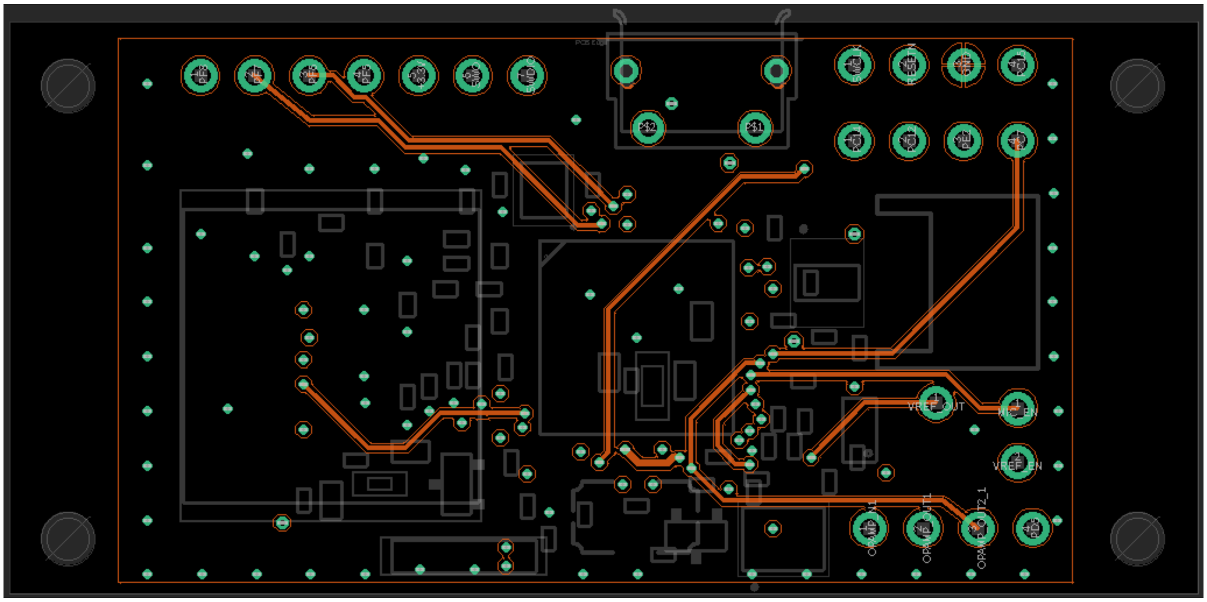

Figure B.3: Layer 15 PCB Layout.

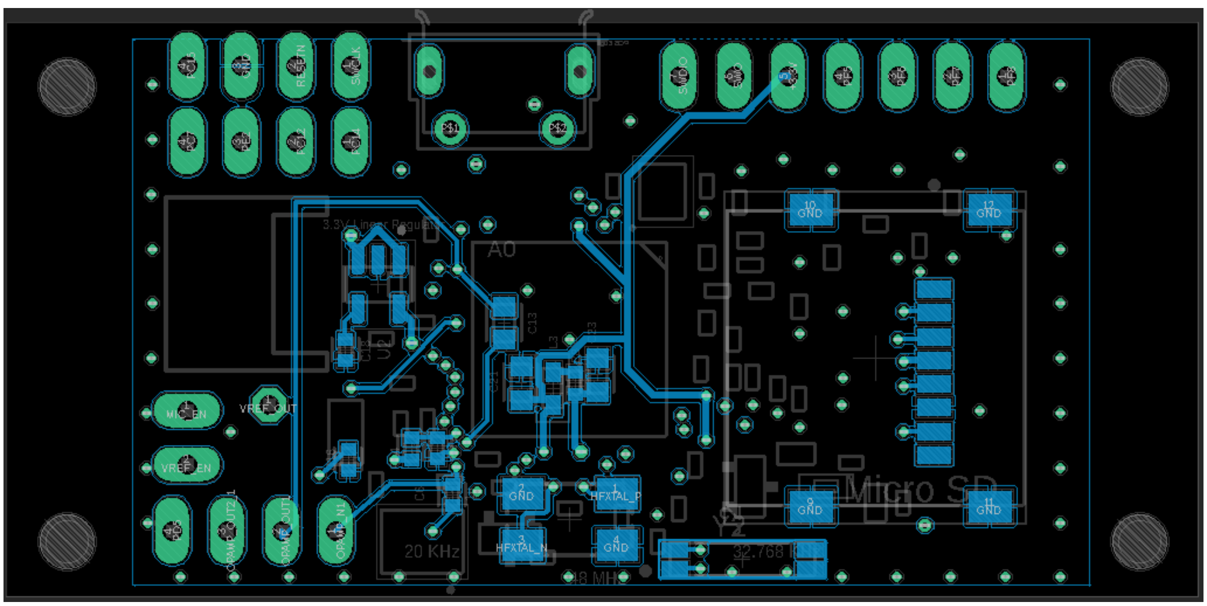

Figure B.4: Bottom Layer (16) PCB Layout.

Appendix C: Component List and Prices

Table C.1 lists all the components used in version 0.4 of the Bird Backpack. Part Ref is from Fusion 360 and there are gaps in the numbers due to components being added and removed between previous and newer versions of the board. Some component values do not line up with the description such as capacitors. This is due to not being able to obtain the value listed in the description and using the closest available one instead. Note that CM1 is an inductor in this list: different power levels and frequencies have a capacitor in this place.

Prices are based on varying quantities purchased at the time e.g., some capacitors and resistors were purchased in bulk and have a lower cost per unit than some purchased in smaller quantities. Additionally, component prices have consistently gone up as the project went on, thus, the prices listed here may not reflect current prices. Vendors used include Digikey, Mouser, Interstate Connecting Components, JLCPCB, Newark, Adafruit and OSH Stencils. The total cost per unit with the below values is $38.40, not including the cost of previous versions or the development kit.

Part No	**Part Ref**	**Value**	**Size**	**Description**	**Qty**	**Cost ($)**
885 012 005 034	CR2	1 pF	0402	1pF +/- 0.5pF, 25V ceramic capacitor	1	0.03
04025A3R0BAT2A	CR1	3 pF	0402	3pF +/- 0.1pF, 50V ceramic capacitor	1	0.03
04025A3R6BAT2A	CM2, CM3	3.3 pF	0402	3.6pF +/- 0.1pF, 50V ceramic capacitor	2	0.05
GJM1555C1H5R1BB01D	C0	4.7 pF	0402	5.1pF +/- 0.1pF, 50V ceramic capacitor	1	0.06
885 012 005 039	C25, C26	7 pF	0402	6.8pF +/- 0.5 pF, 25V ceramic capacitor	2	0.03
885 012 005 040	C27, C28	10 pF	0402	10pF +/- 5%, 25V ceramic capacitor - KIT	2	0.03
AC0402FRNPO9BN120	C29, C30	12 pF	0402	12pF +/- 1%, 50V ceramic capacitor	2	0.06

Table C.1: Component List for Bird Backpack.

Part No	Part Ref	Value	Size	Description	Qty	Cost ($)
885 012 005 043	C1	33 pF	0402	33pF +/- 5%, 25V ceramic capacitor	1	0.03
885 012 005 046	C2	100 pF	0402	100pF +/- 5%, 25V ceramic capacitor - KIT	1	0.03
885 012 205 050	C5, C7, C18, C22	10 nF	0402	10nF +/- 10%, 25V ceramic capacitor - KIT	4	0.03
885 012 105 018	C3, C8, C9, C12	100 nF	0402	100nF +/- 20%, 16V ceramic capacitor - KIT	4	0.03
885 012 105 012	C6, C14, C15, C16, C17, C19, C20	1 uF	0402	1uF +/- 20%, 10V ceramic capacitor - KIT	7	0.03
885 012 105 013	C4	2.2 uF	0402	2.2uF +/- 20%, 10V ceramic capacitor - KIT	1	0.03
885 012 106 006	C13, C21, C23, C32, C33	10 uF	0603	10uF +/- 20%, 6.3V ceramic capacitor	5	0.19
RMCF0402ZT0R00	R0, R9	0 R	0402	0 Ohm jumper 1/16 W	2	0.004
RMCF0402JT1R00	R1, R2, R3, R8, R14	1 R	0402	1 Ohm 5% resistor 1/16W	5	0.006
RMCF0402FT10K0	R4, R5	10 K	0402	10K Ohm 1% resistor 1/16W	2	0.006
LQW15AN3N3B80D	L0	3.3 nH	0402	3.3nH Unshielded wirewound fixed inductor 2A	1	0.23
LQW15AN6N2B00D	CM1	6.2 nH	0402	6.2nH Unshielded wirewound fixed inductor 700mA	1	0.16
LQW15AN11NJ00D	LM, LM3	11 nH	0402	11nH Unshielded wirewound fixed inductor 500 mA	2	0.14
LQW15AN18NG00D	LR1	18 nH	0402	18nH Unshielded wirewound fixed inductor 270mA	1	0.16

Table C.1: Component List for Bird Backpack, Continued.

Part No	Part Ref	Value	Size	Description	Qty	Cost ($)
LQW15AN22NG80D	LR2, LM2	22 nH	0402	22nH Unshielded wirewound fixed inductor 780mA	2	0.23
LQW15ANR10J00D	LC	100 nH	0402	100nH Unshielded wirewound fixed inductor 120mA	1	0.14
7427927291	L2	600 R	0402	Ferrite bead 600 Ohm	1	0.18
74279266	L3	1K	0603	Ferrite bead 1K Ohm	1	0.18
ABM3B-48.000MHZ-10-1-U-T	Y1	48 MHz		48 MHz +/- 10ppm Crystal, 10pF, HF Crystal	1	0.88
ECS-.327-7-38	Y2	32.768 KHz		32.768KHz +/- 20ppm Crystal, 7pF, LF Crystal	1	0.78
FA-20H 30.0000MF20X-W0	Y3	30 MHz		30MHz +/- 10ppm Crystal, 12pF, Radio Crystal	1	0.60
693072010801	J2			MicroSD Card Holder Hinged Type	1	3.24
105017-0001	J3			Conn Rcpt USB 2.0 Micro B	1	0.79
CONUFL001-SMD-T	U$1			uFL Connector Jack	1	0.58
EZR32WG230F256R63G	U$2		QFN-64	Microcontroller	1	13.95
LD3985M33R	U2	3.3 V	SOT23-5	3.3V Linear Regulator, 150mA	1	0.84
MAX6070AAUT18+T	MAX6070	1.8 V	SOT23-6	1.8V Voltage reference	1	2.60
DMP2160U-7	Q3, Q4		SOT23-3	PMOS 20V, 3.2A	1	0.37
AMM-3742-T-R	MK1			20 KHz Analog Omni Mems Microphone	1	1.38
JST-PH2	JP2			JST Connector for battery	1	0.40
	PCB		Varies	The PCB, min qty 5	1	1.40

Table C.1: Component List for Bird Backpack, Continued.

Table C.2 shows the supporting components that are not directly part of the Bird Backpack. Some components were already available for use and were not purchased.

Part No	Description	Qty	Cost ($)
	Stencil	1	6.80
336-3153-ND	Dev Kit Wireless EZR32WG 915MHz	1	299.00
SAM16055-ND	.050 Socket Discrete Cable Assembly, cable used for debugger	1	8.27
EZR32WG230F256R69G-C0	+20dBm microcontroller	1	6.69
EZR32WG330F64R60G-B0	+13dBm microcontroller	1	4.78
KXOB25-02X8F-TB	MONOCRYS SOLAR CELL 26.3MW 5.53V	1	3.08
KXOB25-01X8F-TB	MONOCRYS SOLAR CELL 24.5MW 5.53V	1	3.08
BQ24074	Adafruit Universal USB / DC / Solar Lithium Ion/Polymer charger	1	14.95
	+10dBi Yagi Antenna 433 MHz	1	Unknown
	Pigtail uFL to SMA connector	1	Unknown
	+2dBi Omnidirectional Antenna, 868/900/915 MHz, 4.8 cm	1	Unknown

Table C.2: Component List of Supporting Hardware.

Appendix D: EZR32WG Pinouts

Table D.1 lists the pinout of the EZR32WG230 microcontroller used on the Bird Backpack. Unused pins can be used as GPIO and have other possible functionality listed.

Pin	Name	Purpose If In Use	Other Purposes
0	VSS	Ground	
1	DNC	No Connect	
2	RXP	Radio Receive Pin P	
3	RXN	Radio Receive Pin N	
4	TX_13/16	No Connect	
5	TX_20	Radio Transmit 20 dBm	
6	DNC	No Connect	
7	RFVDD_2	Radio Voltage Supply	
8	TXRAMP	No Connect	
9	RFVDD_1	Radio Voltage Supply	
10	PA0		Timer, I2C, Radio GPIO
11	PA1		Timer, I2C, Radio GPIO
12	IOVDD0	Digital IO Power Supply 0	
13	PB3	SD MOSI	
14	PB4	SD MISO	
15	PB5	SD CLK	
16	PB6	SD CS	
17	PB7	32 KHz Crystal P	
18	PB8	32 KHz Crystal N	
19	PA12	SD_EN	
20	PA13		Timer
21	PA14		Timer
22	RESETn	Reset Pin	
23	PB11		Timer, Low Energy Timer
24	AVDD_1	Analog Power Supply 1	
25	PB13	48 MHz Crystal P	
26	PB14	48 MHz Crystal N	
27	IOVDD_3	Digital IO Power Supply 3	
28	AVDD_0	Analog Power Supply 0	
29	PD0	Op Amp 2 Output/ADC Input	
30	PD1	ADC Voltage Reference	
31	PD2		ADC, Timer, Debug Output, USART
32	PD3	Op Amp 2 Inverting Input	
33	PD4	Op Amp 2 Non-Inverting Input	

Table D.1: Pinouts of EZR32WG230 Used in Bird Backpack.

Pin	Name	Purpose If In Use	Other Purposes
34	PD5		ADC, Op Amp 2 Output, Low Energy UART
35	PD6	Op Amp 1 Non-Inverting Input	
36	PD7	Op Amp 1 Inverting Input	
37	PD8	Microphone Enable	
38	PC6	Analog Comparator	
39	PC7		ACMP Input, I2C, Low Energy UART, Low Energy Sensor
40	VDD_DREG	Power Supply Input	
41	Decouple	Power Supply Decoupling	
42	PE0	Voltage Reference Enable	
43	PE1		Timer, UART, I2C, Pulse Counter
44	PE2		Backup Domain Input, ACMP Output, Timer, UART
45	PE3		Backup Domain Status, UART, ACMP Output
46	PC12		ACMP Input, DAC Output, Op Amp 1 Output, UART
47	PC13	Op Amp 1 Output	
48	PC14		ACMP Input, DAC Output, Op Amp 1 Output, UART, Timer
49	PC15		ACMP Input, DAC Output, Op Amp 1 Output, UART. Timer
50	PF0	Debug Serial Clock	
51	PF1	Debug Serial Data I/O	
52	PF2	Debug Serial Viewer Output	
53	PF3		Timer, PRS
54	PF4		Timer, PRS
55	PF12		No extra functions
56	PF5		Timer, PRX
57	IOVDD_5	Digital IO Power Supply 5	
58	PF6		Timer
59	PF7		Timer
60	PF8		Timer
61	XOUT	30 MHz Crystal Out	
62	XIN	30 MHz Crystal In	
63	GPIO2		Radio GPIO
64	GPIO3		Radio GPIO

Table D.1: Pinouts of EZR32WG230 Used in Bird Backpack, Continued.