PORTABLE HIGH-DEFINITION AUDIO SPECTRUM ANALYZER

Alex Zahn and Jamie Corr

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

Spring 2018

TABLE OF CONTENTS

LISTS OF TABLES AND FIGURES

Table *Page*

Figures

Abstract

The Portable High-definition Audio Spectrum Analyzer (PHASA) allows the user to visualize the audio frequency spectrum of an incoming line-level stereo audio signal. Upon pressing the touch screen spectrum graph, the PHASA displays the corresponding frequency and volume levels as well as crosshairs at the touched location. The PHASA features multiple left/right channel display modes— Left channel only, right channel only, both channels simultaneously, and the average between the two channels. The PHASA features multiple resolution display modes (standard-resolution and high-resolution) and multiple dynamics display modes (standard dynamics, averaging, and peak/hold). The PHASA accepts input audio via a 1/4" TRS jack and outputs via a 1/4" TRS jack. When activated, the output jack outputs a stereo white noise signal for measuring the frequency response of external equipment. This capability makes the PHASA a useful tool for frequency response characterization of audio equipment, which allows for more informed audio engineering design decisions.

Chapter 1 – Introduction

For audio engineers working on equipment such as guitar effect pedals, speakers, or other audio devices, obtaining an accurate frequency response measurement is crucial. Knowing accurately the transfer function of an audio-processing device allows for more informed design decisions, more accurate testing, and ultimately a more successful and higher-fidelity device [7]. Conventionally, engineers use a heavy-duty spectrum analyzer, which does not allow for easy transportation in the field, and which can cost tens of thousands of dollars. For a small startup focused on designing and installing high-quality speaker systems, for example, the weight and cost of a traditional spectrum analyzer make it an unrealistic option. This project report proposes and describes the design of a lightweight, portable, standalone device that can accurately measure the audio spectrum of an incoming signal.

To reliably display the frequency spectrum of an analog stereo audio signal on a digital display, necessary functions include low-passing the input signal to avoid aliasing, sampling to convert it to digital audio, performing fast Fourier transform (FFT) and graphical operations in software, and communicating the graphical FFT data to a digital display. To fulfill these functions in a portable physical package, necessary hardware includes an anti-aliasing filter, analog-to-digital converter, microcontroller, and a digital display [1]. Digital signal processor (DSP) units, while not necessary for all applications, can perform FFT operations significantly faster than general purpose microprocessors due to their FFT-oriented architecture. Additionally, to output analog stereo audio from a digital source, digital-to-analog conversion must occur. Using a microcontroller that integrates many of these functions off-the-shelf could minimize cost and development time.

An accurate frequency response measurement can be obtained using inexpensive and common hardware [3] [4]. A Raspberry Pi microcontroller, for example, can provide fast enough Fourier transform processing for this application; a digital-signal processor with FFT-oriented architecture is not necessary [5]. To measure the actual frequency response (transfer function) of audio gear, white noise can be output from a device and looped back through the equipment under test to provide an even more accurate measurement. Removing the device under test from the loop, such that the white noise output goes straight back into the input of the device, allows for self-calibration— any adverse effects that the measurement hardware has on the accuracy of the measurement, as well as any inaccuracies in the white noise output, can be cancelled out via software to accomplish this [8]. A microcontroller such as the Raspberry Pi can easily output white noise via random number generation for this application [9] [10]. Implementation of a high-order, analog, anti-aliasing filter on the input signal further ensures accuracy, and eliminates the possibility of non-linear aliasing effects [11] [12]. Later chapters of this report outline the design of these system components in further detail.

Chapter 2 – Requirements and Specifications

Customer Needs Assessment

The expected customers are musicians, music producers, sound engineers, acoustic engineers, and consumer audiophiles. To determine the expected customer needs, we spoke with five members of Cal Poly Audio Engineering Society who fit the profile, both current students and alumni working full-time as engineers. We described the project with the same level of detail as in the abstract, and they provided feedback. We found two recurrent opinions: First, the PHASA must have more functionality and features suitable for stereo audio measurements (and preferably more convenience) than existing iOS or Android apps; and second, the PHASA must accurately measure the frequency response of external equipment. Critical features include an easy-to-use user interface, an accurate input signal measurement, and an accurate white noise output. Most potential product users would be willing to pay more for higher quality and accuracy.

Requirements and Specifications

Table I shows the requirements and specifications for the PHASA. We determined the marketing requirements by interviewing people who fit the profile of the expected customer (audiophiles, music producers, audio/acoustic engineers). Interviewees read a short description of the device and reported what features or qualities they deemed necessary for purchase. We used their common opinions and patterns in their responses to form the marketing requirements. To satisfy the marketing requirements, more detailed engineering specifications were created. In developing the engineering specifications, we considered feasibility, cost, and priority.

TABLE I

REQUIREMENTS AND SPECIFICATIONS

Marketing Requirements	Engineering Specifications	Justification
4	The display shows the frequency spectrum of the stereo input signal as a graph, with sound intensity level displayed logarithmically on the y-axis, and frequency displayed logarithmically on the x-axis, including the range of 20 Hz to 20 kHz.	The standard and most useful way to display audio signal characteristics is by a plot of frequency versus amplitude. Human ears can hear at most from 20 Hz to 20 kHz. Most software spectrum analyzers feature a lower volume limit between -80 and -100 dB.
3, 5	The user can switch between all stereo, dynamics, and resolution display modes.	For convenience, the user interface must allow for easy switching between modes.
5	Stereo Display Modes: Left channel only, right channel only, and the average of the two.	The ability to see each channel independently offers useful stereo information.

5	Dynamics Display Modes: Normal (Moderate decay speed), Fast (Faster decay speed), and averaging (Volume averaged continuously).	The ability to see fast-decaying volume levels offers information about the response time of the equipment under test. Averaging allows for a more accurate measurement of response to white noise over time.
5	Resolution Display Modes: Normal (Moderate FFT window length), and High Resolution (Longer FFT window length).	High-resolution spectrum allows the user to see the location of resonant peaks more accurately.
3	Input-to-display latency does not exceed 50ms	Noticeable latency disrupts work flow.
1	The PHASA weighs no more than 2 pounds.	Weighing less than an average laptop gives the PHASA the advantage of greater portability.
2	The PHASA requires at least 5V DC, 2.5A input power.	Standard microcontrollers and peripherals operate on a 5V supply.
2	From full-charge until depletion, the battery life of the PHASA meets or exceeds 5 hours.	Users require a device that doesn't die on them during normal use.
4	The PHASA samples both input audio channels at at least 44.1 kHz.	The Nyquist rate for 20 kHz audio is 40 kHz. Standard sample rate is 44.1kHz for HD audio.
4	The PHASA samples both input audio channels with at least 16 bit resolution.	Any lower than this resolution results in poor audio quality.
4	The PHASA measures input voltages with AC peak amplitude above ±0.2 volts and below ±2 volts.	Line level audio generally ranges from 0.2 to 2 volts. The PHASA does not accept unattenuated speaker-level outputs (generally on the scale of several tens of volts).
4	The PHASA filters out all input signal content above 25 kHz with at least -10 dB attenuation.	Small amounts of signal content above the Nyquist limit can cause nonlinear aliasing effects, rendering the measurement useless.
4	The filter passes signal content below 20 kHz at 0 dB, plus or minus 0.5 dB.	Measurement must be accurate. The frequency response of the hardware must be flat.
6	The audio input impedance of the PHASA meets or exceeds 5 kΩ.	Standard line-level input impedances average around 10 kΩ. For high fidelity, we use impedance bridging techniques.
6	The audio output impedance of the PHASA does not exceed 200 Ω.	Standard line-level output impedances average around 100 Ω.
3	The PHASA accepts user-input via touch screen.	Touch screen capability allows for convenient user input without the need for external input hardware such as a keyboard, mouse, or buttons. External input hardware adds complexity and more potential points of failure.

Marketing Requirements

1. Portable and lightweight.
2. Battery-powered.
3. Easy to operate.
4. Accurately displays the frequency spectrum of an incoming stereo audio signal.
5. Multiple stereo, dynamics, and resolution display modes.
6. Interfaces with standard analog audio hardware.

Table II shows dates for the various deliverables during project development, from Fall 2017 through Spring 2018.

TABLE II
PROJECT DELIVERABLES

Delivery Date	Deliverable Description
2/27/18	EE 461 demo
3/16/18	EE 461 report
5/11/18	System interfaced with display and successfully reading input audio
5/18/18	EE 462 demo
5/25/18	ABET Sr. Project Analysis
6/1/18	Sr. Project Expo Poster
6/8/18	EE 462 Report

Chapter 3 - Functional Decomposition

Figure I shows a Level 0 block diagram of the PHASA which shows all inputs and outputs at the highest level of abstraction. At the top level, the PHASA accepts analog line-level stereo audio and displays it on the frequency spectrum through the output to the display. It also accepts power and power switch inputs, user control input from the touch screen, and outputs analog line-level stereo audio for the white noise function. Table III lists the details of each input and output as well as a description of the functionality of the PHASA overall.

FIGURE I
LEVEL 0 BLOCK DIAGRAM

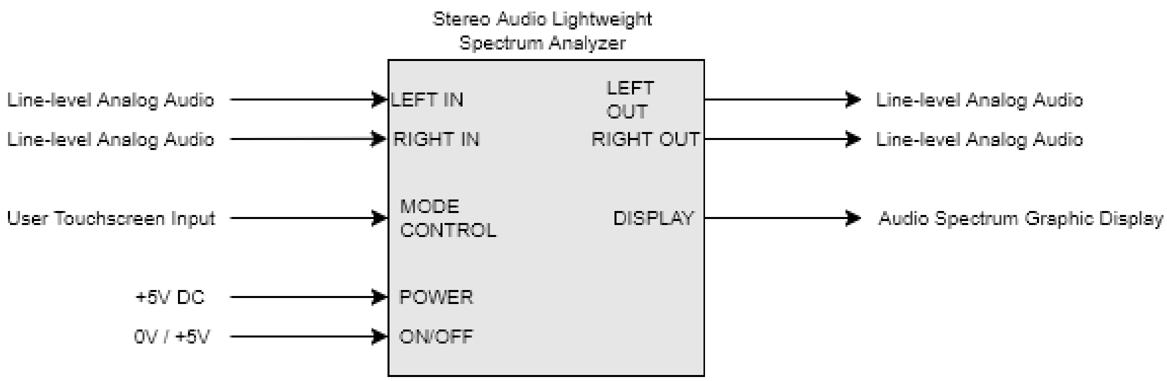


TABLE III
LEVEL 0 FUNCTIONALITY TABLE

Module	Portable High-definition Audio Spectrum Analyzer (PHASA)
Inputs	• Left Channel In: Line level electrical audio signal (0.2-2V) • Right Channel In: Line level electrical audio signal (0.2-2V) • Mode Control: User input via touch screen for operating mode selection • Power: +5 volts DC, max 1000mA • On/Off: 0 volts DC or +5 volts DC (momentary switch, 0V open, +5V closed, normally open)
Outputs	• Left Channel Out: Line level electrical audio signal (0.2-2V) • Right Channel Out: Line level electrical audio signal (0.2-2V) • LCD: Visual information displayed on an LCD
Functionality	The PHASA accepts a stereo line-level audio signal via Left/Right Channel In and displays its frequency spectrum on an LCD. +5 volts DC powers the PHASA, and a temporary +5Vat the On/Off input turns the PHASA on or off. The PHASA accepts Mode Control input from the user to switch between operating modes (stereo, dynamics, and resolution display modes). Left/Right Channel Out outputs a line level stereo audio signal.

Figure II shows a more detailed Level 1 block diagram of the PHASA. This lower-level diagram shows the individual functional components that comprise the system, namely the analog anti-aliasing filter, the analog-to-digital converter, the microcontroller, and the touch screen display. The arrows between blocks represent signal flow. Each arrow (or set of stereo arrows) is labeled with its connection type. Tables IV-VII list the input and output signal details for each of the four functional blocks, as well as a description of each block's function.

FIGURE II
LEVEL 1 BLOCK DIAGRAM

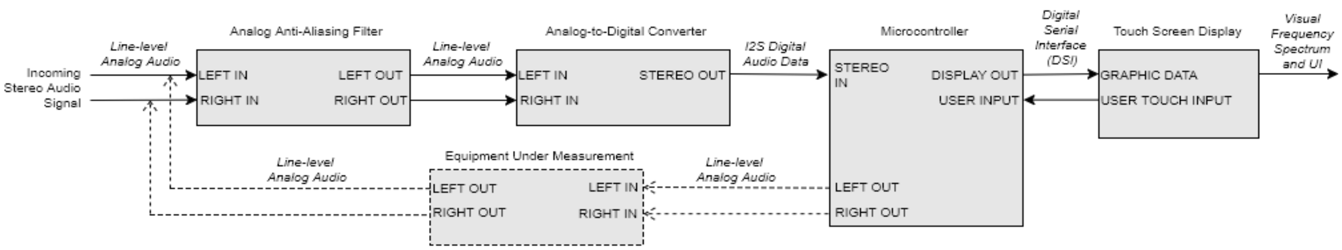


TABLE IV
LEVEL 1 BLOCK DIAGRAM FUNCTIONALITY TABLE: ANTI-ALIASING FILTER

Module	Analog Anti-Aliasing Filter
Inputs	• Left In: Line level analog audio signal (0.2-2V) • Right In: Line level analog audio signal (0.2-2V)
Outputs	• Left Out: Line level analog audio signal (0.2-2V) – Low-passed • Right Out: Line level analog audio signal (0.2-2V) – Low-passed
Functionality	Filters out any signal content higher in frequency than the audio band to prevent nonlinear aliasing effects during analog-to-digital conversion.

TABLE V
LEVEL 1 BLOCK DIAGRAM FUNCTIONALITY TABLE: ANALOG-DIGITAL CONVERTER

Module	Analog-to-Digital Converter
Inputs	• Left In: Line level analog audio signal (0.2-2V) – Low-passed • Right In: Line level analog audio signal (0.2-2V) – Low-passed
Outputs	• Stereo Out: Digital stereo audio, serial interface (I2S)
Functionality	Samples the filtered analog stereo audio at a high sample rate and communicates the information to the microcontroller via an I2S serial interface to a GPIO pin [2].

TABLE VI
LEVEL 1 BLOCK DIAGRAM FUNCTIONALITY TABLE: MICROCONTROLLER

Module	Microcontroller
Inputs	• Stereo Out: Digital stereo audio, serial interface (I2S) • User Input: Touch screen user input data via digital serial interface (DSI)
Outputs	• Display Out: Graphics data to display via digital serial interface (DSI)
Functionality	Receives digital audio, performs FFT, processes graphics for display, and handles user input to change display modes. Outputs analog white noise for self-calibration and equipment measurement.

TABLE VII

LEVEL 1 BLOCK DIAGRAM FUNCTIONALITY TABLE: TOUCH SCREEN DISPLAY

Module	Touch Screen Display
Inputs	• Graphic Data: Graphics data from microcontroller to display via digital serial interface (DSI)
Outputs	• User Touch Input: Touch screen user input data to microcontroller via digital serial interface (DSI)
Functionality	Displays the frequency spectrum of the stereo audio processed by the microcontroller. Displays the graphical user interface and transmits user input data to the microcontroller.

Chapter 4 - Project Planning

Figure III shows the project development schedule as a Gantt chart. The top section shows the schedule for EE460, the middle for EE461, and the bottom for EE462. This chart shows hardware and system integration tasks, not graphic user interface development tasks, which a software engineering student will complete as a separate project.

FIGURE III

STEREO AUDIO LIGHTWEIGHT SPECTRUM ANALYZER DEVELOPMENT SCHEDULE

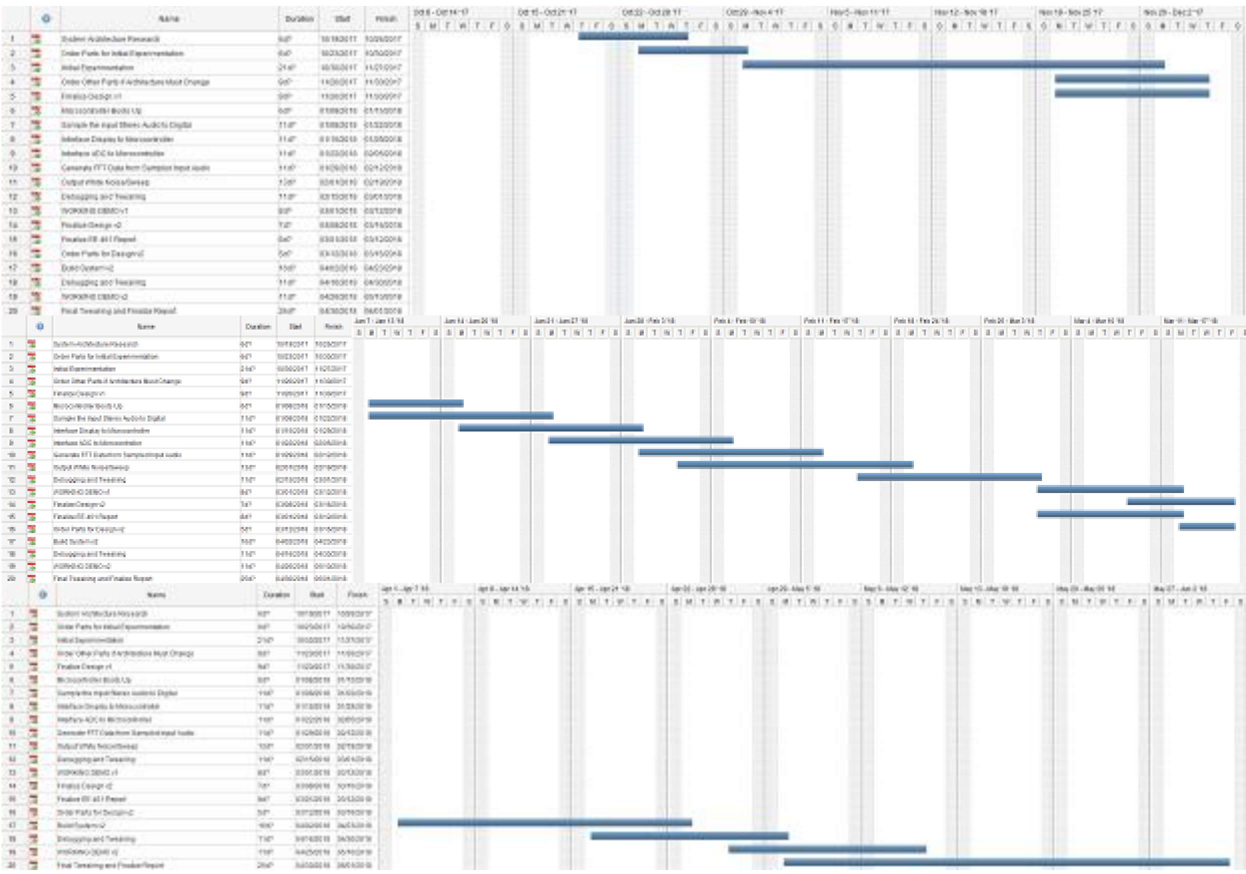
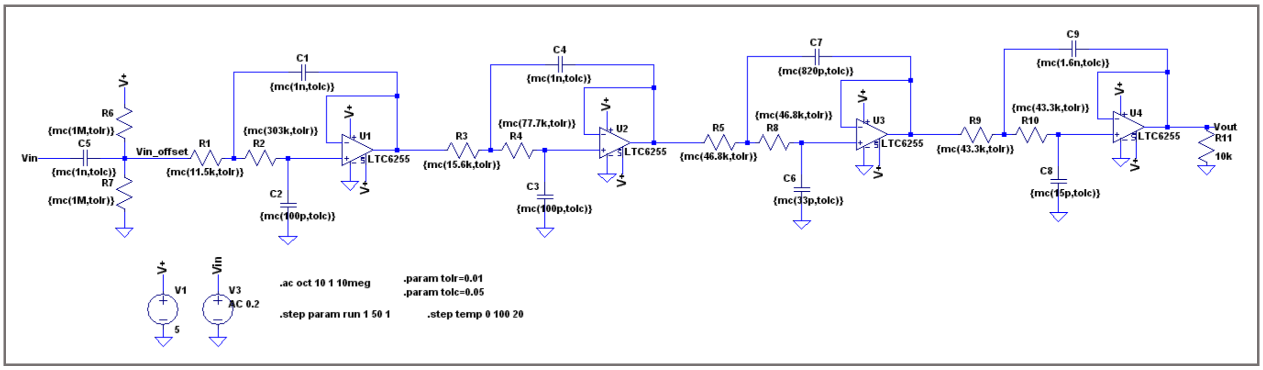


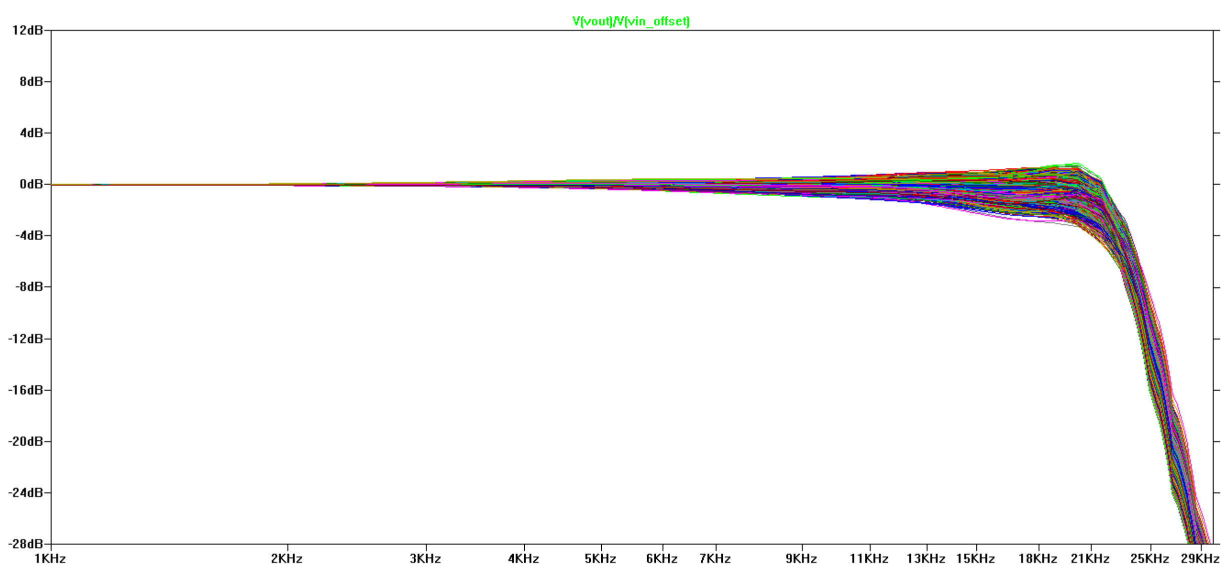
Table VIII shows a cost estimate for parts and labor. We calculated labor costs assuming two people working at a pay rate of $18/hour, for 5 hours per week, for 20 weeks. The estimated cost of parts and labor sums up to about $3767. The estimated cost excluding labor sums up to about $167. In future design iterations, PCB fabrication costs would replace that of the mini breadboard.

The anti-aliasing filter requires a very narrow transition band to meet the engineering specifications. An 8-pole Butterworth low-pass filter using a Sallen-Key topology — implemented using the LTC6257 Quad Op Amp, and designed using Analog Device's filter wizard tool — has proven sufficient in simulations [13]. Figure IV (a) shows the schematic as drawn in LTSPICE, using 1% resistor tolerances and 5% capacitor tolerances. Analog Devices provides the LTC6257 quad op amp simulation model online. Running a Monte Carlo AC analysis, which also takes temperature variations into account, generated the magnitude response shown in Figure IV (b). The filter passes frequencies in most of the passband almost perfectly flat, and exhibits no more than 2.5dB of potential inaccuracies near 20 kHz. The transition width from the passband to the stopband appears steep enough to meet the -10dB at 25kHz specification.

FIGURE IV
ANTI-ALIASING FILTER DESIGN



(a) The filter circuit schematic as drawn in LTSPICE.



(b) The theoretical filter magnitude response, generated using Monte Carlo analysis in LTSPICE.

TABLE VIII

PARTS AND LABOR COST ESTIMATE

Part	Cost ($)
Anti-Aliasing Filter Passive Components	40.00
LTC6257 Quad Op Amp	5.00
PCM1803A A/D Converter	3.76
SSOP20-to-DIP adapter	4.79
Mini-breadboard	1.15
Raspberry Pi	35.00
Touchscreen	70.00
Power & Audio Jacks, Hardware	6.00
Battery Connector	1.00
Labor	3600.00
Total	3766.70

Chapter 5 - Development

Power

To fulfill the requirement that the PHASA be portable, we chose to use battery power. We chose to use common AA batteries because they're easily accessible to users and didn't require much development time. We purchased from Adafruit an off-the-shelf 3xAA battery holder with a built-in on-off switch, as well as the "PowerBoost 1000 Basic" 5V USB DC-DC Boost Converter. Together, three AA batteries provide a nominal 4.5VDC voltage, which the DC-DC converter boosts to 5VDC. The converter then outputs 5VDC through USB to the RaspberryPi microcontroller, which distributes power to the other components via its 5VDC and 3.3VDC rails. Because we didn't achieve a fully operational prototype, we didn't have the opportunity to test the lifetime of the device. However, because three fully-charged alkaline AA batteries typically provide 7500 mAh of charge, the device should theoretically achieve the five-hour battery lifetime requirement with a maximum current draw of 1500 mA.

Anti-Aliasing Filter

We constructed the anti-aliasing filter on a breadboard following the schematic shown in Figure IV, with the exception that an LTC6257 Quad Op-Amp IC was used instead of an LTC6255. We powered the LTC6257 with 5VDC as suggested by the manufacturer datasheet [15]. It didn't require any other peripheral passive components to operate, so once we had connected 5V and ground, we simply used the four op-amps on the IC for the four stages of the filter as shown in Figure IV. The input, labeled as "Vin" in the schematic, should come from either a 1/8" or 1/4" TRS jack as per the device specifications. For testing purposes, we used lab equipment-generated line-level signals by connecting grabber cables to the front lead of capacitor C5. However, because we used a switched 1/8" TRS jack and connected the closed switch terminals to ground, we made sure to leave an 1/8" TRS plug in the jack so that the switches were lifted. Otherwise, the input would be pulled to ground. The filter circuit works for a single channel of audio, so to support stereo audio input, the left and right channels of audio each require their own anti-aliasing filter circuit. We built a single filter circuit for testing, then built the second afterwards.

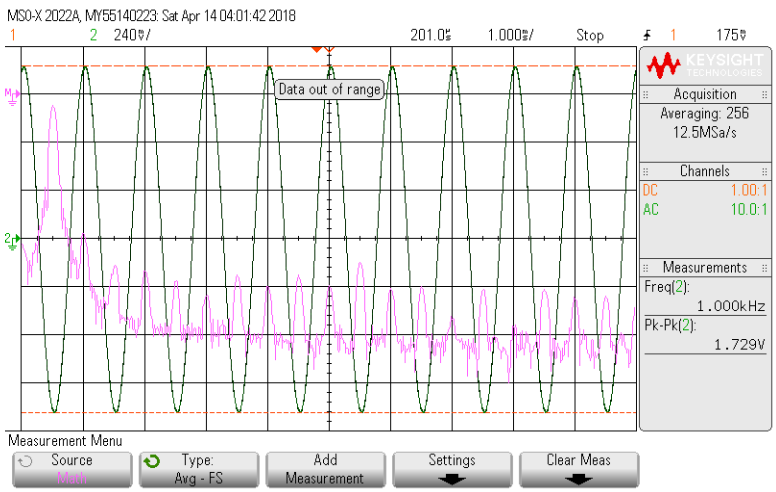
To test the filter performance, we measured both the total harmonic distortion (THD) and the frequency response using an Agilent MSO-X 3014A oscilloscope and a Hewlett-Packard 3582A spectrum analyzer, respectively.

Total Harmonic Distortion

To measure the filter's THD, we inputted a pure 1kHz, 2V peak-to-peak sine wave into the filter using the oscilloscope's built-in sine wave generator. We captured the filter output with the oscilloscope, displayed it using the FFT operator, and measured the root-mean-square (RMS) voltage of the harmonic peaks. Figure V shows a screen capture of the oscilloscope measurement.

FIGURE V

ANTI-ALIASING FILTER MEASUREMENTS: TOTAL HARMONIC DISTORTION



The green trace represents the time-domain output of the anti-aliasing filter, and the pink trace represents the frequency-domain output of the anti-aliasing filter with FFT applied. The oscilloscope measures voltage in FFT mode in units of dBv, but the following equation allows for conversion from dBv to RMS voltage (for a sine wave):

$$v_{RMS} = \frac{10^{(v_{dBv}/20)}}{\sqrt{2}}$$

We measured the fundamental and next five harmonics, up to n=6. We disregarded further harmonics as they have a negligible effect on the THD. Table IX shows the measured values, including their RMS conversions.

TABLE IX
HARMONIC VOLTAGE MEASUREMENTS

Harmonic Order (n)	Voltage (dBv)	Voltage (RMS)
1	-4.375	0.42730
2	-58.75	0.00082
3	-71.25	0.00019
4	-73.75	0.00015
5	-77.50	0.00009
6	-83.75	0.00005

We then calculated the THD using the following equation:

$$\mathrm{THD}(\%) = 100 * \frac{\sqrt{v_2^2 + v_3^2 + v_4^2 + \ldots}}{v_1}$$

where v_n represents the RMS voltage of the nth harmonic peak. v_1 represents the fundamental. Using this equation and the RMS values shown in the table above, we calculated the THD of the filter as only 0.20% at 1kHz.

Frequency Response

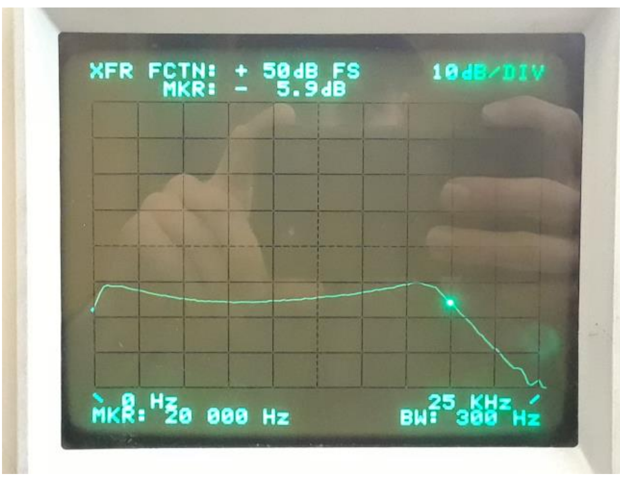
According to the simulations shown in Figure IV, the filter should theoretically have a flat frequency response up until 20kHz, the upper boundary of the audible range, at which point it should steeply drop off. To measure the actual frequency response of the filter, we used the white noise output from the 3282A spectrum analyzer. We used a BNC-T splitter to split the spectrum analyzer's white noise output in two, then connected the first branch to the spectrum analyzer's "B" input, and connected the second branch to the anti-aliasing filter input. We then connected the anti-aliasing filter output to the spectrum analyzer's "A" input, and set the display mode to "A/B". This ensured that the display showed the difference between the "A" and "B" inputs— in other words, the actual transfer function of the anti-aliasing filter. Figure VI shows some of the key measurements.

Our measurements showed that, despite having the correct overall shape of a high-order low-pass filter, the magnitude response was not ideally flat in the audible band. We observed a 6.0 dB dip, centered at 8 kHz. The resonant peaks appeared at 840 Hz and 18.4 kHz with a magnitude of -0.5 dB. The filter rolled off to -6.0 dB (half power) at 20 kHz as desired. However, we observed an unintentional high-pass effect that rolled off to -6.0 dB at 188 Hz, well above the lower limit of the audible range at 20 Hz. The filter cut frequencies below 188 Hz even more, so in its current state, this filter would be inappropriate for audio content in the lower bass and sub-bass frequency ranges. The discrepancies between the measured response and the theoretical response were most likely a result of non-nominal passive component values in the circuit. Each stage of the filter would magnify any error in resistor or capacitor values from the previous stages.

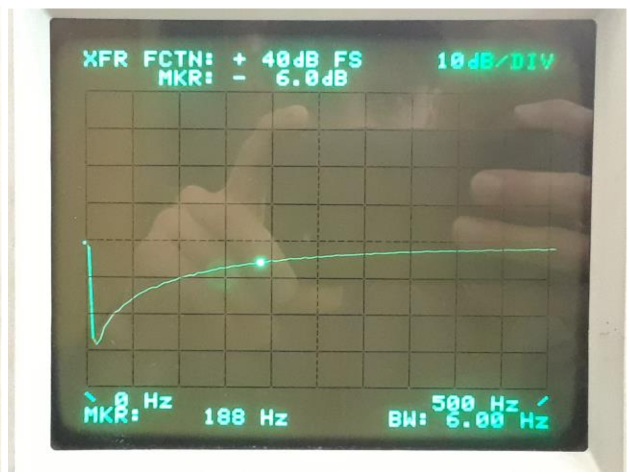
Future development should aim to accomplish the following: Eliminate the high-pass effect, or at least shift its corner frequency (-6.0dB) down to 20 Hz; and account for the dip in the passband, either through tweaking the filter components, appending another filter stage to compensate for the error, or hard-coding compensation into the PHASA software (although the error in the filter response could easily vary between builds).

FIGURE VI

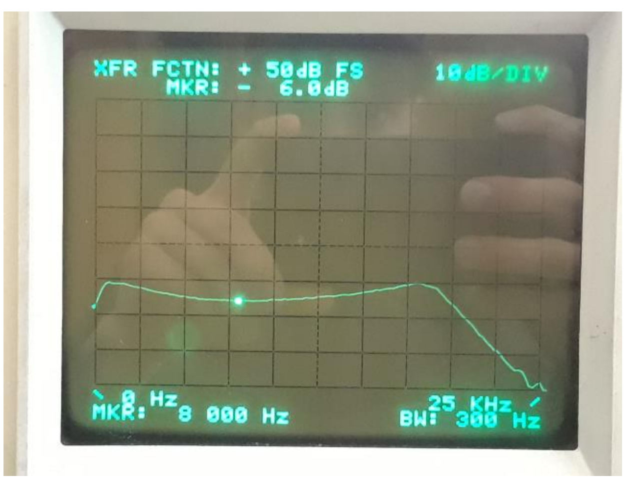
ANTI-ALIASING FILTER MEASUREMENTS: FREQUENCY RESPONSE



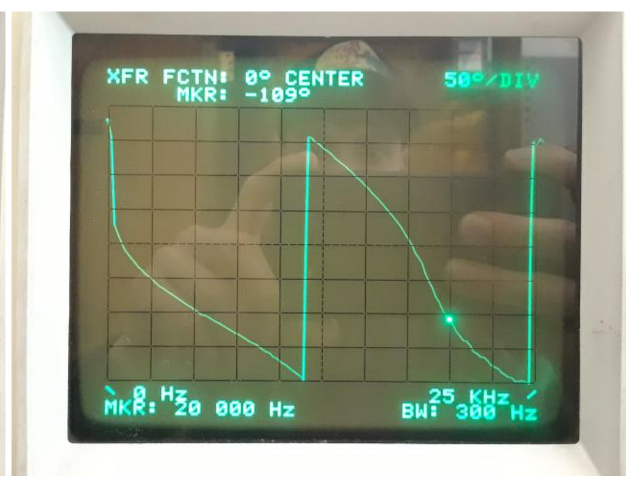
(a) Magnitude response. -5.9 dB at 20 kHz.



(b) Magnitude response. -6.0 dB at 188 Hz.



(c) Magnitude response. -6.0 dB at 8 kHz.

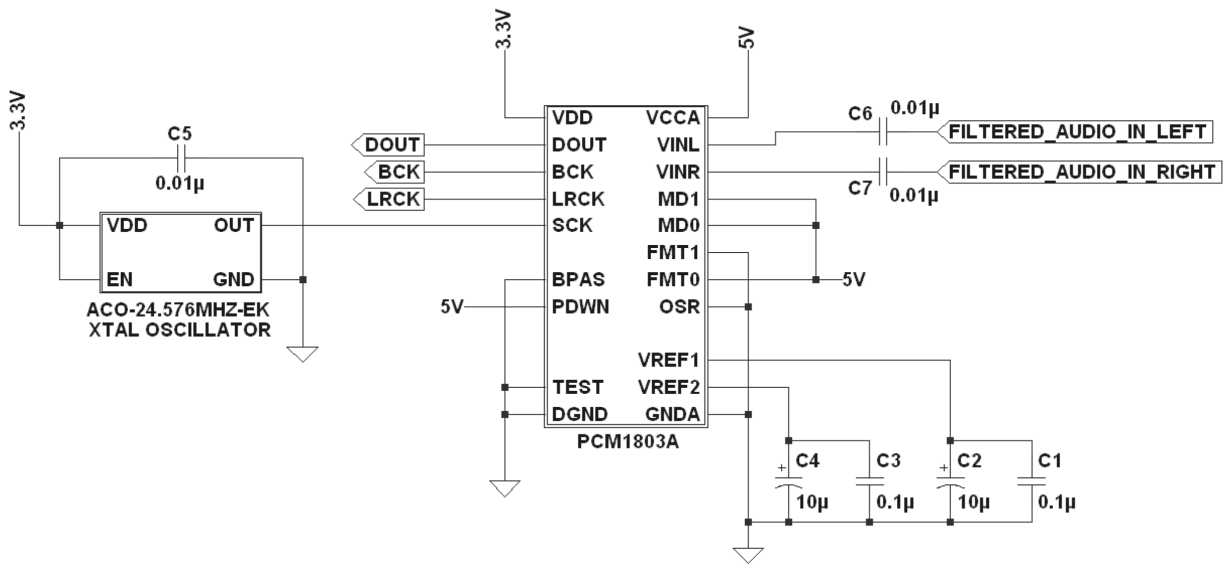


(d) Phase response.

Analog-to-Digital Converter

For the analog-to-digital conversion (ADC) stage, we chose to use the Texas Instruments PCM1803A Stereo A/D Converter, because it supports stereo high-definition audio (24-bit resolution and 96 kHz sample rate) and the I2S serial bus interface. I2S is a digital interface intended for stereo audio that uses three lines: a bit clock (BCK), a word clock (LRCK), and a data line (DOUT). We built the ADC subcircuit on a breadboard, using peripheral passive components and mode settings as suggested by the manufacturer datasheet, Peter Onion's tutorial, and Sparkfun.com's breakout board schematic [2] [3] [14]. Figure VII shows the schematic for the complete ADC stage.

FIGURE VII
ANALOG-TO-DIGITAL CONVERTER SCHEMATIC



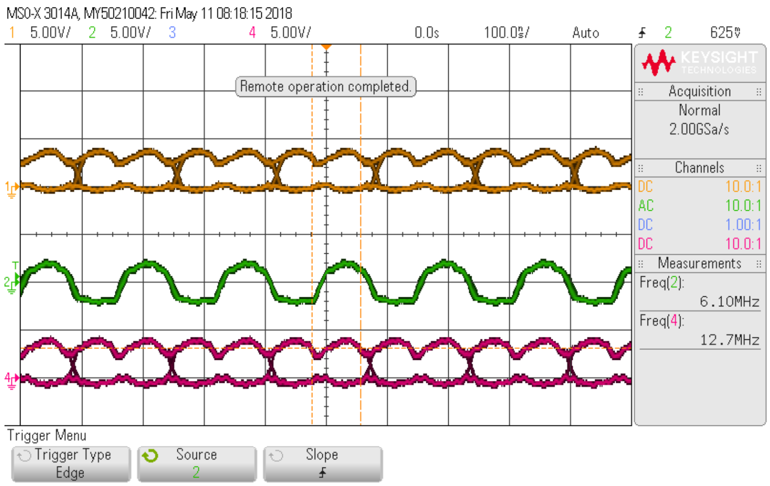
The I2S lines go to the Raspberry Pi (DOUT, BCK, and LRCK). The audio inputs come from the outputs of the anti-aliasing filter stage. For testing, any line-level analog audio signal can be used via grabber cables, an 1/8" TRS jack, or a 1/4" TRS jack. VDD and VCCA can come from either the Raspberry Pi's regulated rails, or for testing, a dual power supply.

To test the ADC operation, we powered it with 5VDC and 3.3VDC using an Agilent E3648A dual power supply; connected a stereo analog audio signal to the PCM1803A left and right inputs using a computer audio out, 1/8" TRS cable, and 1/8" stereo jack; and probed the three I2S lines with an oscilloscope. Figure VIII shows oscilloscope screen captures of the three lines. We observed the ADC successfully acting as the I2S master device, generating a 6 MHz bit clock and a 96 kHz word clock, and transmitting 24-bit digital audio samples on the data line. A 96 kHz word clock frequency implies a sampling frequency of 48 kHz per channel, which fulfills the PHASA sampling frequency engineering specification of at least 44.1 kHz per channel. Because full-range audio has a Nyquist rate of 40 kHz per channel, the ADC does not cause any measurable aliasing. Therefore, the digital audio output by the ADC is high-definition and not measurably distorted.

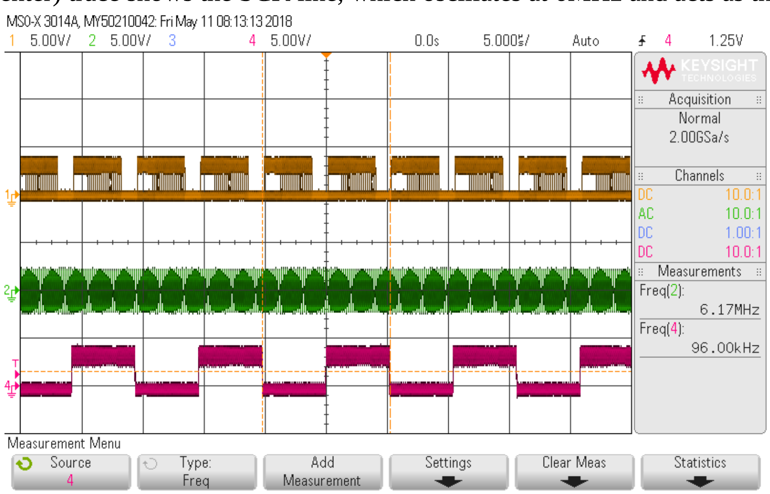
The digital logic signals as measured by the oscilloscope do not appear as perfectly rectangular signals. However, this is a result of the internal capacitance of the oscilloscope probes and wires, not imperfections of the ADC itself. The high frequencies of the digital signals, along with the internal capacitance of the probes and wires, caused a longer time constant which smoothed out the logic transitions, making the signals appear less rectangular than they truly are.

FIGURE VIII

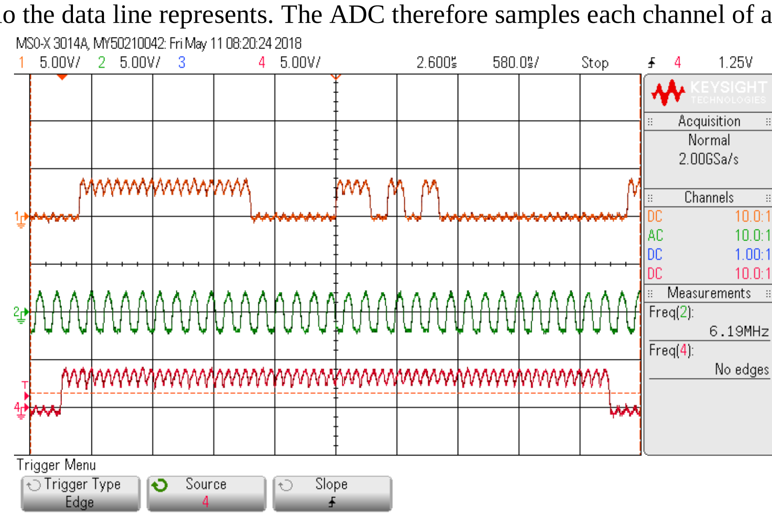
ANALOG-TO-DIGITAL CONVERTER MEASUREMENTS: I2S LINES



(a) The green (center) trace shows the SCK line, which oscillates at 6MHz and acts as the master clock signal.



(b) The pink (bottom) trace shows the LRCK line, which oscillates at 96kHz to signify which channel of stereo audio the data line represents. The ADC therefore samples each channel of audio at 48kHz.



(c) The yellow (top) trace shows the DOUT line, which transmits 24 bits to represent the analog audio voltage.

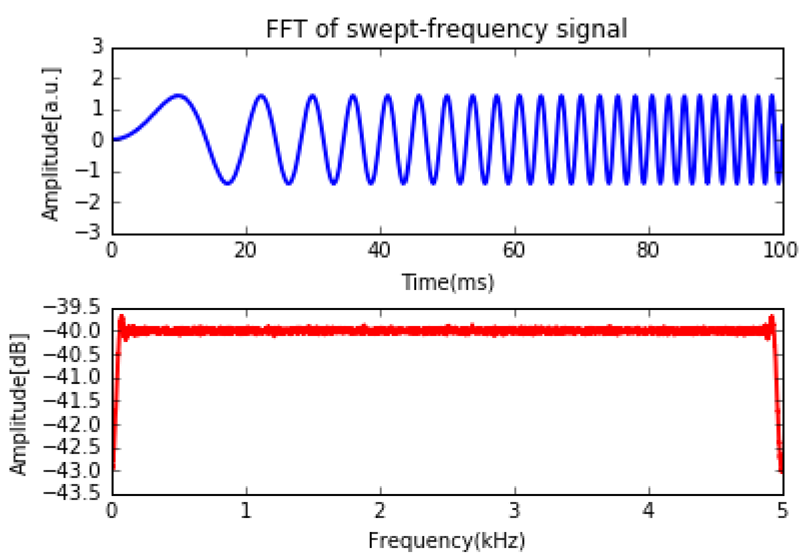
Microcontroller

The microcontroller acts as the heart of the PHASA— it performs the necessary FFT and graphics processing to display the frequency content of the input audio, and it handles white noise generation, which is crucial for measuring the transfer function of external equipment. To minimize development time, we chose to use a Raspberry Pi 3 Model B microcontroller running Raspbian (a Linux-based operating system for the Raspberry Pi). This allowed us to avoid having to write graphics and touch-input drivers, as the "plug-and-play" Raspberry Pi 7" Touchscreen Display simply connects to the Raspberry Pi's DSI port and sets up in minutes.

FFT

When running the FFT code with libraries including python numpy and matplotlib, package updates were required. However, there was a conflicting package version with the preinstalled packages on the Raspbian Stretch operating system. Since this may have been caused by previous usage of the RaspberryPi prior to work on this project, we reinstalled Raspbian. After reformatting the memory card and flashing the operating system, limited memory was still an issue, so we had to delete several preinstalled programs to update the necessary Python libraries. Figure IX shows the successful FFT script output from a sine sweep on the Pi, generated after we resolved those issues. Future development should focus to alter the FFT function to take a continuous stream of audio rather than an audio file.

FIGURE IX
FFT PYTHON SCRIPT OUTPUT



I2S

To receive digital audio from the ADC via I2S, we had to configure the Raspberry Pi to detect the ADC as an ALSA recording device. User "plugh" on the RaspberryPi.org forum provided[1] instructions to allow the system to load custom kernel modules, as well as code to build an ALSA "simple-card" driver. Unfortunately, building the driver became a major obstacle for us, and we were unable to interface the ADC to the Raspberry Pi. We followed an edited putty log[2] of the procedure to setup system to allow building loadable kernel modules, then to build the simple-card and DMIC drivers, and finally to build[3] the hello_world example and my_loader.

Loadable kernel module

Communication from the ADC to the RaspberryPi requires a simple I2S interface that does not depend on a particular DAC or CODEC device that requires configuration over I2C or SPI. To avoid recompilation of the entire Linux kernel, a loadable kernel module can be used to create such an interface. This enables the driver for the I2S peripheral inside the BCM2708 device to be loaded dynamically[4]. Our development of the PHASA could not progress without the ability to record a stream of audio. The rpi-source file from the wiki (required to build the module) differed slightly from the forum post log output, which became an issue when the log editor prompted a patch[5] to be written in the source file. Consequently, there was an unclear complier error when building the hello_world example. Figure X shows screenshots of the errors we received when attempting to build the modules. An alternative approach would be to run a barebones operating system and simply write a custom I2S driver using three GPIO pins for the interface. However, that approach would also require writing custom drivers for the graphics and system operation, which would require an unnecessary and unrealistic amount of development time. Therefore, future development should focus on continuing the ALSA simple-card driver setup instructions where we left off.

---

[1] https://www.raspberrypi.org/forums/viewtopic.php?f=44&t=8496&start=575#p604231

[2] https://github.com/notro/rpi-source/wiki

[3] https://github.com/notro/rpi-source/wiki/Examples-on-how-to-build-various-modules

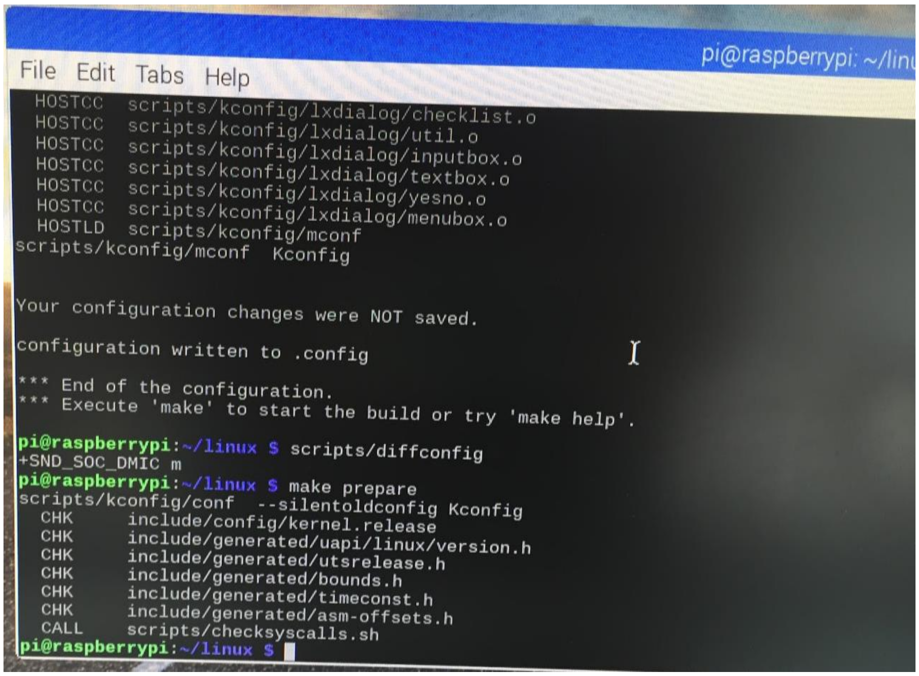
[4] https://www.raspberrypi.org/forums/viewtopic.php?t=91237&start=50

[5] https://github.com/raspberrypi/linux/commit/e244bb9bc1883547d44642c99f483c2e57e2a940

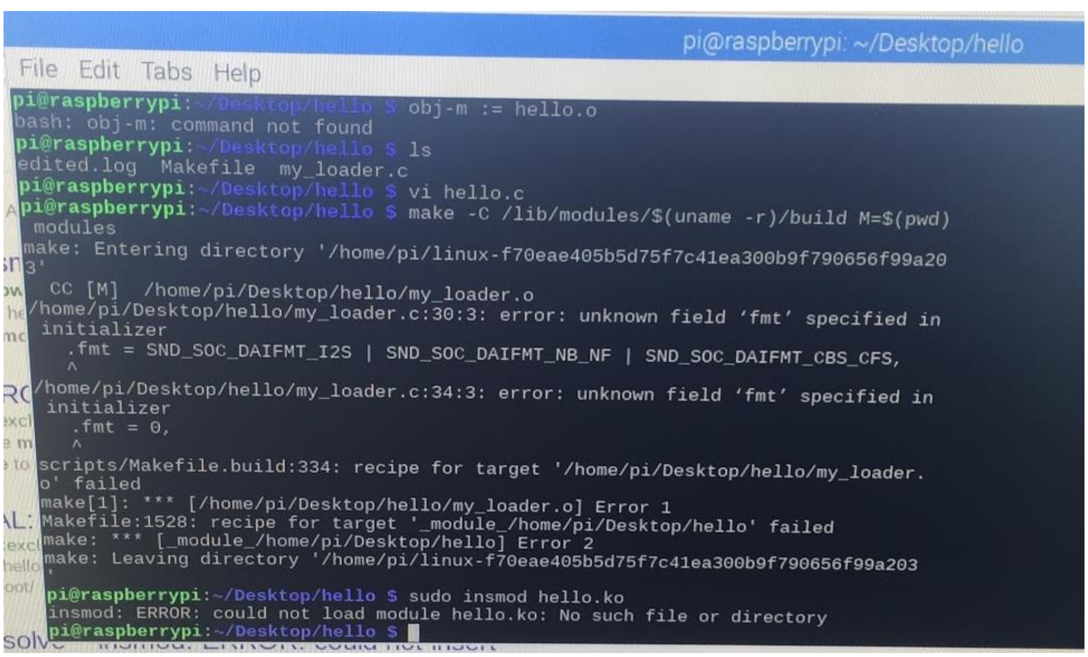
Once the “unknown symbols” error is resolved and the simple-card driver is successfully installed on the RaspberryPi, development of the PHASA audio processing and GUI software can begin.

FIGURE X

ALSA I2S AUDIO DRIVER SETUP ERRORS



(a) RaspberryPi terminal output when loading the kernel module.



(b) Error building the kernel driver from the script provided on the RaspberryPi Forum.

Chapter 6 – Conclusion

Consumer audiophiles, music producers, audio equipment developers, and audio technicians desire a portable device that allows for quick and easy audio spectrum analysis. The PHASA fulfills that desire by performing accurate audio spectrum analysis, and offering an easy-to-use touch screen display with multiple display modes. Its battery power and standard 1/4" or 1/8” TRS input allows for reliable measurement of line-level, stereo, analog audio. The PHASA’s white noise output with self-calibration allows for quick and accurate characterization audio equipment frequency response. The PHASA in this sense acts as a useful tool for the development of high-definition, true-to-sound audio equipment.

During this senior project, we successfully developed an analog anti-aliasing filter that prevents nonlinear aliasing effects at the digital sampling stage of the input chain. We successfully sampled filtered audio and converted it to a digital signal to be transmitted via I2S. We set up a microcontroller platform to act as the brain of the PHASA which handles the software processes— the frequency spectrum analysis, white noise output and self-calibration, graphics, and user input. However, while we did interface a touch screen with the microcontroller to accept user input and display digital audio on the frequency spectrum, we were unable to install the necessary drivers to receive I2S audio from the analog-to-digital converter. Because of the inability to receive external audio, we could not further develop the PHASA software to perform the desired functions. We did not achieve the capability to display the frequency spectrum of audio from the input, nor to characterize the frequency response of external audio equipment with white noise. Both of those functions require a continuous stream of input audio.

Future development should first and foremost seek to resolve the errors we received in attempting to install the I2S drivers. Once I2S audio can be received by the microcontroller, software development can continue and the defining capabilities of the PHASA can be achieved. To ensure the most accurate audio measurement possible, future development should also aim to improve the frequency response of the anti-aliasing filter as discussed in Chapter 5. Overall, we have completed initial development of the core hardware components of the PHASA. With the help of future software development, the PHASA can serve as a valuable tool for the audio, live sound, and music industry.

Works Cited

[1] Lin, Tseng, Cox, et al. "A monolithic audio spectrum analyzer". IEEE Transactions on Acoustics, Speech, and Signal Processing. Volume: 31, Issue: 1, Pages: 288-293. Feb 1983.

[2] "PCM1803A Single-Ended, Analog-Input 24-Bit, 96-kHz Stereo A/D Converter". Texas Instruments. Datasheet. 2016. <http://www.ti.com/lit/ds/symlink/pcm1803a.pdf>

[3] Onion, Peter. "Using a PCM1803A I2S Codec with a Raspberry Pi". PeterOnion.org.uk. 2014.

[4] Inns, Simon. "Real-Time Audio Spectrum Analyser". WaitingForFriday.com. 2011.

[5] Bombard, Thomas J. "Audio spectrum analyzer implemented with a minimum number of multiply operations". United States Patent #6,760,674. July 6, 2004.

[6] Miller, et al. "Automatic equalizer". United States Patent #5,506,910. April 9, 1996.

[7] Brixen, Eddy. "Audio Metering: Measurements, Standards and Practice". Burlington, Elsevier Science. 2010.

[8] Stanley, Andrew. "Multi-platform audio spectrum analyzer incorporating fast Fourier transform and exponential forgetting factor methods". Microfiche. 2001.

[9] Zukerman, L. G. "Application of a Spectrum Analyzer for Use with Random Functions". IRE Transactions on Instrumentation. Volume: I-10, Issue: 1, Pages: 37-43. June 1961.

[10] French, A. S. "Coherence Improvement in White Noise Analysis by the Use of a Repeated Random Sequence Generator". IEEE Transactions on Biomedical Engineering. Volume: BME-27, Issue: 1, Pages: 51-53. Jan 1980.

[11] Baker, Bonnie C. “Designing an anti-aliasing filter for ADCs in the frequency domain”. Texas Instruments Analog Applications Journal. 2015.

[12] Baker, Bonnie C. “Tutorial: Basics of choosing and designing the best filter for an effective data-acquisition system”. Planet Analog Magazine. Aug 2006.

[13] Analog Devices. “Analog Filter Wizard”. Analog.com. Nov 2017.

[14] Weiss, Aaron. “PCM1803A_Breakout-v11”. Schematic. Sparkfun.com. 9 March 2015.

[15] "LTC6255/LTC6256/LTC6257 6.5MHz, 65µA Power Efficient Rail-to-Rail I/O Op Amps". Revision D. Linear Technology. Datasheet. 2010.

APPENDIX A

STEREO AUDIO LIGHTWEIGHT SPECTRUM ANALYZER ABET SENIOR PROJECT ANALYSIS

Project Title: Stereo Audio Lightweight Spectrum Analyzer
Student's Name: Alex Zahn **Student's Signature**:
Advisor's Name: Brian James Mealy **Advisor's Initials**: **Date**:
Summary of Functional Requirements:
This portable spectrum-analyzer device allows the user to visualize the audio frequency spectrum of an incoming stereo audio signal. Upon pressing the touch screen spectrum graph, the device displays the corresponding frequency and volume levels as well as crosshairs at the touched location. The device features multiple left/right channel display modes— Left channel only, right channel only, both channels simultaneously, and the average between the two channels. The device features multiple resolution display modes (standard-resolution and high-resolution) and multiple dynamics display modes (standard dynamics, averaging, and peak/hold). When activated, the output jack outputs a stereo white noise signal for measuring the frequency response of external equipment. The device accepts input audio via either 1/4" or 1/8" TRS jacks and outputs via a 1/4" TRS jack.
Primary Constraints:
Determining what system architecture to use and choosing an appropriate microcontroller was a major challenge for this project. We needed a controller with the capability to both input and output stereo audio, perform fast Fourier transform operations and graphics code, and drive an LCD. The controller needed to perform all those functions at a rate fast enough for high-fidelity audio. Another limiting factor was the portability requirement. The device needed to be portable and lightweight, which meant using power from a standard wall outlet was impossible and components for the system had to be chosen with overall weight and size in mind. (See Chapter 2: Requirements and Specifications).
Economic:
This product has an impact on the economy because, for each one produced, the companies who source the microcontroller, LCD, and other parts earn money. Purchasing parts from them allows them to pay their employees higher wages or to reinvest what they've earned and expand their company. This product has a positive impact on the electrical and mechanical manufacturing industries. It draws capital into the audio industry, because its intended purpose is to measure high-fidelity audio equipment. It also impacts the Earth's natural resources in that silicon, copper, brass, nickel, bronze, tin, gold, and plastics are used for the embedded electronics, enclosure, and audio connectors. The people and industries who process and source those raw materials benefit from this product.
This portable spectrum analyzer costs the cost of all parts per product plus the cost of labor during development. Estimating two persons working at a pay rate of $18/hour, for 5 hours per week, for 20 weeks, labor itself costs minimum $3,600. A Gantt chart of the estimated development schedule is shown in Figure 2. We estimated the cost of parts per unit as $125, without buying parts in bulk. In addition, development required the use of an electrical multimeter, an oscilloscope, and electrical power—all of which have their own costs.
Assuming this product sells for $400, each unit sold would profit about $275 after subtracting the estimated cost of parts. The cost of development labor then would be covered by the profits of 13 units sold. Using the parts cost estimate above, the cost of parts makes up about 30% of the $400 selling price estimate. For each product sold, about 30% of earnings would go to the parts distributors, manufacturers, and companies who source the raw materials for the manufacturers. The rest of the earnings would go to the developers of this product.
If manufactured on a commercial basis:
Assuming a small first production run, the number of devices sold per year may be about 100, depending on the time and money spent on marketing. The cost for parts would be about $130 (replacing the cost of a breadboard with the cost of a small custom printed circuit board). Assuming a person would solder components, flash the software to the system, and assemble the product in 2 hours at a pay rate of $15/hour, it would cost $30 to manufacture one unit.

This cost could be brought down significantly by investing in automation equipment to solder components and flash the software, but for that to be worthwhile, the product would have to be sold on a much larger scale. After subtracting the $130 for parts and $30 for assembly from an estimated selling price of $400, $240 in profit results. At 100 units sold per year, the estimated profit is then $24,000 per year.

After the initial product purchase, the user only needs to cover the cost of battery power to use the product. Assuming the product runs on a supply of four AA batteries (6VDC) and draws 400mA of current. A typical AA alkaline battery is rated to about 2,500 mAh. At 400mA, the product could be operated for over 6 hours for the cost of four AA batteries. In future iterations, a rechargeable battery pack could lower this energy cost for the user.

Environmental:

The portable spectrum analyzer does have a negative environmental impact. The finite raw materials used for the parts need to be mined from the Earth, and the plastics used need to be created. Both processes are harmful to the environment and can potentially disrupt the ecosystems in their areas. This impact however could be lessened by purchasing parts from companies that have proven they take care to minimize their environmental impact. Conveniently, parts distributors display environmental certifications on their websites for individual parts. The electrical power used for development has an environmental impact as well due to generation at the power plant and distribution. Using AA battery power also has a negative environmental impact, as battery waste (acid and metals) is very harmful. This impact could be lessened by using a rechargeable battery pack in future iterations. The current iteration of the product minimizes this negative environmental impact by using RoHS-compliant components, power-efficient components, power-conservative software practices, and power-efficient circuit design wherever possible. Decreasing the amount of environmentally-hazardous e-waste generated by this product is vital for the health of the earth's ecosystems and water supplies, and for the conservation of its resources.

Manufacturability:

To manufacture the product, the printed circuit board needs to be soldered with components manually, software needs to be flashed to the microcontroller, and the enclosure and hardware need to be assembled. These are time consuming processes, but on a larger scale, automation could increase speed and efficiency. Automated assembly machines require large amounts of funding (more than this project has) and a larger scale production to be cost-effective. However, they can significantly improve the assembly rate, soldered-connection reliability, and assembly cost. Labor costs for assembly would decrease, and efficiency would increase. These benefits would appear in the long run; to offset the initial cost of the machines, they would have to produce several large batches of products.

Sustainability:

The device should not physically degrade over time or fall out of calibration, because it can self-calibrate. Once it is built, no maintenance is required aside from replacing (or charging) the batteries. Using a rechargeable battery pack in future iterations would greatly improve product sustainability. Sourcing components and circuit board fabrication from environmentally responsible and sustainable companies, as well as using only lead-free solder, also ensures sustainability. The device uses RoHS-compliant electrical components rather than noncompliant versions wherever possible. Leakage currents in the electronics are minimal. The device does not draw excessive current, and it uses low-power electrical components wherever possible and power-conservative software practices. Together, these design strategies decrease the amount of electrical waste caused by the product; battery packs will last much longer and therefore need to be replaced much less often. Minimizing hazardous waste such as battery packs which contain environmentally-harmful acid, and which drain the earth's rare metal resources, is extremely important for resource conservation and for minimizing harm to ecosystems. The product consequently exhibits sustainability, both in its development and use.

Ethical:

The most significant ethical impact of the product is likely in sourcing the raw materials used by the parts manufacturers. To ensure ethical manufacturing of the product, we purchase parts only from companies that follow laws regarding ethical work conditions and environmental impacts, from the raw material mining to the parts manufacturing. We took care during development to reflect the "golden rule" and utilitarianism in our actions and choices. We chose only parts manufacturers and providers that follow those ethical frameworks. We also followed those ethical frameworks in our use of development equipment and facilities, by considering the needs of other students and faculty who use those facilities.

Unfortunately, due to the limited time and funding for the project, the system uses a battery pack with non-ideal sustainability. Alternative courses of action for future iterations are as follows: Use single-use batteries; Use a rechargeable battery pack that provides more power over its lifetime; Use a rechargeable battery pack that features less environmentally-harmful battery chemistry. Considering a utilitarian ethical framework, these alternative design decisions can be rated morally. Single-use batteries result in the most harmful waste and don't seem to cost less overall. They don't offer the greatest good to the greatest number of people. Rechargeable batteries with long lifetimes such as lithium-ion batteries are an improvement, as they generate less e-waste, which is better overall for the environment, therefore causing less harm to the greatest number of people. Of the three alternative options considered here, the most ethical in terms of utilitarianism is to use rechargeable batteries with less environmentally-harmful chemistry and materials. The least harm will be done to the environment and to the earth's resources and ecosystems, therefore this option causes the greatest benefit (or at least minimal harm) to the greatest number of people.

Health and safety:

During the design, testing, and development of the spectrum analyzer, we take care to follow safe practices regarding the use of electrical equipment. While the DC voltages used should not exceed 6 volts, we always use proper grounding and wiring techniques, as well as extreme care regarding battery-handling. We measure and minimize leakage current. On the user's end, battery corrosion is always a potential concern if batteries are defective or are not cared for properly. In development, we also ensure that the temperature of the product while running does not exceed safe limits, and that lead-free solder is used.

Social and political:

Due to the intended use of the product as well as the high price of the product, most likely only audio professionals and enthusiasts would purchase it. Basically, customers would be mostly upper middle-class professionals and audiophiles of above median income. However, manufacturing of the product would affect people of lower and medium economic brackets as well; workers involved in fabrication and manufacturing electronics are typically within the median income bracket, or just below the national median income. The same goes for the workers involved in mining and transporting the raw materials. The mining and transport of materials and well as the fabrication of some of the electrical components happens outside of the United States. Using international parts impacts the economy of the source countries, as well as U.S. foreign policy and import tax laws.

Other stakeholders include the employees of the part-distributors and online vendors. Device production provides them with business and revenue. Their business strategies, management, and internal systems directly affect the cost of device production. Those of the companies that provide raw materials and transportation also indirectly affect device production costs through the parts-providers.

Development:

During development, I learned how to design an anti-aliasing low pass filter, and how to use SPICE to perform Monte Carlo and temperature-sensitive simulations to verify the design. I also learned how to use the I^2S serial interface to transmit digital high-fidelity stereo audio between devices. I learned that an initial project plan can change greatly as more research is done, and that often the best approach is to start with simpler solutions and develop upon them once they're up and running. Rather than aiming for an extremely well-performing device from the beginning, we chose to create a proof of concept device using simpler hardware with extensive online documentation. The cost and time to develop low-level hardware and software from scratch seemed unrealistic from early on in development.

References:

[1] Lin, Tseng, Cox, et al. "A monolithic audio spectrum analyzer". IEEE Transactions on Acoustics, Speech, and Signal Processing. Volume: 31, Issue: 1, Pages: 288-293. Feb 1983.

Authority: Peer reviewed IEEE journal. Authors are M.S. or Ph.D. holders or have at least a decade of experience.

Use: As a reference for multi-channel processing and filtering techniques, and to understand the underlying architecture of a spectrum analyzer.

[2] "PCM1803A Single-Ended, Analog-Input 24-Bit, 96-kHz Stereo A/D Converter". Texas Instruments Datasheet. 2016. <http://www.ti.com/lit/ds/symlink/pcm1803a.pdf>
Authority: Manufacturer datasheet for an ADC designed for high-quality stereo audio.
Use: As a design reference, to determine how to interface with a controller.

[3] Onion, Peter. "Using a PCM1803A I2S Codec with a Raspberry Pi". PeterOnion.org.uk. 2014.
Authority: Peter Onion is an experienced software engineer. His article shows screenshots of his setup working, and he has a great beard.
Use: To follow his method for getting the device communicating to raspberry pi, and will watch out for the issues he ran into.

[4] Inns, Simon. "Real-Time Audio Spectrum Analyser". WaitingForFriday.com. 2011.
Authority: The author achieved a working spectrum analyzer as intended.
Use: As a rough guide for creating a spectrum analyzer with commonly available low-power hardware.

[5] Bombard, Thomas J. "Audio spectrum analyzer implemented with a minimum number of multiply operations". United States Patent #6,760,674. July 6, 2004.
Authority: Experienced engineer backed by an engineering corporation.
Use: As a reference for low-power, efficient audio spectrum analyzer implementation methods.

[6] Miller, et al. "Automatic equalizer". United States Patent #5,506,910. April 9, 1996.
Authority: Experienced team of engineers backed by an engineering corporation.
Use: Methods for implementing an automatic equalizer could prove very useful for the frequency response measurement feature of this project.

[7] Brixen, Eddy. "Audio Metering: Measurements, Standards and Practice". Burlington, Elsevier Science. 2010.
Authority: Published book by a publisher of scientific texts. Years of experience in the industry.
Use: To gain a solid understanding of the standards for audio measurements and define benchmark requirements for resolution, latency, and bandwidth.

[8] Stanley, Andrew. "Multi-platform audio spectrum analyzer incorporating fast Fourier transform and exponential forgetting factor methods". Microfiche. 2001.
Authority: Published after a review process. Engineering process and methods that led to a working product are explained.
Use: For implementing fast Fourier transform and other methods for an audio spectrum analyzer.

[9] Zukerman, L. G. "Application of a Spectrum Analyzer for Use with Random Functions". IRE Transactions on Instrumentation. Volume: I-10, Issue: 1, Pages: 37-43. June 1961.
Authority: Published in a peer reviewed journal. Experienced engineer backed by a corporation.
Use: Digital random noise creation methods used for the frequency response measurement feature.

[10] French, A. S. "Coherence Improvement in White Noise Analysis by the Use of a Repeated Random Sequence Generator". IEEE Transactions on Biomedical Engineering. Volume: BME-27, Issue: 1, Pages: 51-53. Jan 1980.
Authority: Published in a peer reviewed journal. Experienced engineer backed by a corporation.

Use: Create reliable, true white noise by digital random number generation for frequency response measurement feature.

[11] Baker, Bonnie C. "Designing an anti-aliasing filter for ADCs in the frequency domain". Texas Instruments Analog Applications Journal. 2015.
Authority: Published by Texas Instruments, a successful and reliable electronics corporation. Author is an experienced senior applications engineer.
Use: To design an anti-aliasing filter for the input audio signal before the ADC in the signal chain.

[12] Baker, Bonnie C. "Tutorial: Basics of choosing and designing the best filter for an effective data-acquisition system". Planet Analog Magazine. Aug 2006.
Authority: Author is an experienced senior applications engineer, and she uses reliable references.
Use: To design an anti-aliasing filter for the input audio signal before the ADC in the signal chain.

[13] Analog Devices. "Analog Filter Wizard". Analog.com. Nov 2017.
Authority: Created by a very well-established corporation that designs and produces reliable analog electronics, components, and peer-reviewed literature.
Use: To design an anti-aliasing filter for the input audio signal before the ADC in the signal chain.

[14] Weiss, Aaron. "PCM1803A_Breakout-v11". Schematic. Sparkfun.com. 9 March 2015.

[15] "LTC6255/LTC6256/LTC6257 6.5MHz, 65µA Power Efficient Rail-to-Rail I/O Op Amps". Revision D. Linear Technology. Datasheet. 2010.