Amica Aura:
Wireless Headphones with Audio Sharing

Ethan Bless-Wint, Winston Ching, Michaela Laurencin

Electrical and Computer Engineering Department, Carnegie Mellon University

***Abstract*—Amica Aura is a wireless headphone system that has active noise cancellation as well as audio sharing as additional capabilities. The system also features gesture controls and wireless charging. With this system, users can listen to audio, host a broadcasting session of their own audio, or join a nearby mesh station another user is broadcasting.**

***Index Terms*—Headphones, Bluetooth, Mesh, Active Noise Cancellation, Wireless**

I. Introduction

AMICA AURA is set of wireless headphones containing active noise cancellation and wireless audio sharing capabilities. Audio prosumer technology has experienced rapid development in the last few years. Specifically, the rise of headphones has dramatically increased the convenience of on-the-go listening. In this new age of auditory freedom, one area that has still not been fully developed is the ability to share. With our project, we would not only be able to have a device that can satiate this growing desire for wireless audio technology, but also fill the void of communal audio enjoyment that is in the current market. Competing products do exist, but they miss some aspect of this demand that our product covers, either the mobile nature, isolated sharing, or ability for standard headphone usage.

We have chosen to optimize four particular areas of our project to achieve a successful audio listening and sharing experience: sound quality, audio sharing, usability, basic level functionality. On the whole, we aim to achieve a pair of headphones capable of (a) in normal playback mode, playing 48kHz stereo audio at a bitrate of 345kbps via Bluetooth while maintaining a distortion factor of smaller than one thirtieth and a signal-to-noise ratio of no worse than 6dB when compared against Bose QuietComfort 35 version 2; (b) detecting users' in-air hand gestures within 3cm of the right module with electrodes carrying 115kHz oscillating electric field and recognizing said gestures within 1s after their completions; (c) charging wirelessly at 5W or connectedly at 15W; (d) in multi-playback mode, broadcasting 345kbps stereo audio to at least two other headphones simultaneously with less than 30ms of latency (and less than 180ms for six other headphones in a mesh network) while maintaining robust enough of a connection such that any node that is not the sound source can fail with no longer than 10s of an impact on the streaming of audio.

II. Design Requirements

The main aforementioned design requirement, paired with the requirements for the four broken down aspects will make up the total requirements for our system in order to be comparable to market standard.

Firstly, for the area of Sound Quality, our metrics related to the sub-features of low-noise circuitry, active noise cancellation, and passive noise isolation. For low-noise circuitry, we will need to keep the complexity low and compact such that it can fit within a 1800mm2. We will also need to keep the power consumption less than 125mA during either playback mode. Finally for the circuit, it must have a difference in signal-to-noise ratio of less than 6dB and total harmonic distortion plus noise of less than 1% as compared to the Bose QuietComfort 35 version 2's Related to active noise cancellation, the signal-to-noise ratio between our output and the input audio needs to be at least 12dB and the total harmonic distortion must be less than 3% in order to achieve successful cancellation. And lastly, for the passive noise isolation in the form of cushioning, it needs to be less than $120cm^3$ in volume and must have at least a 6dB sound pressure level reduction in order to provide both comfort and assist in the noise cancellation.

The audio sharing system will send audio from a broadcasting headset, or root node, to a series of receiver headsets, or sub-nodes. The latency from the root node to any given sub-node must not exceed 180ms. The failure of any given sub-node should not cause a disruption to system operation exceeding 10 seconds. In order to meet the bandwidth requirements of the chosen codec, SBC a minimum effective link speed of 450kbps is required. In this way, users will not experience noticeable desynchronization or disturbance in their audio.

Regarding usability, we will need to have a number of metrics met for battery life for a successful experience. We have determined a playback time of over 40 hours, a multi-playback time of 6 hours and a standby time of over 450 hours should facilitate this. For charging we are looking to have a USB charging unit within less than $900mm^2$ and have a performance of 3A (5V), and a wireless charging coil within a dimension of 40mm by 60mm and a performance of 1A (5V).

Lastly for the basic functionality we are looking Bluetooth playback conform to the A2DP profile and the overall power consumption remain less than 125mA during playback.

III. Architecture and/or Principle of Operation

Our device consists of a digital audio playback system with a novel digitally configurable analog noise cancellation system. Each headphone half consists of an ESP32-WROOM-32D module, an analog noise cancellation system with feedback topology, a battery, and an amp-and-driver pair. The headphones halves are connected via a ribbon cable that transmits power and data. Some portions of the headphone are asymmetric. One half will contain a 3D gesture sensing subsystem, and the other will contain wireless charging circuitry.

The ESP32-WROOM-32D module is in charge of the headphone and is responsible for configuring all the other hardware. The two ESP32-WROOM-32D modules are arranged in a master-slave pair, with once accepting commands from the other. The modules are responsible for controlling the other hardware inside the headset, as well as wireless reception, transmission and local playback of audio. The amp and driver receive an I2S signal from one of the two ESP32 modules, and then decode and amplify the signal, sending it to the drivers. Each headphone halve also contains a 2500mAh battery for power, as well as a USB-C port for charging and UART based debugging and firmware updates. The 3D gesture sensing subsystem emits an electric field and detects perturbations to the field, which allow the subsystem to recognize in-air gestures of the users of our device. The ESP32 module can then react according to the recognized gestures per a predefined FSM.

A. UI and UX

The device exists in two main modes: Single Playback and Multi-Playback. Within the Single Playback mode, users can choose to either play their audio as they would with any normal pair of headphones or enter the branch Broadcast mode. Upon activation of this broadcast mode, the user will trigger the creation of a mesh system created over WiFi. This mesh is then joinable by other headphones, creating a tree topology for the relaying of audio packets with the broadcasting headphone as the root node. In the second main Multi-Playback mode, the user can choose from the already established meshes that exist immediately around it, adding on as a leaf to the tree that has already been established. The switching between these modes will be governed by inputs to the MGC3130 gesture control module and the Bluetooth/WiFi communication for the audio transfer will be controlled by the ESP32 module. Across all modes active noise cancellation will be present using the RLS Adaptive Filtering for the most optimum audio listening experience.

B. Software

The software itself consists of multiple tasks controlled by a Free RTOS scheduler. The software is responsible for accepting, routing, and dispatching network packets, configuring multiple aspects of the headphone hardware (including the analog filter, DAC/Amp, power button, and status LEDs), and sending audio packets to the audio hardware. The software stack will consist of the mesh network stack, drivers for Bluetooth audio, drivers for I2S audio transmission, driver for intra-headphone communication, an FSM for handling the user states, and an algorithm to adjust the configurable analog filters in each pair of headphones.

C. Mesh Network Design

The mesh network allows for a root to disseminate music to multiples sub-nodes. It is a tree-based topology that is optimized for broadcasting from the root node to the sub-nodes. The mesh topology will be supported by multiple daisy-chained 54mbps WiFi networks. Each one-to-many collection of parents and nodes will comprise a separate WiFi network. Each node is capable of acting as both an AP and a Station. The parent nodes in the mesh will broadcast a special SSID that will indicate it is part of a mesh as well as what layer it occupies. Each root node will set a nonce in the SSID to differentiate separate mesh networks. All the parent nodes in the mesh topology subtree will have the same nonce bits in the SSID as the root node of the mesh. 15 bytes of the SSID will be reserved for the nonce. Since characters are restricted to the alphanumeric subset of ASCII characters, each byte can be one of 62 values. Thus, assuming a high-quality source of randomness, there is a 1 in 7.689097049E26 chance of collision for each additional overlapping mesh network.

Each node that wishes to join the mesh will first scan the SSIDs to discover parent nodes. Then it will contact parent nodes, going in ascending layer order. It will connect to the first parent node that indicates it has not reached the maximum node limit. A node will be considered lost if it does not receive data for 3 seconds. The node will then reset itself and try and join the mesh as if it was new.

The mesh will route data between nodes using a custom IPv6/ICMP stack. Not all of the packet fragmentation and broadcast features of IPv6 will be supported. In order to support routing, each node will maintain a list of its neighbors, as well as routes for all the IP addresses it is aware of in the mesh. In order to support smooth playback, the mesh network must support a minimum effective link speed of 450kbps to all nodes, as well as a reconnection speed of 10 seconds, and a packet jitter below 40ms.

IV. Design Trade Studies

A. Device Formfactor

One of the most prominent challenges in this project is accommodating not only the components required for supporting all the aforementioned features, but also the battery needed to power said components. To add to the challenge, both wireless charging and 3D gesture sensing require large flat non-conductive surfaces made available on our device, to house a inductive coil and five electrodes respectively. Moreover, to be competitive with existing products in the market, our device must be in a reasonable formfactor such that it is not too bulky to fit on our users' head. As such, we jump-started our design process by scrutinizing an artist's rendition of a pair of headphones and then physically laying out various subsystems within the predetermined enclosure. Though this would not function as our final enclosure design, it gave us a rough idea on the constraints in terms of real estate that we must conform to, which cannot be obtained otherwise.

B. Hardware Considerations

Due to the aforementioned constraints in real estate, we have a stringent constraint on the size of our electronics. As a result, we must design our own PCB with SMD components in lieu of breadboards or perma-proto boards with breakout boards and through-hole components. Such a design decision incurs a large amount of engineering complexities that must be accounted for.

Moreover, within PCB design, there is a constant struggle among reducing the components' footprints, minimizing the number of different components, and avoiding extremely expensive parts. For example, a highly integrated IC might reduce the external passive components needed and thus reduce power consumption, number of components, and area occupied. However, such an IC could be prohibitively expensive due to budgetary constraints.

To aggravate the challenge, the optimizations as described above are non-linear, since trade-offs in one subsystem do not necessarily correspond linearly to trade-offs in another subsystem. For example, the integration of battery charging, monitoring, and protection and load-balancing circuitry into a smaller footprint allows for a more efficient design of power supply due to freed-up space, which in turn permits high power consumption elsewhere in the headphones.

C. Software Design

The software has significant considerations with respect to the time of this project, as well as the compute power of the system. One of the most powerful mechanisms the design has is the intra-headphone communication path via I2S. This is primarily used for forwarding I2S packets between headphones, but it can also be used to enable complicated operations such as dual radio network operation and dynamic task scheduling. However, we've restricted its use, and the corresponding interactions between ESP32 modules for the sake of time and power consumption. The system has a 40hr battery life target, so the software cannot be wasteful, especially with wireless data transmission and complicated operations. To that end, the software has the ability to utilize the ultra-low-power coprocessor in the ESP32 when idle to avoid needless power consumption by entering "deep sleep." Also, the software has to be limited with respect to our time for implementation and debugging. It is difficult to debug embedded software, so we must go with more simple designs so that they can be constructed and debugged in a reasonable amount of time.

D. Mesh Network Design

The effective link speed is the number of packets from the root node that can reach a given node in 1s. This is independent from the link speed of a single WiFi connection, as it takes into account the latency of the entire mesh infrastructure. A node that has a low effective link speed may still be able to play back audio smoothly, but it will miss the latency targets.

$$\text{Effective Link speed} = \frac{min(Linkspeed_i)}{N} - Nc$$

Fig. 1. N is the number of nodes between the target node and the root node. Link Speed is the speed of the connection between two nodes. C is a constant that represents the average processing time for a packet when it reaches a node.

Effective link speed is a critical metric and drives mesh design. A previous WiFi proposal, *WiFi Proposal B* was abandoned because it did not have a high enough effective link speed. WiFi proposal B was a novel connection-based design that has redundant links for data transmission. It was optimized for broadcasting from the root node and had no support for node-to-node or node-to-root communication. The network could sustain two node disconnection/failures per 10 seconds without disruption. However, that design (and thus its reliability) were abandoned in favor of the current design because it was too slow.

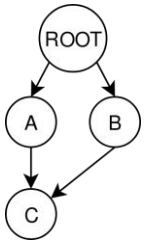


Fig. 2. *"While audio does not require a particularly high data-rate, the paltry 1 MBPS link speed could prove troublesome. Consider the path from Root to C. The link speed is only 500kbps from ROOT→A and from ROOT→C if we consider a packet processing time of 10ms (c=0.03125) we get an effective link speed of 186kbps, which is already below our requirement. Node C will get eventually get all the audio packets, as each individual link has enough speed. However it will have a latency of roughly 367ms. This latency shortcoming cannot be overcome by using multicast packets, as wifi has significant issues with multicast reliability that could severely impact playback performance. These issues make it difficult to recommend implementing this mesh design."* - Two proposals for Mesh Networking on ESP32

E. Active Noise Cancellation

Due to the aforementioned constraints on signal-to-noise ratio and total harmonic distortion plus noise, two algorithms were considered in order to create the necessary filters to achieve this. The first of which was least mean square (LMS) algorithm. With this algorithm, one can create an adaptive cancelling algorithm, but the main issue is that it is a stochastic algorithm. In this case, there is some degree of randomness that exists within the model itself and it does look to other past data to create the current filter. In comparison, recursive least squares (RLS) filtering is an adaptive filter algorithm that is deterministic, in that the output is fully determined by the initial condition and the corresponding input parameters. It also is shown to converge faster than LMS. With these two main facts, as well as the following mathematical definition of RLS, for our purposes of a fast-responding algorithm that updates its filters with new incoming information, RLS was the best of the available options.

The approximation of FIR filters produced from RLS with an analog counterpart is a non-linear approximation, since the frequency response of a second-order filter is a non-linear function with respect to any one of the resistance values in said filter. Hence, a coordinate descent (CD) algorithm is used to find the best setting that can be used in the digital potentiometers such that their corresponding analog filter can achieve the desired frequency response.

Parameters: p = filter order
λ = forgetting factor
δ = value to initialize $\mathbf{P}(0)$

Initialization: $\mathbf{w}(n) = 0,$
$x(k) = 0, k = -p, \ldots, -1,$
$d(k) = 0, k = -p, \ldots, -1$
$\mathbf{P}(0) = \delta I$ where I is the identity matrix of rank $p + 1$

Computation: For $n = 1, 2, \ldots$

$$\mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-p) \end{bmatrix}$$

$$\alpha(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n-1)$$
$$\mathbf{g}(n) = \mathbf{P}(n-1)\mathbf{x}(n)\{\lambda + \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)\}^{-1}$$
$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\lambda^{-1}\mathbf{P}(n-1)$$
$$\mathbf{w}(n) = \mathbf{w}(n-1) + \alpha(n)\mathbf{g}(n).$$

Fig. 3. *Summary of the RLS algorithm.*

V. Project Management

A. Schedule

The diagram for our schedule can be viewed on Appendix III of our report, but we will provide an overview and the relation between tasks. The design and implementation of the mesh is fairly separate from the other portions of the project until we build our first physical iteration at the beginning of April. Before this time, there will be further work done virtually for the mesh routing and the creation of bluetooth drivers for playback in general. During the time Ethan is working on this, Michaela will be working on dimensioning out the physical enclosure design so that Winston can finish his layout. Before Winston has finished this, a physical design iteration will occur, and the final design iteration will depend on the final layout. During this time as well, Michaela and Winston will work on the RLS design and hardware features of the headphones independent of each other and the previously mentioned processes. After April, the dependencies begin again as the mesh is integrated with the physical build and the firmware is tested with the hardware. After the final CAD design is developed testing will begin for the hardware and RLS, as well as the final tests and adjustments for the mesh. With that, then everything will then be iterated upon and assembled towards the end of April.

B. Team Member Responsibilities

As previously touched upon in the section VI. A and previous sections of the report, the project is broken into 3 main categories: hardware, software, and signal processing. Ethan's main focus is on the software. This includes designing and implementing the mesh, as well as creating the related firmware to support it. Finally, he is working on the high level software to support the entire system. Winston's main focus is on the hardware. This includes the designing, simulating, and testing the schematic, layout, intra-headphone communication, and/or power supply and the implementation of the filters derived for active noise cancelling. Finally, Michaela's main focus is on the signal processing. This includes, the development and testing of the RLS algorithm used to derived the necessary filters for the active noise cancelling system. Since this is relatively small related to the other areas, she is also tasked with designing the physical enclosure for the project that will then be printed.

C. Budget

The chart for our budget is viewable on Appendix II of our report.

D. Risk Management

For our project risks related to design, we did our best to do our research on similar products beforehand so as to avoid pitfalls in designs. In the case of the mesh, this included researching different forms of wireless communication to determine the optimal form for our use case. For the hardware, we attempted to stick to chip that are popular and feature more pre-established schematics. For those features that we had to use chips other than the ones previously described, we had to meticulously read the data sheets in order to develop an effective design.

For our project risks related to resources such as components, we did our best to research the components that both fit our design requirements and the requirements of the routines and algorithms we are using in order to mitigate the chance of choosing ill-fitting components.

Finally, for our project risks related to schedule, we built in multiple iterations of our design to allow for time to change any aspects that did not work the way we had planned.

VI. Related Work

As one of our main objectives was to create a pair of headphones comparable to those on the market today, there are many products similar to ours. The key thing that sets our product apart from these however is the wireless audio transfer system that is integrated in. However, there are portions of the overall system that exist in a similar way across devices.

In terms of active noise cancellation, many headphones employ a system to achieve this functionality. Comparable systems include those within the Bose QuietComfort 35 version 2, Sennheiser HD 4.50 BTNC, and the Audio Technica QuietPoint ATH-ANC700BT. More specifically, our hardware consists of a configurable set of filters controlled by a software system that establishes the desired filter for cancelling the ambient noise. This differs from the aforementioned systems however as they are of a feedforward topology, while ours contains a feedback counterpart.

Regarding the Bluetooth and wireless mesh system for audio sharing, there are two systems that utilize these technologies to a similar end. The first of which are the Sonos Multi-room speaker sets which utilize a Bluetooth/WiFi combo chip. The speakers in these sets can be paired together in order to have synchronized audio throughout multiple rooms. This technology, however, has not been replicated in any of their portable audio devices. Another example of this is an application called *AmpMe* which synchronizes audio between mobile devices in order to compete with portable speaker systems. They also employ AI and machine-learning in order to

account for things like the few fractions of a second delay that exists when transferring data between packets in order to achieve perfectly synchronized audio. For our use case however, it does not make sense to optimize in this way as users will still have isolated audio during sharing, versus their use case of creating a collective speaker system.

VII. Summary

Since we have not completed our project yet, we cannot speak to whether or not we have met all of our specifications. However our previous tests and design lead us to believe that we will be able to reach our targets.

A. Future work

Though have not completed our project just yet, we realize there will probably be a number of areas we wish to further explore. At the moment one of these is the possibility of integrating a machine-learning algorithm into the mesh design such that perfectly synchronized audio can be achieved, similar to *AmpMe*.

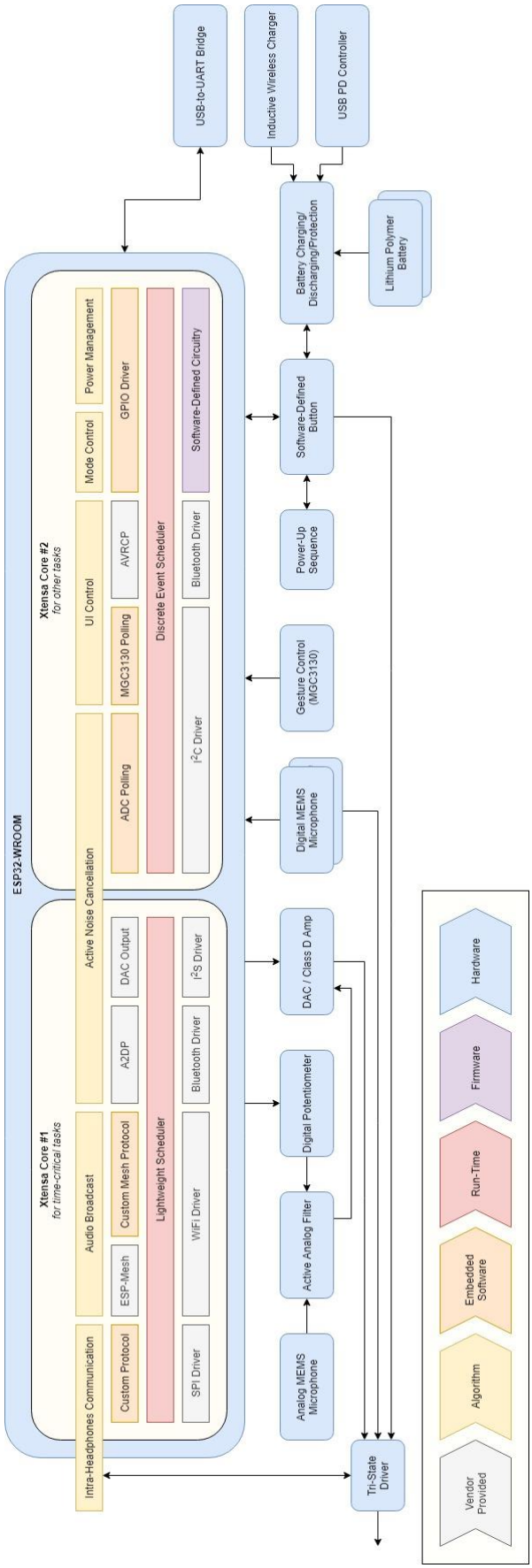
B. Lessons Learned

Similarly, to the idea of our future work, though we have not completed our project just yet, we have a few lessons we have learned thus far. The first is to establish dependencies in our Gantt chart before we finish agreeing on the work. For example, physical components of the headphones being dependent on the sizing and design of the enclosure. Additionally, when choosing components, though we want to choose the most efficient and cost effective options,we want to also be sure to cross-reference our related requirements for routines and algorithms so that the parts also comply with those. For example, when choosing our charging cable, we instinctively went for the smallest option in order to fit in our headphones, but realized that would not comply with the standard we were working with.

References

[1] https://www.ampme.com/about?locale=en_US
[2] https://venturebeat.com/2018/07/03/ampme-plans-to-kill-bluetooth-speakers-by-syncing-music-between-smartphones/
[3] https://www.mathworks.com/help/dsp/examples/adaptive-noise-cancellation-using-rls-adaptive-filtering.html
[4] https://www.mathworks.com/help/dsp/examples/acoustic-noise-cancellation-lms.html
[5] 802.11 Wireless Networks: The Definitive Guide, 2nd Edition. O'reilly Media
[6] "802.11 Association process explained - Cisco Meraki." 18 Mar. 2016, https://documentation.meraki.com/MR/WiFi_Basics_and_Best_Practices/802.11_Association_process_explained. Accessed 4 Feb. 2019.
[7] "Wi-Fi — ESP-IDF Programming Guide v3.3-beta1 ... - Espressif Systems." https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/network/esp_wifi.html. Accessed 6 Feb. 2019.
[8] https://en.wikipedia.org/wiki/Recursive_least_squares_filter

APPENDIX I: SYSTEM BLOCK DIAGRAM



APPENDIX II(A): BUDGET

Item	Cost	Qty.	Amt.
Electronics			$399.62
Ear Cushion	$10.95	3	$32.85
3D Printing			$45.00
PCB Manufacturing	$33.00	2	$66.00
Shipping			$55.00
Total			**$598.47**

APPENDIX II(B): DETAILED PARTS LIST

Type	Item	Manufacturer	Package	Cost	Qty.	Amt.
Passive Components	Capacitor (Multi-Layer Diaelectric)	Samsung	0603	$0.03	100	$3.10
	Capacitor (Film ECPU)	Panasonic	1210	$0.60	24	$14.35
	Capacitor (Film ECHU)	Panasonic	1210	$0.60	24	$14.35
	Capacitor (Film ECWU)	Panasonic	2820	$0.60	36	$21.53
	Capacitor (Electrolytic)	Nichicon	6.3mm dia	$0.55	12	$6.60
	Driver Unit (CMS-15113-076S)	CUI	15 x 11mm	$2.07	6	$12.42
	Ferrite Bead	Samsung	0603	$0.03	48	$1.44
	Inductive Coil	Abracon	N/A	$4.91	3	$14.73
	Resistor (1%; Thick Film)	Stackpole	0603	$0.07	200	$13.40
	Resistor (0R; Metal Film)	Panasonic	0603	$0.04	12	$0.52
	Resistor (0.5%; Thin Film)	Susumu	0603	$0.49	36	$17.68
	Resistor (Current Sense; 4-Terminal)	Bourns	1612	$0.61	3	$1.82
	Thermistor (NTC)	TDF	0402	$0.09	3	$0.26
	TVS Diode	Littelfuse	6-TSSOP	$1.04	3	$3.11
Active Components	LED (Green)	Liteon	0603	$0.22	24	$5.16
	NPN/PNP	Diodes	6-TSSOP	$0.32	3	$0.97
	Op-Amp (Audio, OPA1679)	TI	14-TSSOP	$1.07	6	$6.42
	Op-Amp (Measurement, INA216A4)	TI	4-UFBGA	$1.08	3	$3.24
	P-FET	Infineon	8-SOIC	$1.29	3	$3.86
Connectors & Physical Components	Battery (400mAh)	Adafruit	N/A	$6.95	2	$13.90
	Battery (2500mAh)	Adafruit	N/A	$14.95	2	$29.90
	Enameled Wire	Adafruit	38 AWG	$0.95	2	$1.90
	FPC	Amphenol	N/A	$0.43	24	$10.34
	Ribbon Cable	Molex	N/A	$1.82	12	$21.78
	Tactile Switch (1301.9319)	Schurter	N/A	$0.24	3	$0.72
	USB Type-C	CUI	N/A	$1.84	3	$5.51
ICs	BQ51013B (Wireless Charging)	TI	20-VQFN	$3.49	3	$10.47
	CP2105 (UART Bridge)	Silicon Labs	24-QFN	$1.56	3	$4.68
	DRV135 (Audio Balanced Driver)	TI	8-SOIC	$4.42	3	$13.25
	ESP32 (Microcontroller)	Espressif	N/A	$3.80	6	$22.80
	FUSB302B (USB Controller)	ON	14-WQFN	$1.57	3	$4.72
	INA137 (Audio Balanced Receiver)	TI	8-SOIC	$2.65	3	$7.95
	LT3042 (Voltage Regulator)	Analog	10-DFN	$5.00	3	$15.00
	MAX98357A (DAC & Class D Amplifier)	Maxim	16-WQFN	$2.15	6	$12.91
	MCP4451 (Digital Potentiometer)	Analog	20-TSSOP	$1.39	6	$8.34
	MCP73871 (Battery Charger)	Microchip	20-QFN	$1.84	3	$5.52
	MGC3130 (Gesture Control)	Microchip	28-QFN	$4.49	3	$13.47
	PAM8302A (Class D Amplifier)	Diodes	8-SOIC	$0.57	6	$3.42
	SN74LVC541APWR (Bus Driver)	TI	20-TSSOP	$0.41	12	$4.91
	TCK107G (Load Switch)	Toshiba	4-UFBGA	$0.41	32	$13.18
	TPS62802 (Voltage Regulator)	TI	6-DSBGA	$1.67	3	$5.01
	SPW2430 (Analog MEMS Mic)	Knowles	N/A	$0.63	6	$3.77
	SPH0645LM4H (Digital MEMS Mic)	Knowles	N/A	$2.36	9	$21.21
Total						**$399.62**

APPENDIX III: GANTT CHART