DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Preliminary Design: Acoustic and Environmental Monitoring System

Revisions

Revision	Author	Changes	Date
001	Haoyang Cheng Jehanzeb Mirza Junri Zhu	Initial Release	2022-02-11
002	Haoyang Cheng Jehanzeb Mirza Junri Zhu	Made Revisions Suggested in Review Preliminary Design Updated Added Chapter 10 Alpha System Design Added Chapter 11 System Budget	2022-03-18

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Table of Contents

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Acronyms

Acronym	Full Description
AEMS	Acoustic and Environmental Monitoring System
AEMI	Acoustic and Environmental Monitoring Instrument
PC	Personal Computer
FRs	Functional Requirements
PRs	Performance Requirements
SRs	System Requirements
BLE	Bluetooth Low Energy
GUI	Graphical User Interface
PCB	Printed Circuit Board
NRE	Non-Recoverable Engineering costs

References

[1] S. Knudsen, Lecture Help Session, University of Alberta, Feb. 8, 2022.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

1 Purpose

This document describes the preliminary design for the Acoustic and Environmental Monitoring System (AEMS) for lab animals.

2 Concept of Operation

The high-level operation of the AEMS is illustrated by the following user stories and use cases.

2.1 User Stories

■ User Story 1

The AEMS was set up inside of a lab rat cage and set to monitor the environmental temperature and humidity conditions over the course of a day. We started continuous logging measuring early in the morning and left it to record a sample once every 30 seconds. We came back the following day and downloaded the data off the system to analyze how the temperature and humidity in the lab had dropped overnight.

■ User Story 2

An experiment was being conducted where we were having lab rats complete physical tasks. We were using an external robotic system which monitored when the lab rats had completed certain actions and recorded their motions using a camera system. Simultaneously we were recording ultrasonic audio using the AEMS. The robotic system electronically triggered the AEMS on task completions to save all acoustic data from the last 4 minutes and the following 1 minute. Once the experiment was over we correlated the ultrasonic audio and video data to study 100 kHz vocalizations.

■ User Story 3

We had set up the AEMS to log temperature and humidity over a day. The system was also configured to automatically save audio data if the sound level had reached a specific threshold. The humidity was very high at night due to rain and the lab rats were very active. Every time the lab rats made loud squeaks we saved the last 30 seconds and the following 20 seconds of audio data. In the morning we came back and performed research trying to understand the times and humidities at which the rats were most active.

■ User Story 4

We performed many overnight recordings for a few days using the AEMS. We noticed that our data was very noisy and that many of the collected audio samples were long and contained substantial silent sections. Once we analyzed our recordings we had a better understanding of thresholding parameters and what kind of vocalizations we specifically wished to study. So we set the system to a threshold with high sensitivity on sounds between 30 kHz and 75 kHz and much less sensitivity in the audible spectrum which had a lot of background noise.

■ User Story 5

Using a later developed iteration of the AEMS we used the system to monitor the CO2 concentrations, ambient light levels, and vibrations in the lab rat cages to perform further research. We noticed vocalizations visibly using LEDs and this helped us better understand the behavior of the rats. We were performing an experiment in a sealed cage so we charged the battery of the system before use and interfaced with it wirelessly as we couldn't feed a wire to it.

2.2 Use Cases

We present the following use case diagram for the AEMS in Figure 1 to depict system use cases. Additionally, each use case is briefly described below in Table 1. The AEMS is partitioned on the diagram into two subsystems to separate the user control from the hardware. The physical device which contains the various sensors is labeled as the Acoustic and Environmental Monitoring Instrument (AEMI) while the user control interface is labeled as the Controller App.

Three actors are present in the use case diagram, the researcher, who is the principal user of the AEMS, is the sole actor for the Controller app and is largely responsible for system management while simultaneously has little direct involvement with the AEMI. A second actor is the external electronic control system, which represents an arbitrary system developed to interact with the AEMI using automation. Lastly the third actor in the system are the lab rats, secondary actors who's vocalizations trigger and are recorded by the system.

Table 1. System Use Cases

Use Case	Description
Controller App Use Cases	
Connect to AEMI With BlueTooth	Establishing a connection with the AEMI, allows basic recognition of the system being on and ready.
Start Autonomous System Logging	Signal the start of an experiment / logging session using requested parameters.
Stop Autonomous System Logging	Signal the end of an experiment / logging session. Allowing different experiments to be kept track off.
Configure Audio Window Size	Allow configuration of how many seconds of audio are saved before and after a manual or autonomous triggering.

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Configure Audio Automatic Saving Thresholds	Allow configuration of a threshold such that the system saves audio data automatically when the threshold for sound level is met. May involve a mathematical model for sound level.
Save Current Audio Buffer	A manual trigger performed by a researcher when doing an experiment. The system may not always be used autonomously.
Check Current Conditions	View the current acoustic and environmental information being collected by the sensors without downloading the logs files. Allows the system to serve as a basic measurement instrument for temperature, humidity, and ultrasonic sound.
AEMI Use Cases	
Remove Storage Media with Data	A method to access the possible large amounts of data generated by an experimental session. Standard media formats are required (e.g. SD card or USB thumb drive).
Save Current Audio Buffer	An electronic interface which can be used to manually save the audio buffer should it be connected as a simple microphone part of a larger setup. Likely done using a simple GPIO pin which may be pulled down.
Trigger Audio Threshold	A saving of the audio buffer is caused by noises created by the lab rats.

There are few direct use cases for the AEMS, this is mainly as the system is purposely intended to be a largely autonomous logging system and demand minimal researcher attention beyond setup. As most of the work is performed by the system itself, use cases are limited [1].

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

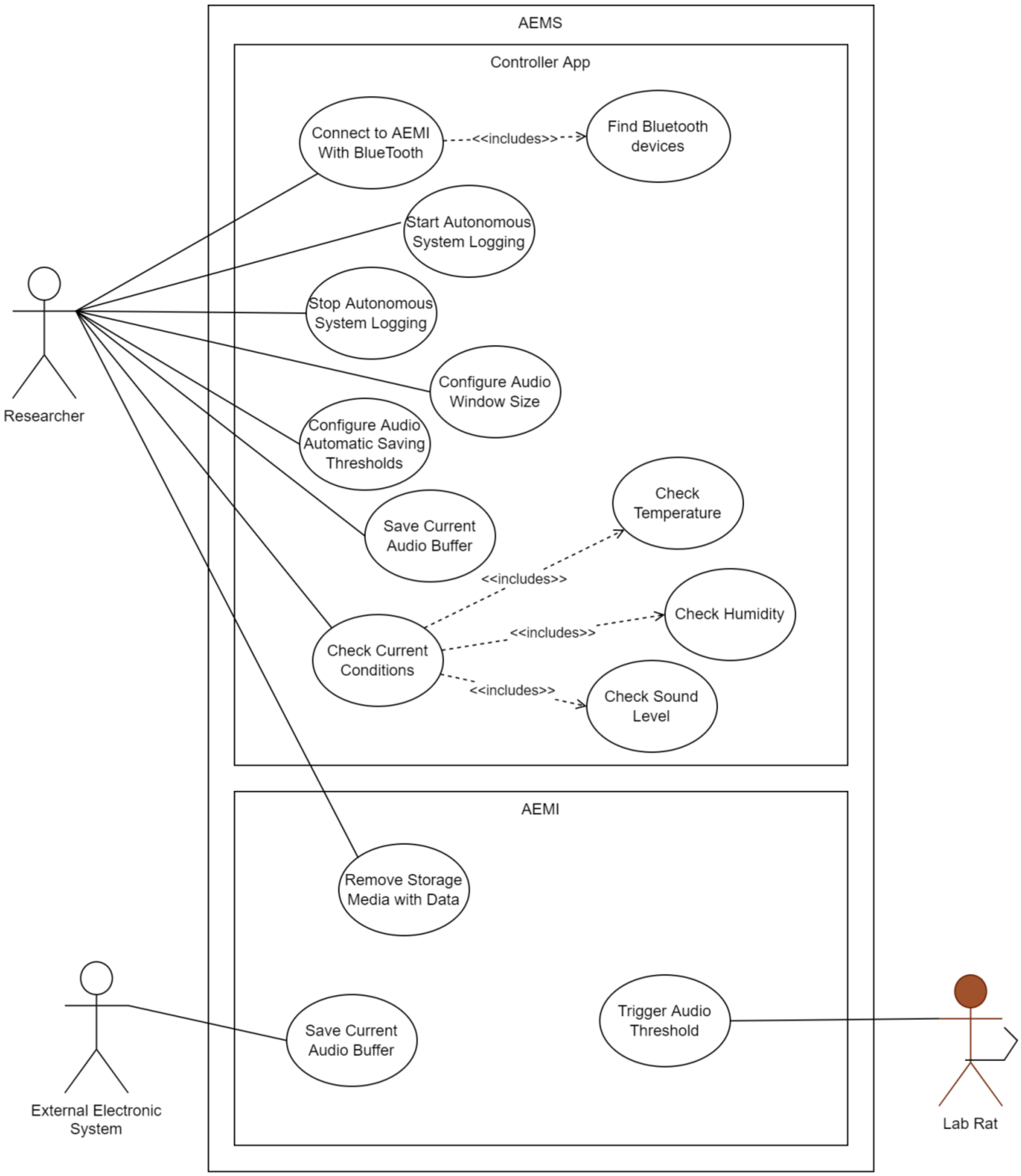

Figure 1. System Use Case Diagram

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

3 Functional and Performance Requirements

Functional requirements (FRs) are presented below in Table 1 and are categorized into two groups. FRs numbered < 100 are primary requirements to be present in the preliminary design and discussed in detail. Ones numbered ≥ 100 are extended FRs which aren't prioritized for the initial design, however should still be taken into consideration to allow easy implementation later. As a higher-order consideration, extensibility itself is a functional requirement of the device.

Table 1 Functional Requirements

FR #	Functional Requirement Description and Rationale
FR-00	Record ultrasonic sound from lab animals. The primary use for this system is to research the association between animal ultrasonic vocalizations and animal behavior and environmental conditions.
FR-01	Measure temperature of the animals' living area. The animals are sensitive to temperature. Making it important to monitor and control temperature.
FR-02	Measure humidity of the animals' living area. The animals are sensitive to humidity as it can impact their skin. Making it important to monitor and control humidity.
FR-03	Log all environmental and audio data in a standard format in system storage for later retrieval. Autonomous logging is the primary high level design goal of the system.
FR-04	A method to download the recorded data off the system for analysis.
FR-05	The user should be able to manually save an audio recording.
FR-06	The user should be able to start and stop automatic data logging.
FR-07	A method to configure the system acquisition parameters. Specific acquisition parameters of consideration are data sampling rates, automatic sound triggering thresholds, and audio window sizes for saving samples.
FR-100	Record vibration data. This serves as a source for automated triggering and animal activity monitoring.
FR-101	Record atmospheric CO2 concentrations.
FR-102	Record ambient light levels.
FR-103	Visual System Feedback using LEDs.
FR-104	System is entirely wirelessly configurable. This allows it to be used flexibly in sealed environments.
FR-105	System is battery powered and requires no external wiring.
FR-106	Data can be downloaded wirelessly without removing a storage device.

Performance Requirements (PRs) in Table 2 below follow a similar numbering scheme as functional requirements above.

Table 2 Performance Requirements

PR #	Performance Requirement Description	Related FRs
PR-00	Record sound frequencies ranging from 20-100k Hz.	FR-00
PR-01	Measure ambient temperature to at least 0.5℃ accuracy.	FR-01
PR-02	Measure ambient humidity to at least 5% accuracy.	FR-02
PR-03	The system should have sufficient storage space for data. For low bitrate data, such as periodic temperature measurements, the system is able to store > 30,000 samples. For ultrasonic sound, the system should store at least 2 hours of audio recordings.	FR-03 & FR-04
PR-04	The saved audio data window shall have an M-second pre trigger and N-second post trigger section. Where the sum of N+M, the total window size, is at least 300 s (5 min).	FR-05 & FR-07
PR-05	System shall be compact enough to fit comfortably in a lab rat cage. No linear dimension shall be larger than 16 cm. Volume shall be less than 1000 cm^3.	
PR-100	Battery should last for 24 hour between charges.	FR-105
PR-101	System average current consumption shall be less than ~500 mA to allow 24 hour operation with a large commercial USB battery bank.	
PR-102	Utilizing data compression to reduce file sizes to improve transfer times and storage space.	FR-03 & FR-04

4 System Design

The following few sections describe the high level preliminary system design for the AEMS.

The AEMS comprises two major subsystems, a monitoring system built using the Raspberry Pi as a platform (AEMI) and a personal computer (PC) application for wireless configuration (Controller App). The AEMI is an autonomous logging instrument that minimally captures ultrasonic audio, ambient temperature, and ambient humidity levels and has removable storage media from which data can be collected. A bluetooth module is present on the Raspberry Pi board which is used for wireless communication with the Controller app device.

The controller app is an application for a Windows PC which uses Bluetooth to wirelessly communicate with and configure the AEMI.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

4.1 System Architecture

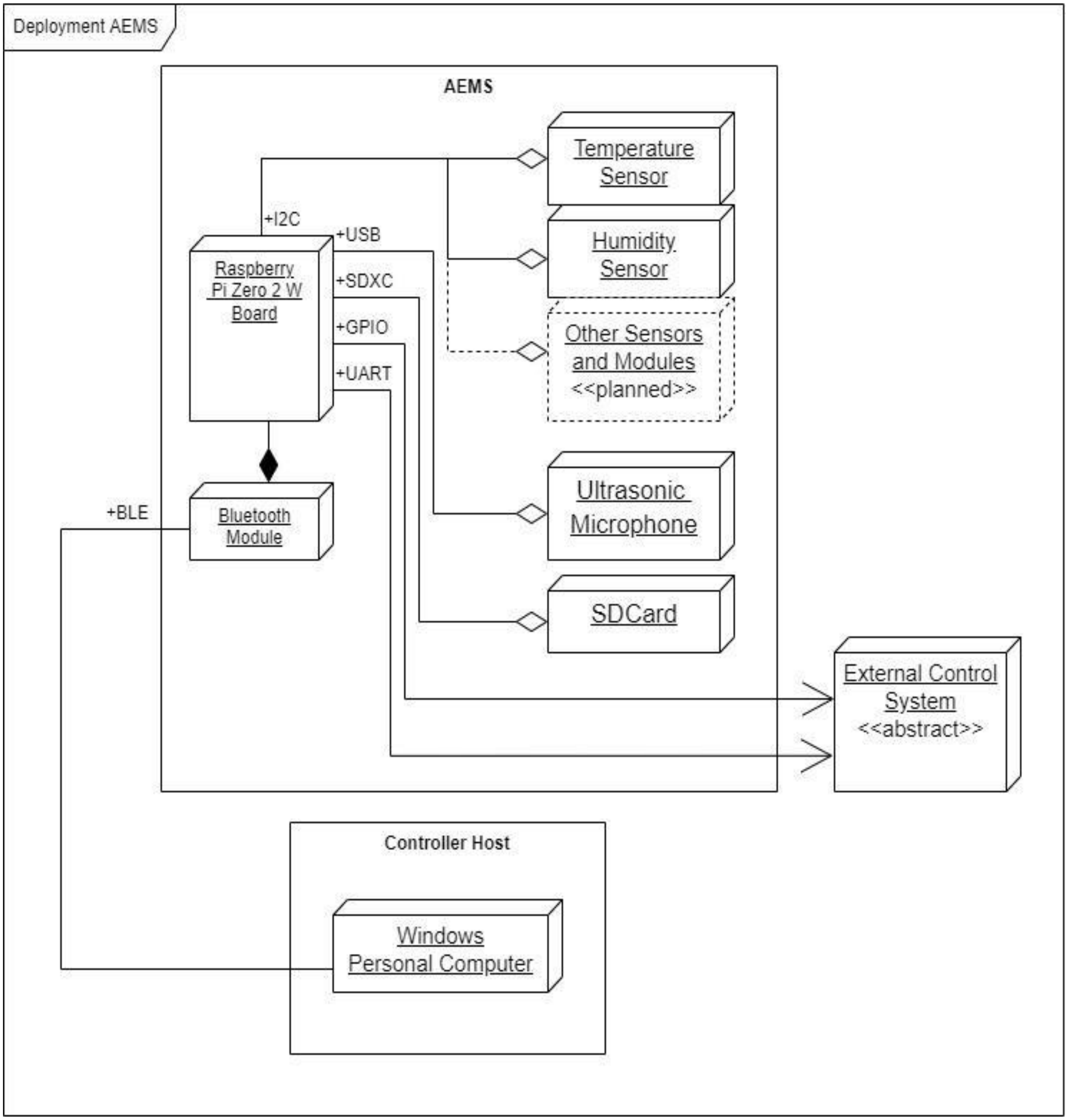

Figure 2. Deployment Diagram

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

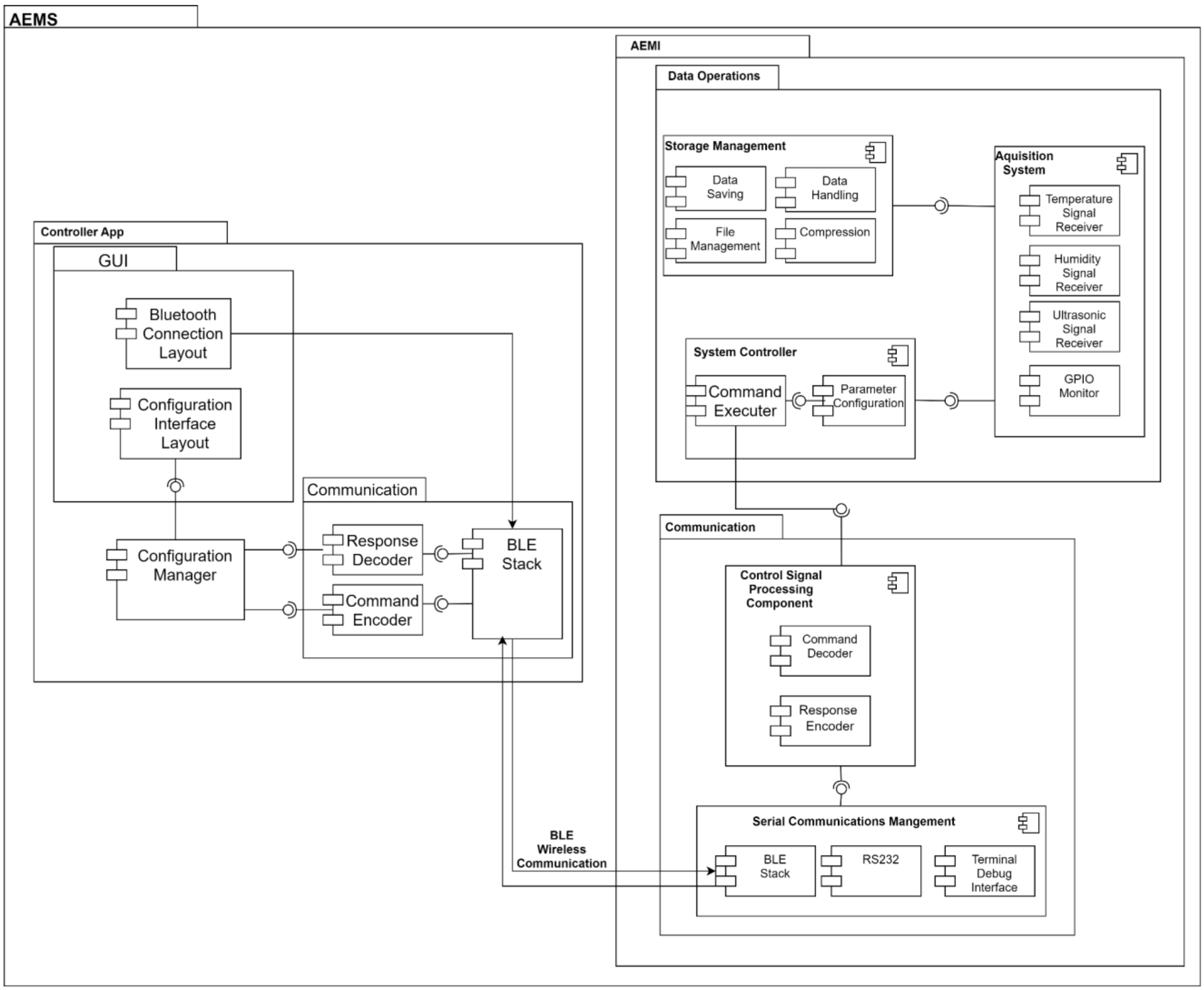

Figure 3. Component Diagram

The above figures, Figure 1 and Figure 2, depict the system architecture as a deployment diagram and component diagram respectively. The deployment diagram reflects the hardware configuration of the planned system while the Component details the overall software architecture of the Controller application and AEMI Software/Firmware.

The following subsections tabulate the system hardware, firmware, and software components with respect to the Component and Deployment Diagrams.

4.1.1 Hardware Components

Component Name	Description
Raspberry Pi Zero 2 W	A small ARM based linux capable computer which serves as the platform for the AEMI. Has sufficient RAM, compute, and IO to meet requirements.
Temperature Sensor	An electronic temperature measuring chip.
Humidity Sensor	An electronic humidity measuring chip.
Ultrasonic Microphone	A microphone capable of recording ultrasonic (~100 kHz) sounds.
SDCard	Serves as the storage space for data. Removable by the user. Also contains the Raspberry Pi's operating system and software (firmware).
Bluetooth Module	A Wifi/BlueTooth wireless module which is embedded as a part of the Raspberry Pi Board. Capable of Bluetooth Low Energy (BLE) and Bluetooth 4.2.
Controller Host (Android Phone/PC)	The platform for the controller app software. Must be BlueTooth capable. Either an Android Phone platform or Windows computer will be used.
Other Modules and Sensors (Planned)	A set of planned/requested components to be implemented after the minimal design is proven. Includes ambient light, accelerometer, and barometer. Eventually, addition of gas sensors is also planned.

4.1.2 Firmware Components

Firmware components are presented in the component diagram, Figure 3, under the AEMI system. The Raspberry Pi itself will be running Raspberry Pi OS, an operating system based on Debian/Linux. As such the code written for it may be considered software as it wouldn't be bare metal. However as the user has no access to the underlying operating system, we consider all modules on the AEMI to be firmware.

Component Name	Description
Acquisition System	The subsystem on the AEMI that handles communications with the various hardware sensors. Includes submodules for temperature, humidity, ultrasonic audio, and GPIO inputs.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

System Controller	The module which coordinates the other firmware/software components of the AEMI using a command execution protocol.
Control Signal Processing	A system which decodes and manages the execution and response of commands. Commands are the underlying low level AEMI communication protocol.
Serial Communications Management	Manages serial protocols to allow simultaneous access from multiple separate connections methods. Contains the BlueTooth stack for serial communications over BLE. Also includes RS232 for UART communications and Terminal Debug for development and testing.
Storage Management	The storage management system manages the saving of logged data. Will handle file management and temporary data handling, will also perform required functions such as format conversion and file compression.

4.1.3 Software Components

Software components are presented in the component diagram, Figure 3, under the Controller App system. As all AEMI components are considered firmware, software exclusively refers to the Controller App.

Component Name	**Description**
Controller User Interfaces / Graphical User Interface (GUI)	At minimum two GUIs are required. One to allow users to view and modify parameters of the system as described in the use cases. The other is to allow the user to set up a Bluetooth connection with the AEMI.
Configuration Management	A software subsystem of the controller app that manages the low level configuration parameters of the AEMI.
Communication	A subsystem to communicate with the AEMI via bluetooth. Includes the BLE stack, and a method to encode configuration as commands and decode responses to those commands to present.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

5 System Requirements

The following table, Table 3, highlights the major system requirements (SRs) for our perliminarydesign.

SR #	System Requirement Desc	FR#	PR#	Notes
SR-00	Raspberry Pi Zero 2 W	03,04,104	04,05,100,101	
SR-01	Ultrasonic Microphone	00	00	
SR-02	Humidity Sensor	02	02	
SR-03	Temperature Sensor	01	01	
SR-04	SDCard	04	03	
SR-05	Raspberry Pi OS	03		The operating system running on the Raspberry Pi, provides many required libraries.
SR-06	I2C Interface	100,101, 102		Required for sensors and extending hardware.
SR-07	USB Interface	00		Required for Ultrasonic Microphone
SR-08	GPIO Ports	05		Basic Electronic Control Interface
SR-09	Raspberry Pi Bluetooth Stack	05, 06		
SR-10	Controller's OS supports Bluetooth Low Energy (BLE)	04, 06	100, 101	Usually standard on Windows
SR-11	Supports Windows 10 and 11	04, 07		
SR-12	GUI library for controller platform	05, 06	04	Having avenues for cross platform support is desirable.

Table 3. System Requirements

6 Minimum Design

In this section a minimum design, the "walking skeleton" is described. The purpose is to define the functionality to be implemented in the first development iteration. The outcomes are reported to the client providing an opportunity for early feedback.

1. 6.1 Individual Module Implementation

Early on we will need to demonstrate functionality of many of our individual modules before we can demonstrate an integrated system which can provide end-to-end functionality.

- Demonstrate the proper functionality of the ultrasonic microphone by recording samples to memory buffers on the AEMI.
- Demonstrate successful I2C communication and use of the sensors by recording temperature samples to memory.
- Set up the BlueTooth stack such that the AEMI advertises itself and can be connected to.
- Create a Controller App that can search for and successfully connect to the AEMI via BlueTooth and read current sensor information.
- Create a minimal command protocol implementation to control the AEMS over Bluetooth.

2. 6.2 End to End Demonstration

After implementing and successfully demonstrating each individual module described above, the next step will be to combine those modules into a cohesive system and integrate them together to show a minimum satisfactory result.

The specific requirement is that we should be able to use the Controller app to find and connect to the AEMS, where we can read the current temperature.

7 High-level Hardware Design

Raspberry Pi Zero 2 W:

- The Raspberry Pi Board is the major processing component of the AEMS. It will connect with the Temperature & Humidity Sensor through I2C, connect with the Ultrasonic Microphone through USB port. Also, it will be wirelessly connected to the Android/PC Controller App through Bluetooth Low Energy.
- The Raspberry Pi is powered through a dedicated power microUSB port.

Temperature Sensor:

- The Temperature Sensor will be connected to the Pi Board through an I2C interface provided on the Pi Board. It is powered by the Pi Board GPIO pins.

Humidity Sensor

- The Humidity Sensor will be connected to the Pi Board through an I2C interface provided on the Pi Board. It is powered by the Pi Board GPIO pins.

Ultrasonic Microphone:

- The microphone will be connected through the USB interface of the Pi Board
- The ultrasonic microphone will be a high level digital device providing a USB audio class interface to read samples from.

SDCard:

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

- The SD card will be plugged in the SD card slot on the Pi Board.
- The SD is easily removable and plugged into a computer to download data off of.

Bluetooth Module:

- The Bluetooth Module is already embedded as part of the Pi Board. It supports Bluetooth 4.2 and BLE, of which the latter will be used.

Controller Host (Windows PC):

- It will be a Windows PC with BlueTooth functionality and can communicate with the Pi Board.

8 High-level Software/Firmware Design

Acquisition System (AEMI):

- The Acquisition system will be responsible for receiving data from sensors and sending them to the Storage Management.
- One of the main purposes of the acquisition system is to serve as a driver for our hardware components.
- The system will use an I2C library to interface with the sensors and receive environmental data.
- The Ultrasonic audio data will be received by the acquisition system, which will also handle processing and triggering detection. The system will use libraries such as ALSA and pyaudio for acquiring raw data.
- Communication through GPIO port is handled by the acquisition system
- The system will be controlled by the System Controller, including parameters changes and starting/stopping reading.

System Controller (AEMI):

- The System Controller serves as a system manager that handles execution of the decoded command from Control Signal Processing Component and performs direct execution onto the Acquisition System by modifying the specified attributes inside the system.

Control Signal Processing Component (AEMI):

- This system contains an encoder and decoder to translate the control message from the Serial Communications Management System to the Commands expected by the system controller.

Serial Communications Management (AEMI):

- Serial Communication management system receives a serial data command stream from various sources such as the BlueTooth stack, RS232, and from the terminal environment of the Raspberry Pi.
- The subsystem manages synchronization of the streams and duplication across them.

- The subsystem will make use of the Raspberry Pi BlueTooth stack, such as Bluez, for BLE communication.

Storage Management (AEMI):

- Storage management includes data formatting, file management, data handling and compression is developed using published libraries such as tinywave and scipy so it can be easy for the user to access from the SD card.

Controller User Interfaces (Controller APP):

- A Windows app is developed using available third party Python libraries that includes Tkinter for GUI and Bleak for BLE communication.

Configuration Management (Controller APP):

- Can store configuration settings, which are provided by the user and transfer new parameters once updated.

Communication (Controller APP):

- Serial connection using bluetooth is built based on published Java API for android app and python or C API for window app, it encodes the command and sends it to the Pi and receives real-time parameters back from it.

9 Prototype Budget

The bill of materials for the major components used in our preliminary design are presented below in Table 4.

Table 4 System Budget (ROM)

Component	Mfr P/N	Mfr	Qty	Unit Price	Extended Price
Raspberry Pi Zero 2 W	1799	Raspberry Pi Foundation	1	$18.95	$18.95
12.5W Power Supply	1793	Raspberry Pi Foundation	1	$10.95	$10.95
Raspberry Pi Zero Adapter Kit	K2B-1306	Raspberry Pi Fundation	1	$6.95	$6.95
SanDisk 128GB MicroSDXC (SD Card)	SDSQUA4-128G-GN6MA	SanDisk	1	$21.16	$21.16

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Sparkfun DHT20 (Temperature and Humidity Sensor)	SEN-18364	SparkFun Electronics	1	$8.76	$8.76
Dodotronic UltraMic384K EVO (Ultrasonic Microphone)	UM384EVO	Dodotronic	1	$450	$450
				Total Cost	$516.77

❖ 10 Alpha System Design

This section introduces our alpha system design. The alpha system is an iteration on our prototype preliminary design, presented in the prior sections of this document, designed specifically for low volume (< 1000) manufacturing and initial public sales. Towards this goal we make several changes to our design to optimize on several measures such as size, weight, power, and cost as well as refine usability and improve manufacturability. Our alpha design makes no changes to the high level architecture in which the AEMS consists of a hardware system with many sensors, the AEMI, and a Controller App for wireless configuration. There are however several major and minor changes made to the AEMI and the Controller app themselves discussed in detail below.

The AEMI in our alpha design is revised to both improve usability, functionality, and to better suit low volume manufacturing. As in the preliminary design, we continue to use the Raspberry Pi Zero 2 W as the basis for the AEMI system. We included several additional sensors in our alpha design to provide better coverage of our customers' needs. The alpha design in total includes an ultrasonic microphone; sensors for temperature, humidity, luminous intensity, and altitude/pressure; and an accelerometer. One no-sensor component now included is a real time clock to maintain time without an internet connection. Our alpha system design is still flexible and modular with potential for even more additional sensors. Particularly sensors for ammonia, carbon dioxide, and volatile organic compounds are planned in an even further revision.

The AEMI in the alpha design will be manufactured using a custom printed circuit board (PCB) onto which the Raspberry Pi is embedded instead of the breadboard construct used in the preliminary design. Sensors previously packaged for connecting using jumper cables would be replaced with surface mounted ICs and throughole components. This both saves on cost and size and is a requirement for manufacturing.

One major change in the user experience of the AEMI is that the responsibilities of the microSD card have been divided into two separate components. Previously we stored both the AEMI firmware and the

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

user collected data on a single SD card. This is convenient for development as we required no additional hardware and would be frequently removing the SD card anyway. However, it makes little sense to combine this responsibility in an alpha product as the ability to view and potentially corrupt firmware files when trying to download data is confusing for an end user. This also meant that the AEMI needed to be powered down when removing the data storage medium and rebooted upon re-insertion.

Instead data collected by the AEMI is now written to an external USB storage device that may be plugged into a new USB port added to a system enclosure created for the AEMI. Aside from convenience another benefit of this approach is that storage requirements may be decided by the end user themselves. Necessary support hardware for this change mainly consists of a USB Hub controller IC to extend the single USB interface of the Raspberry Pi Zero 2 W. To improve user experience further we also incorporated two status LEDs as part of the AEMI. These LED's let the user know visually at a glance the current status of the system, and may help prevent a situation where a user forgets to start system logging.

On the firmware side much of the firmware created for the preliminary design would be reusable in the alpha design with extended functionality added for new components, interfaces, and the status LEDs. This represents one major advantage of keeping with a Raspberry Pi as the basis for an alpha product as it saves greatly on non-recoverable engineering costs (NREs). Further discussion on the use of the Raspberry Pi is included in the following section on system architecture.

The Controller App in our alpha design remains similar to what was used in the preliminary design, mainly including additional GUI options associated with additional sensor configuration. However we now extend our support for several additional platforms beyond just Windows PC. Primarily we consider creating mobile applications to configure the AEMI using Android or Apple smartphones due to the ubiquity of these devices and the large number of potential users.

The AEMS in the alpha design continues to operate using an ASCII command protocol that controls the AEMI over a serial bluetooth interface. In an alpha design we now provide a small application programming interface for technical users to be able to directly send arbitrary commands and have greater control over the device for automation. We include this API as it would be desirable for customers who wish to purchase several units in bulk to monitor a large number of lab animals and locations.

➢ 10.1 System Architecture

In this section we further discuss differences in the system architecture as mentioned previously above, specifically we discuss both the motivations for making changes as well as reasons for not changing parts of the design. For this discussion we also provide Figures 4 and 5, the alpha design deployment diagram and the alpha design component diagrams below.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

One major change in our AEMI architecture is the inclusion of more sensor units. This is a natural progression as the AEMS is meant to be a combined all-in-one environmental logging system that manages to capture a diverse set of data replacing the need for multiple individual units. Our focus on design for sensor modularity means that such an addition is well poised and may be implemented rapidly without high NRE costs in firmware and software development.

The other major change in our AEMI architecture is the replacement of our storage system from using the onboard SD card to an external USB media. The motivations for this are already discussed in detail previously. This particular change was motivated by inconvenience and poor user experience discovered during early prototyping.

Notable aspects of the preliminary design that are unchanged are the use of a Raspberry Pi board and the use of our integrated USB ultrasonic microphone. Hypothetically the entire system may be converted to using a true embedded platform such as an STM32 microcontroller or other similar offerings from competitors. This approach may save substantial unit costs due to all the unused Raspberry Pi functionally and would greatly reduce system power consumption which is dominated by fairly high current Raspberry Pi. However such a change would impart possibly tens of thousands of dollars in NRE costs as much of the firmware would need to be entirely rewritten to be ported to a lower level platform compared to the linux environment of the Pi, something that may take a few months to accomplish.A second major issue would be the need to perform certifications for our product since our product has wireless capabilities. Volumes of even 1000 are optimistic for a niche sensor system so such costs could never be recovered.

Similar reasons are had for the use of the integrated USB ultrasonic microphone. The component is very expensive and makes up the majority of our product's cost. However the major benefit of an integrated solution is that it saves on the engineering costs of designing and testing an ultrasonic microphone from analogue components. Such a change could require several months of work and represents both excessive risk and NRE costs for this project due to the large number of unknowns. These uncertainties stem from the fact that commercial components that work at ultrasonic frequencies are rare and possibly very expensive specialized equipment would be needed.

Our architecture for the controller app is largely unchanged since it was already designed to be flexible and easily upgraded. Most of the software work in that regard would be porting our code to a cross platform system that works on both IOS and Android. This should be reasonable however since the Controller app is ultimately designed to be a thin veneer over the underlying serial command protocol upon which it is built. Porting would mostly be a matter of implementing a basic GUI that makes use of an already developed protocol.

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

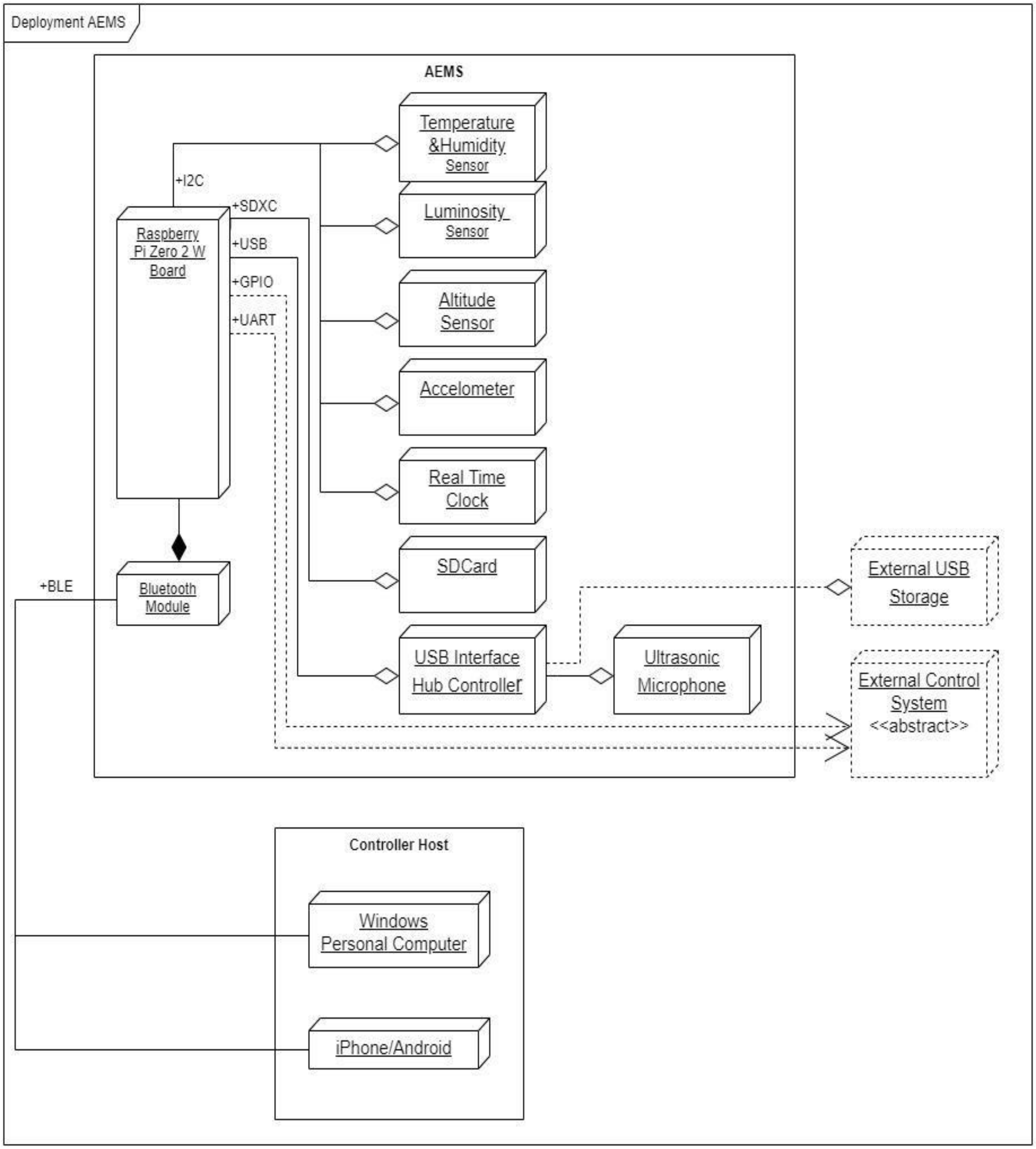

Figure 4. Alpha Design Deployment Diagram

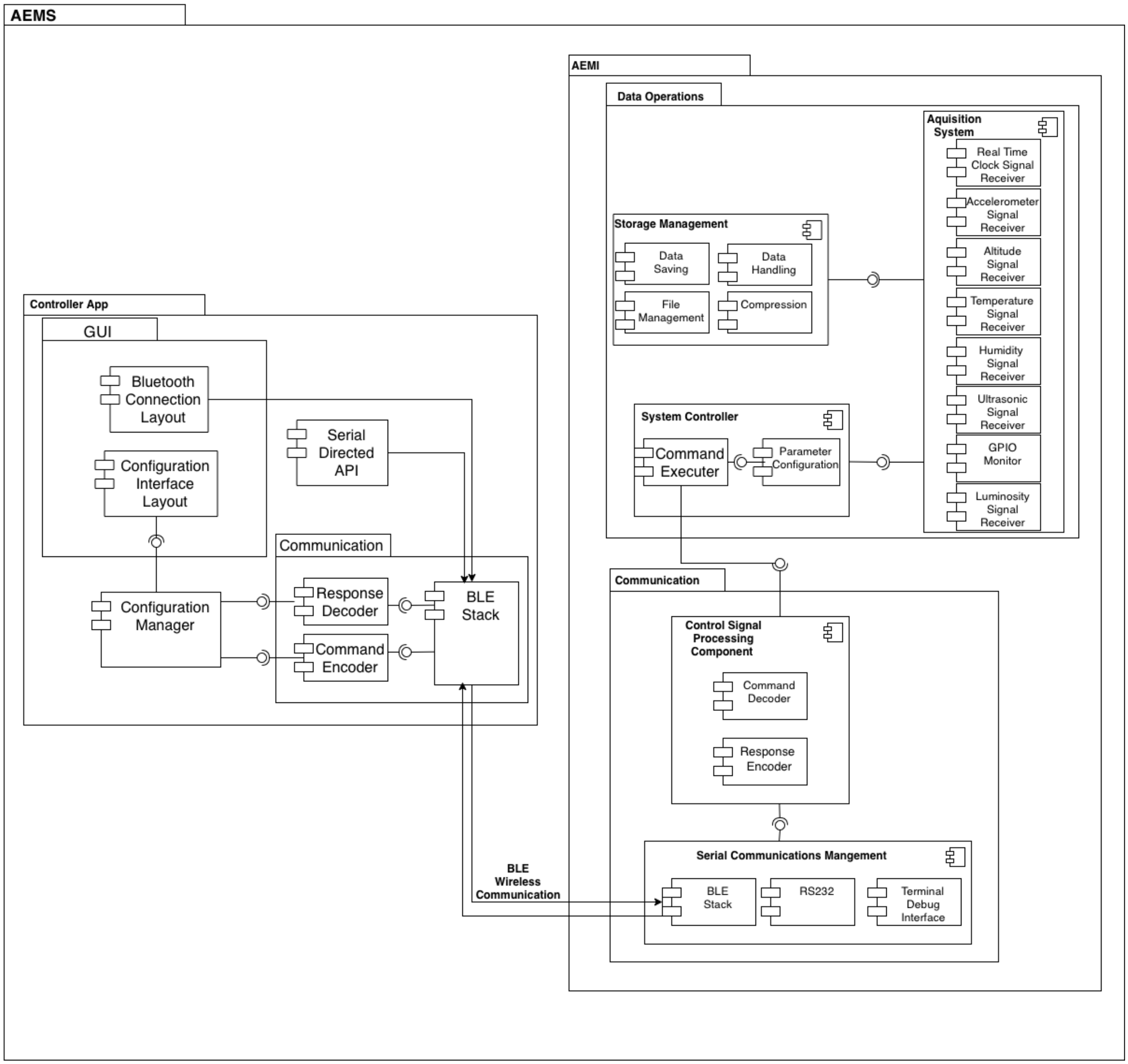

Figure 5. Alpha Design Component Diagram

➢

➢

➢

➢ 10.2 Block Diagram

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

The following figure, Figure 6, shows our alpha design system block diagram. This block diagram shows the major components present in the system alpha design including the various interconnections involved. Minor supportive components such as capacitors and LEDs have been left out. The block diagram reflects the architecture described in the previous two sections.

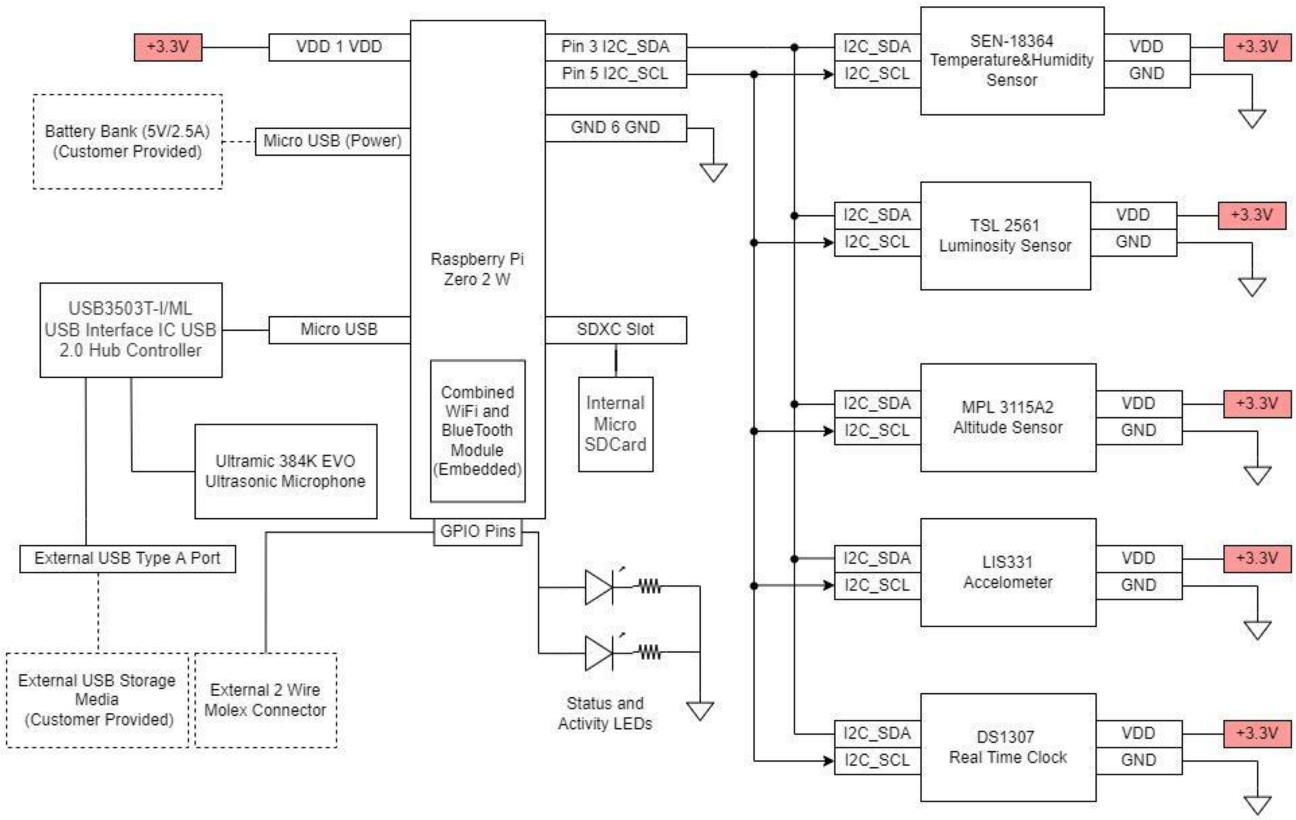

Figure 6. Alpha Design System Block Diagram

❖ 11 System Budget

The following table shows an approximate landed cost for our alpha design should we produce a volume run of 1000 components. It includes specific manufactures from which we may source our components. Some values in the table are educated guesses based on historical information and we assume that the current major chip shortage is resolved in that regard.

Table 5 System Budget (Alpha design)

Description	Mfr PN	Manufacturer	Qty	Vendor	Unit Cost ($CAD)	Ext. Cost(1000)

DEPARTMENT OF
Electrical and Computer Engineering

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

Raspberry Pi Zero 2 W	1799	Raspberry Pi Foundation	1	Digikey	$18.95	$14.00[1]
microSD card	IOMC001	NA	1	AliBaba	$10.00	$2.50
Temperature & Humidity Sensor	SEN-18364	SparkFun Electronics	1	Digikey	$8.76	$6.6
Ultrasonic Microphone	UM384K EVO	Dodotronic	1	Dodotronic	$450	$250[2]
Adafruit Luminosity Sensor	TSL2561	Adafruit Industries	1	Adafruit Industries	$5.95	$4.76
Maxim Integrated Real Time Clock	DS1390	Maxim Integrated TM	1	Digikey	$7.42	$4.20
SparkFun Accelerometer	LIS331	SparkFun Electronics	1	Digikey	$13.95	$3.17
SparkFun Altitude/Pressure Sensor	MPL3115A2	SparkFun Electronics	1	Digikey	$16.5	$9.08
USB 2.0 Hub Controller	USB3503T-I/ML	Microchip	1	Mouser	$6.04	$6.04
LED diode	CZ-F5	NA	1	AliBaba	$0.02	$0.02
PCB	NA	NA	1	NA	$2	$0.5
Yield					100%	95%
Subtotal					$539.59	$316.71
Housing		NA	NA	Container Distributor, Inc.	$50	$6
Packaging		NA	NA	TBD	$15	$4
Shipping		NA	NA	Canada Post	$50	$0.05
Returns	NA				0%	5%
Landed Cost					$654.59	$343.09

The unit costs are likely substantially unchanged for many of the components until very large volumes begin to be manufactured. Much of the high landed cost is attributed to the ultrasonic microphone and

[1] We assume a roughly 25% discount on a bulk purchase of the Pi, in line with a ~35% discount seen for bulk purchases of ARM microcontrollers units.
[2] The large discount is since much of the cost of a single unit is shipping and handling a single item from Europe.

11-203 Donadeo Innovation Centre for Engineering
9211-116 Street NW
University of Alberta
Edmonton, Alberta
Canada T6G 1H9

as there is a single low volume supplier of the item. It's possible that for several thousands of units we get an even greater discount of up to 70%+ on the component should we develop strong relations with the company manufacturing them.

The market for the product is quite limited since it is designed for researchers using lab rats specifically interested in monitoring environmental conditions and ultrasonic audio. Our preliminary design is intended to meet the needs of a single sponsor who is conducting research on the usefulness of such a system. Should a study yield that such a system produces meaningful data that assists in or necessary for biomedical research, we may still have only a few tens of researchers interested in a product. Even a volume run of a thousand as discussed above seems highly unlikely.

In order to go forwards with a commercial product suitable for volumes in the tens of thousands, it would be necessary to re-architecture the product in ways that lend itself towards that. Particularly going from an integrated ultrasonic microphone system to implementing the ultrasonic microphone using analogue components would greatly reduce unit prices. This isn't done here as the NRE costs of such an endeavor would be great in managing audio test equipment.

Another potential avenue for reducing costs in high volumes would be manufacturing the product with multiple slews. Our alpha design is just a collection of many sensors of which much of the data may not be useful to many users. Some slews may omit some of these components to save on costs.