Orientation Tracking Module For Headphones

Molly Fane, Sally Zhou, and Cary Zhu

ECE 445 Design Document

Spring 2019

Team 46

TA: Zhen Qin

Table of Contents

1. Introduction

1.1 Objective

Commercially, headphones today are able to create the illusion of sound coming from different orientations through the use of stereophonic sound. This is commonly achieved by having two different channels of audio with slightly different timing and intensity. In combination, a listener's brain will decode this information by triangulating the position of sounds [1]. However, when a listener rotates his/her head in real-life, sensory input to the brain changes, and a perceptual experience of space occurs. Developing this type of spatial effect in audio is not yet readily accessible and is increasingly necessary in the context of virtual reality.

The goal of this project is to create a device capable of augmenting an existing pair of over-the-ear headphones. This device would allow for the ability to track head orientation so that composers can create music in a 3-D space, while listeners can have a more immersive and realistic experience that varies with the motion of the user's head.

1.2 Background

There are technologies, such as virtual reality (VR), that have implemented the concept of head-tracking. In addition, some museums have even started to integrate 'engaging' experiences in exhibits that entail a walk through tour with sounds. Evidently, this sort of technology and experience is gaining in popularity and is increasingly sought after for development. For this reason and for the sake of innovation, the project we propose combines these ideas: head-tracking for engaging sound experiences.

Currently, such products that use this technology are the Oculus Audio SDK [2] where spatialization and head-tracking can modify monophonic sound sources to make them sound as if they were originally from a specific desired location. VR headsets also have volumetric sources, which produce sounds that vary in magnitude depending on the user's virtual distance from them. The varying magnitude of virtual sources will simulate distance for the use in a virtual space, as the sounds will be spread out and change over the space as one virtually moves.

Another relevant existing project includes the wearable FLORA project. The FLORA project uses a high precision 3-axis accelerometer and compass sensor. In our project, we will also include both the accelerometer and a magnetometer. How the accelerometer works is it will measures the headphone's acceleration forces such as gravity, and most importantly dynamic motion of the user. The output for the accelerometer sensor help determine roll and pitch, while the addition of the magnetometer data help give yaw. As such, the accelerometer and the magnetometer must function in unison for this project to be successful.

1.3 High-Level Requirements

- The device should perform head-tracking in “real-time,” i.e. send data a refresh rate that is suitable: at least 30 Hz, but preferably 60 Hz or more.
- The device must have a button to calibrate/reset the system so that the direction that the user is facing when the button is pressed is considered “forward-facing.” This means that angle data (roll, pitch, yaw) should be sent as 0 radians.
- The device must be able to send angle data accurately up to ± 3° (or about 0.05 rad) within the absolute measure. The absolute measure for angle can be measured independently and compared to the output data.

2. Design

2.1 Block Diagram

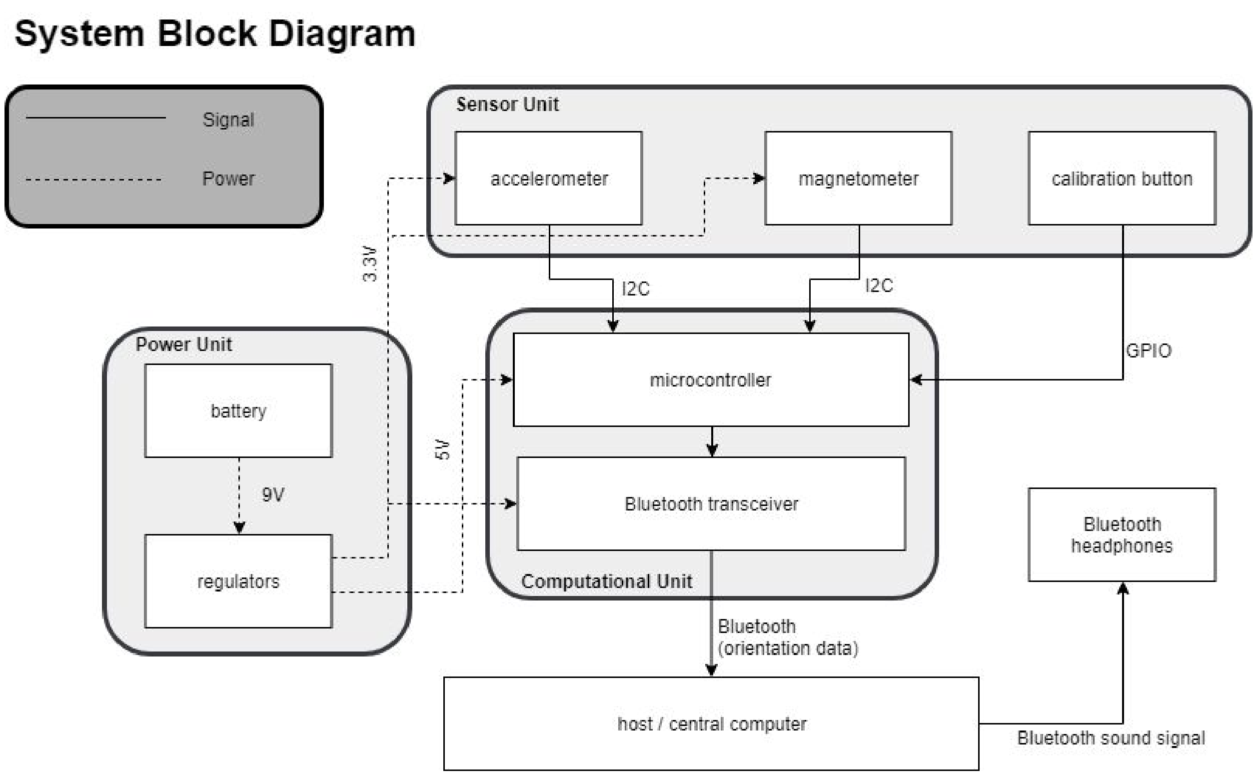

Figure 1. Block Diagram

The augmenting device has three main subsystems: the power unit, the sensors, and the computational unit. The power unit consists of a 9V battery connected to a regulator, which provides the necessary voltage to the sensors and computational unit. The sensors unit is made up of an accelerometer, magnetometer, and button. Together the accelerometer and magnetometer take and send data to the computational unit that help indicate angle, i.e. head-tracking data. The sensitivity of these individual sensors is key to obtaining the accuracy as defined by the high-level requirements. They should should also send data fast enough to facilitate the real-time, immersive effect. The button input serves as the signal to the microcontroller that the device should be calibrated. The computational unit is made of the microcontroller and the bluetooth transceiver. These process the data from the sensors (calculate head orientation data and recalibrate when necessary) and allow the transmission of this data at a high enough rate for achieving a realistic effect the host computer. From there, the host computer sends the appropriate sound signal to the user's headphones. The host computer and Bluetooth headphones are separate entities from the augmenting device.

2.2 Physical Design

Figure 2. Physical Design of Device Working within the Whole System

The computational unit, the power unit, and the sensor unit will all exist in one chassis attached to the top of a pair of off-the-shelf, over-the-ear, Bluetooth headphones. The headphones and the box will not interact electronically; they will communicate with the host computer using separate Bluetooth signals. Somewhere in the same room the user is wearing the headphones, there will be the host computer. The host computer receives the orientation information from the sensor box and sends a sound signal to the Bluetooth headphones.

2.3 Block Descriptions

2.3.1 Computational Unit

2.3.1.1 Microcontroller

The microcontroller receives and processes data from the accelerometer, magnetometer, and button to produce the relative orientation of the headphones to relay back to the host computer.

The button status is used to set the origin state of all three orientation data points. If the button is pressed, the current orientation of the headphones becomes the new zero from which to measure all deviations from that initial state. The orientation vector (roll, pitch, and yaw) of the headphones is deduced by the data from the magnetometer. The magnetometer acts as a 3-axis digital compass and relays information about the direction of magnetic north relative to the orientation of the sensor on 3 axes. A Weiner filter, a signal processing technique, will be utilized to ensure the orientation data is stable and as accurate as possible. The orientation data (yaw, pitch, roll) is sent in radians as floating-point values to the Bluetooth transceiver.

Requirement	Verification
1. The microcontroller must return frames of data at frequency with minimum 30 Hz (may be greater frequency) to Bluetooth transceiver.	1. A. Create a short Python script that retrieves the times between signals received from the Bluetooth signal to the main computer. B. Connect Bluetooth transceiver to computer. C. Run script and confirm that output times from the script are at most 0.2 second intervals. (Note: this should be tested after testing the Bluetooth transceiver).
2. The microcontroller must be able to program and drive the Bluetooth transceiver with connection with baud rate of at least 11520 (6 float-64 values per frame with a frame rate of 30Hz).	2. A. Confirm Bluetooth signal has been received from the device to the main computer. B. Create Python script that monitors incoming Bluetooth signal C. Confirm from the output of the script that the minimum baud rate supports 6 double values at 30Hz.

2.3.1.2 Bluetooth Transceiver
Relays orientation information from the computational unit to the host computer over wireless Bluetooth connection.

Requirement	Verification
1. The Bluetooth transceiver must sustain a connection with the host computer for distance at least 5 meters away in a standard indoor room environment.	1. A. Pair Bluetooth transceiver to host computer. B. Walk a distance of at least 5 meters radially away, measured using a tape measure or meter stick, from the host computer with the Bluetooth transceiver. C. Check to see if host computer is still connected to the device. The device should indicate whether it is still connected and transmitting data from an LED indicator.
2. The Bluetooth transceiver should be able to send data with baud rate of at least 11520 (6 float-64 values per frame with a frame rate of 30Hz).	2. A. Create Python script on the host computer that will calculate the average time between two received values. B. Connect transceiver to the host computer and run the script. C. Graph the results showing time between consecutive received data points to confirm data rate.

2.3.2 Sensor Unit
2.3.2.1 Accelerometer
Records data on 3-axis acceleration of the sensor box and relays information to the microcontroller over an I2C connection.

Requirement	Verification

1. The accelerometer should return data with a frame rate of at least 30 Hz.	1. A. Create Python script on the host computer that will calculate the average time between two received values. B. Connect device to computer. C. Run script and verify the frame rate from the output of the script. (Note this should be tested after the Bluetooth transceiver and the microcontroller have been tested.)
2. The accelerometer must return data on 3 axes to microcontroller.	2. A. Create Python script to view incoming acceleration data. B. Flip device 180° vertically so that 2 g of acceleration occurs on device. C. See that this is shown from the incoming data. D. Repeat for each axis.

2.3.2.2 Magnetometer

Records data on 3-axis direction of magnetic north relative to the orientation of the sensor box and relays information to the microcontroller over an I2C connection.

Requirement	Verification
1. The magnetometer should return data with a frame rate of at least 30 Hz.	1. A. Create Python script on the host computer that will calculate the average time between two received values. B. Connect device to computer. C. Run script and verify the frame rate from the output of the script. (Note this should be tested after the Bluetooth transceiver and the microcontroller have been tested.)

2. The magnetometer must return data on 3 axes to microcontroller.	2. A. Create Python script to view incoming magnetometer data. B. Rotate the device 180° horizontally so that magnetic field data has reversed directions. C. See that the coordinates of the magnetic north signal have inverted signs.

2.3.2.3 Calibration Button

Sends a signal on its state (pressed or not pressed) to the microcontroller over GPIO connection.

Requirement	Verification
1. The button must send digital high signal to microcontroller when pressed.	1. A. Create Python script on the host computer that will display the status of the button. B. Connect device (sensors, computational unit, power unit) to computer. C. Press the button and confirm data change from output of script.

2.3.3 Power Unit

2.3.3.1 Battery

A standard, off-the-shelf 9V battery as the supply for the regulator.

Requirement	Verification
1. The battery must be within 5% of the nominal voltage of 9V.	1. A. Probe the terminals of the battery with multimeter. Note down voltage reading B. Repeat at least 3 more times. C. Ensure that the average of the collected data is within 5% of 9V.

2.3.3.2 Regulator

The regulator will take the 9 volts from the battery and produce a stable 5 and 3.3 volts power supplies for the devices within the sensor box (accelerometer, magnetometer, bluetooth transceiver, and microcontroller).

Requirement	Verification
1. The 3.3V regulator must be capable of delivering at least 0.60 A.	1. A. Create test load using a 5Ω resistor. B. Connect multimeter in series to measure the current. C. Ensure that the current is never below 0.60 A.
2. The 5V regulator must be capable of delivering at least 0.50 mA.	2. A. Create test load using a 10kΩ resistor. B. Connect multimeter in series to measure the current. C. Ensure that the current is never below 0.50 mA.
3. The 3.3V regulator must output 3.3V ± 0.3V DC supply.	3. A. Attach 10Ω resistor as a load. B. Attach oscilloscope across load. C. Supply regulator with 9V DC. D. Verify that the voltage is between 3V and 3.6V.

4. The 5V regulator must output 5V ± 0.5V DC supply.	4. A. Attach 10kΩ resistor as a load. B. Attach oscilloscope across load. C. Supply regulator with 9V DC. D. Verify that the voltage is between 4.5V and 5.5V.

2.3.4 Host/ Central Computer

The host computer will receive orientation data from the computational unit over Bluetooth signal. Using this data, the host computer will produce and process the appropriate audio signal according the composer's soundscape. The host computer sends this audio signal to the Bluetooth headphones.

Requirement	Verification
1. The host computer must maintain 2 Bluetooth connections.	1. A. Open settings on host computer to verify that it has bluetooth capabilities.
2. The host computer must be capable of streaming audio over Bluetooth.	2. A. Pair the Bluetooth headphones to the host computer. B. Put headphones on a test user. C. Pull up any audio file on the host computer and play the sound. D. Audio streaming ability should be confirmed by the test user.
3. The host computer must be capable of running SuperCollider development platform.	3. A. Check to see if computer can run SuperCollider. If not, it can be installed for Mac, Linux, and Windows operating systems.

2.3.5 Bluetooth Headphones

The Bluetooth Headphones will be a pair of off-the-shelf, over-the-ear headphones with Bluetooth streaming capability. These headphones will simply play the signal from the host computer into the user's ears.

Requirement	Verification
1. The Bluetooth headphones must be be over-the-ear headphones.	1. A. Verify this by visual inspection.
2. The headphones must be able to stream Bluetooth audio to your ears.	2. A. Connect headphones to the host computer. B. Play any audio file. C. Verify that the sound can be sustained.

2.4 Circuit Schematics

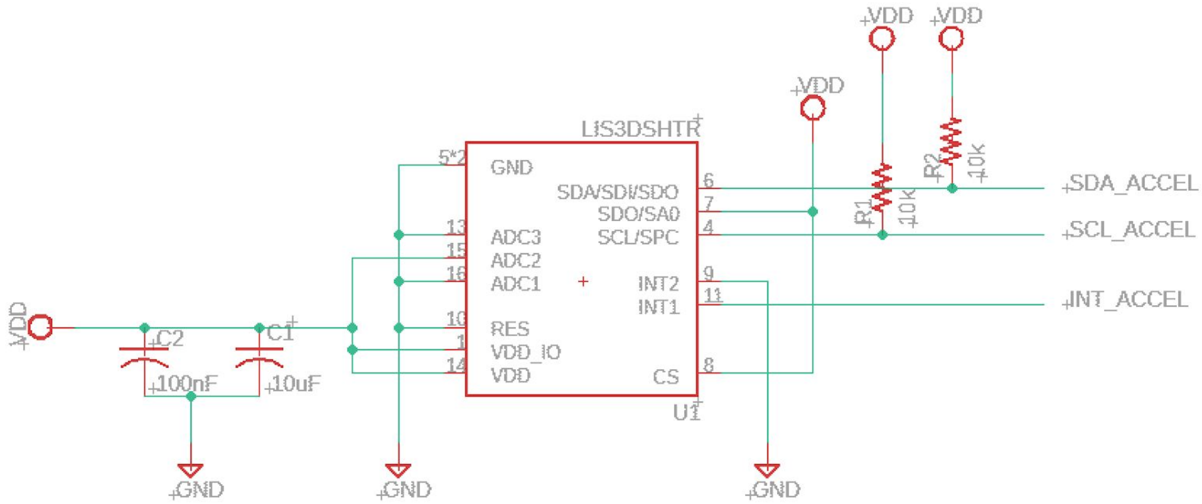

Figure 3. Accelerometer Schematic

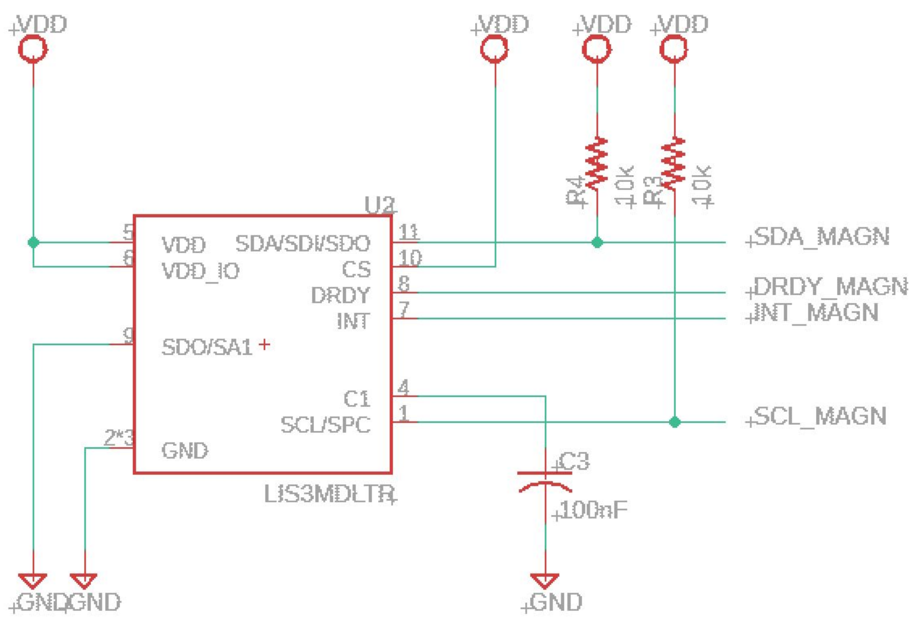

Figure 4. Magnetometer Schematic

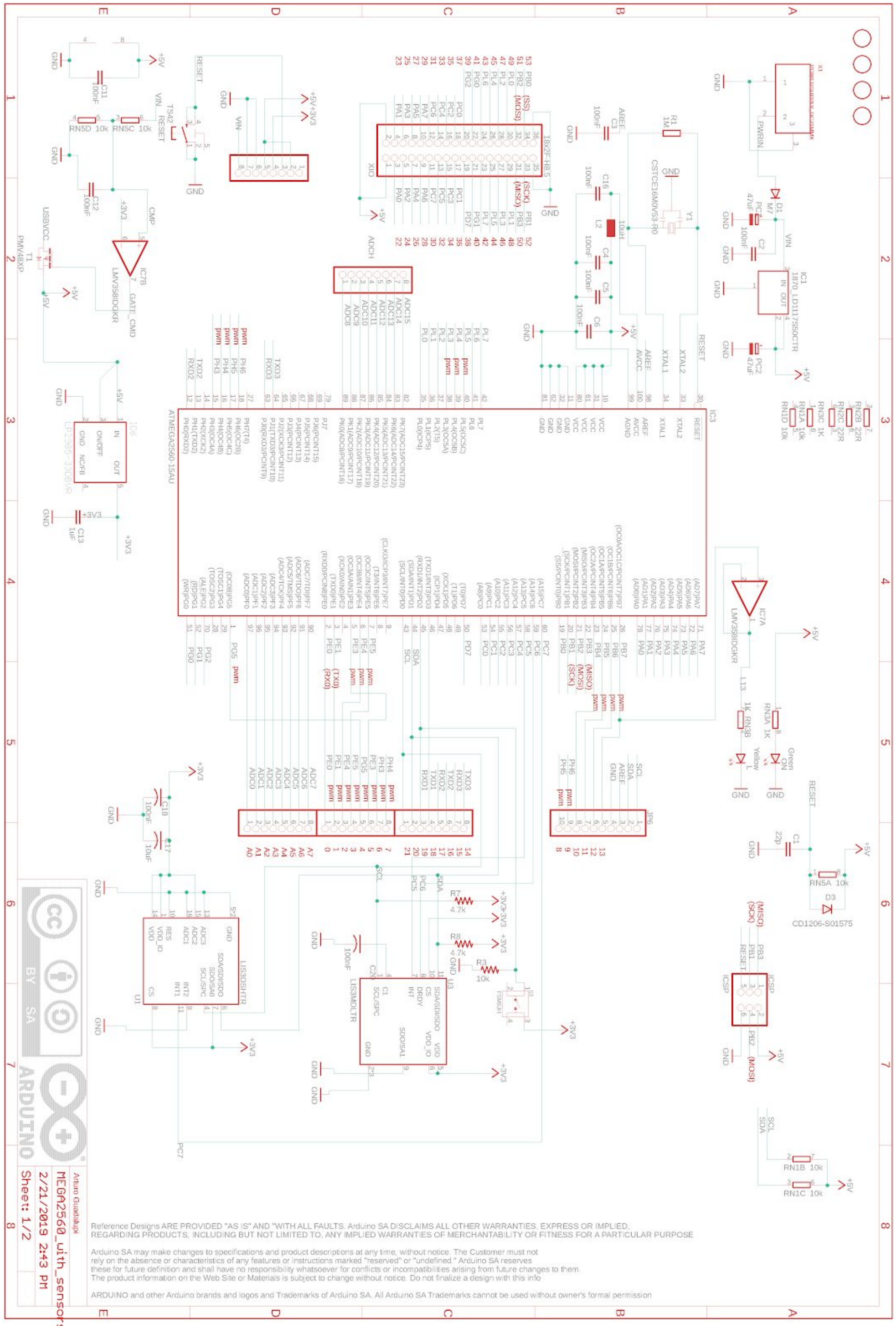

Figure 5. Sensor and Microcontroller Schematic

2.5 Calculations -- Software Algorithms

2.5.1 Orientation Calculation

Roll Pitch and Yaw Calculations

Given gravity vector G with elements (G_1, G_2, G_3) and given magnetic north vector N with elements (N_1, N_2, N_3), Roll, Pitch, and Yaw are calculated as follows:

$$\theta_{roll} = \arctan(\frac{G_2}{G_3})$$

$$\theta_{pitch} = \arctan(\frac{-G_1}{\sqrt{G_2^2 + G_3^2}})$$

$$\theta_{yaw} = \arctan(\frac{N_2 \cos(\theta_{roll}) + N_3 \sin(\theta_{roll})}{N_1 \cos(\theta_{pitch}) + N_2 \sin(\theta_{pitch}) \cos(\theta_{roll})})$$

2.5.2 Signal Processing and Filter Design

Given that the signal will be coming from a real-world sensor, we can expect a certain amount of noise from the signal. Since the calculation accumulates error as time passes, it is important that we denoise our signal as best as possible. We will use a Wiener Filter to denoise the signal. In the calibration phase of our project schedule, we will acquire data on the measured orientation of the device and the true orientation of the device by moving the device to specific test angle and recording the outputs. This will give us a good understanding of the characteristics of the sensor noise that we can filter. We will use this test data to form the power spectrum for the noise and the power spectrum for the signal. We will assume white noise so our filter frequency response will be a lowpass filter as follows:

$$G(\omega) = \frac{P_s(\omega)}{P_s(\omega) + \sigma_n^2}$$

Where $P_s(\omega)$ is the power spectrum of the signal and σ_n is the variance of the noise.

2.5.3 Calibrating Front Orientation

The roll, pitch, and yaw calculations above are absolute angles based on the direction of gravity and magnetic north. Our user would like to calibrate a front setting using a button. The calibration button will save the absolute roll, pitch, and yaw values of the user's current orientation which will be the "front" orientation for the soundscape. When the user moves to a new orientation, we will calculate the new roll, pitch, and yaw values by finding again the absolute values and then subtracting the saved front roll, pitch, and yaw values. This will yield an orientation angle system that is relative to the calibrated front.

2.5 Tolerance Analysis

Purpose

The main purpose of this system is to track the orientation of the user's head using accelerometer and magnetometer data. The orientation data will be used to produce a 3D soundscape experience in real time that will change based on the orientation of the user's head. It will be important that the three axes of rotation are sufficiently accurate to generate a convincing user experience.

Human Auditory System

The main concern is whether the system will return data accurate enough for any users listening to the soundscape to be convinced that their orientation matches the sounds they hear through the headphones. Aspects of the Human Auditory System will inform how accurate the system must be. Humans can localize sounds with an accuracy of about 3° when the sound is near (less than 10 feet away). With this in mind, for the user to hear the soundscape accurately, we need to determine their head orientation within ± 3° of their actual orientation for every data frame [1].

Error as A Function of Time

The user experience will be directly affected by the real-time output of our sensors. If the sensors produce unstable data, the user could experience a jumpy audio signal that does not accurately capture the soundscape. It will be important that the derivative of the error remain low and that at each dataframe, the error stays below ± 3°. Smoothing will be necessary. In order to achieve a smooth experience for the user, a 3-tap digital low pass averaging filter will be employed on the microcontroller end to ensure that change between any two frames is not extreme.

Distribution of Error

The distribution of error between the readings and the actual head orientation may be of any type of distribution. It would be best if the error was a Gaussian distribution with an expected value of zero, but it is a necessity for the success of the project. Since a human's auditory capabilities are only so accurate at localizing sound, if the expected value of the error distribution lies anywhere ± 3°, it will successfully convey a believable soundscape experience. It is necessary that none of the frames of data ventures outside of the ± 3° tolerance. These frames, especially if the error persists for longer than 1 frame, will be noticeable to the user.

Sensor Capabilities and Error Calculation

We will be employing a ST Microelectronics LIS3MDL 3-axis magnetometer. This magnetometer has a sensitivity of 6842 LSB/gauss, meaning 0.00014 gauss per least significant bit in each axis register and RMS noise of 0.0032 gauss. The magnetic field is between 0.25 and 0.65 gauss. In a worst case scenario, assume it is 0.25 gauss and assume the magnetometer axis readings are off by mean noise and one total LSB: 0.0032+0.00014=0.00334 gauss. Each axis of the magnetometer is a dot product of the earth's magnetic field direction and the direction of the axes. In the worst error situation, two axes of the magnetometer are perpendicular to the EM field. The coordinate readings should be (0,0,0.25) but could be as bad as (0.00334,0.00334,0.25334) gauss. But since we will be calibrating from an original orientation, the difference between the readings could result in error as bad as twice this. (0.00668,0.00668,0.25668). Coordinate calculations for the perfect situation would be (0deg,0deg,0deg). The error incurred by the worst case scenario in degrees would follow the formula of the angle between two vectors: $arccos\left(\frac{u \cdot v}{|u||v|}\right)$. In this case that the difference in angle would be 1.49° which is well within our allowed tolerance. In application, incorporation of the gravity vector obtained from the accelerometer can help to create an even more stable signal with higher accuracy. More on this discussion in the Calculations section.

3. Cost

3.1 Labor

Assuming an average salary of $40/hr per person and that each person works 10 hours/week over the semester (16 weeks), and taking into account a 2.5 multiplicative factor (which accounts for overhead costs, setbacks, etc.) the total manual labor cost comes out to be $48,000. This calculation is demonstrated in Table 1.

Name	Hourly Rate	Hours	Total	Total x 2.5
Molly Fane	$40	160	$6,400	$16,000
Sally Zhou	$40	160	$6,400	$16,000
Cary Zhu	$40	160	$6,400	$16,000
Total				$48,000

Table 1. Labor Costs

3.2 Parts

Description	Manufacturer	Part Number	Quantity	Cost/unit	Total
3-Axis Magnetometer	STMicroelectronics	LIS3MDLTR	1	$1.67	$1.67
3-Axis Linear Accelerometer	STMicroelectronics	LIS3DSHTR	1	$2.01	$2.01
Microcontroller Chip- ATMega	Microchip Technology	ATMEGA2560-16AU	1	$12.35	$12.35
3.3V Regulator	ON Semiconductor	NCP565D2T33G	1	$1.35	$1.35
5V Regulator	STMicroelectronics	LD1117S50CTR	1	$0.45	$0.45
Button for Calibration	TE Connectivity ALCOSWITCH Switches	1825910	1	$0.13	$0.13
Bluetooth Chip	Espressif Systems	ESP32-WROOM-32D	1	$3.80	$3.80
9V Battery	AmazonBasics	6LR16-8PK	1	$1.19	$1.19
Miscellaneous resistors, capacitors, connectors	Digikey	Assorted	40	$0.05	$2.00
PCB	PCBWay	N/A	N/A	$150	$150
Total					$174.95

Table 2. Component Costs

3.3 Grand Total

Taking into account the cost of the manual labor and the component costs, the total cost is $48,000 + $174.95 = $48,174.95.

4. Schedule

Week	Fane	Zhou	Zhu
2/18/19	Work on design document: supporting materials (e.g. schematics), tolerance analysis	Work on design document: power unit R&V, review proposal feedback and make changes to design document	Work on design document: computational unit and sensor unit R&V
2/25/19	Create PCB design, begin writing software code	Review PCB design, order parts	Begin writing scripts needed for verifications
3/4/19	Begin assembly of computational unit, program firmware on device	Begin assembly of sensor unit, contact and consult with Prof. Fieldsteel	Begin testing and verification of computation unit
3/11/19	Begin testing and verification of power unit	Begin testing and verification of sensor unit	Begin assembly of first prototype, create encasing for system
3/18/19	Continue testing and verification of power unit, start second PCB design	Continue testing and verification of sensor unit, debug software code	Debug firmware on device chip
3/25/19	Finalize second PCB Design	Review and order second PCB	Collect and manage data from testing
4/1/19	Assemble second iteration of computational unit, begin modular testing	Assemble second iteration of sensor unit, begin modular testing	Assemble second prototype with encasing
4/8/19	Demonstrate testing of second prototype	Set up meeting with Prof. Fieldsteel, demonstrate progress, make note of any concerns	Fix issues brought up by Prof. Fieldsteel regarding software
4/15/19	Consolidate bugs in firmware, prepare for mock demonstration	Start working on presentation	Finalize device for demonstration, begin final report
4/22/19	Review and make changes to final presentation	Continue working on final report	Review final report and final presentation
4/29/19	Review final report, especially technical style and tone	Review and make necessary changes to final report, make final submission	Finalize document formatting

Table 3. Project Schedule and Responsibility Allocation

5. Ethics and Safety

The system we propose to create involves creating a product that will attach to existing headphones. Should there be poor power management, a malfunction could pose a large safety risk, as the user would be wearing the product, and could potentially damage the user's headphones. To adhere to the IEEE Code of Ethics, we would disclose this fact in its release to the public as a potential risk with the use of this product [3]. Additionally, the risk of injury should be minimized with the design being low power.

Based on our current design, we will be using a 9V battery. If metals or any conductive materials come in contact with the batteries, then a short circuit can occur and lead to safety hazards, such as a fire. Our design would ensure that the battery will be in its own compartment, so as not to come in accidental contact with other materials.

Another safety risk this design may pose may stem from the fact that it is to be mounted on top of over-the-ear headphones. Poor mounting may cause the whole assembly to fall off the user, causing unpredictable damages or outcomes depending on the environment the user is in. Again, this risk should be communicated with the release of the product to adhere to the IEEE Code of Ethics: "to disclose promptly factors that might endanger the public [...]" [3].

Since the completed project will likely be physically small, the device could become a choking hazard should it come in contact with infants. We will make sure that any mounted pieces on the headphones will be securely intact, so that no small pieces come loose, to minimize this risk.

6. References

[1] R. A. Dobie and V. H. S. B., Hearing loss: determining eligibility for Social Security benefits. Washington, DC: National Academies Press, 2005.

[2] Introduction to VR Audio. [Online]. Available: https://developer.oculus.com/audio/. [Accessed: 08-Feb-2019].

[3] Ieee.org, "IEEE Code of Ethics", 2019. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 7- Feb- 2019].