USB Audio Interface: An Open-Source Reference Design for Digital Recording

DONOVAN DREWS, University of California, San Diego, USA

Extensive documentation exists online for assembling custom electrical musical gear, such as synthesizers and effects pedals. However, there are significantly fewer guides for creating audio interfaces, despite their rising importance in music recording and distribution. We present an inexpensive and easy-to-assemble reference design for recording high quality audio on a USB host. This design allows both a balanced XLR input and 1/4” input, with adjustable gain for each channel. In addition, it operates as a compliant USB Audio Class device for cross-platform compatibility. In the project repository, open source firmware, schematics, and board files are available to the public in order to bring accurate recording to DIY communities.

ACM Reference Format:
Donovan Drews. 2022. USB Audio Interface: An Open-Source Reference Design for Digital Recording. 1, 1 (June 2022), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Purpose of an Audio Interface

An audio interface is a computer peripheral for “interfacing" analog audio sources to a digital host such as a computer. This allows the user to record audio and store it digitally, for further processing or use. Generally, most computers already contains hardware to perform this operation, which is usually referred to as an audio codec for integrated solutions, and a sound card for discrete versions. However, there are a few key differences. Generally, audio codecs support only a single input channel, and this channel only supports condenser microphones over 1/8" connectors. Codecs like these generally have lower audio quality than external devices as well, partially due to cost-reduction measures as well as interference from nearby computer hardware.

For these reasons, professional audio recording for purposes such as music, podcasting, or video production generally requires an external audio interface. Many microphones have bulky XLR connections which provide superior common-mode rejection, but require special hardware and are not found on integrated audio on most computers. In addition, dynamic microphones have very low sensitivity and require significant gain from a preamplifier, which again is usually not possible without an interface. Lastly, 1/4" input jacks for line-level or instrument audio are also only available on external interfaces. Therefore audio interfaces are important to support the wide ranges of equipment and sources seen in professional audio.

1.2 Motivation

For the above reasons, there are many commercial options that have surfaced in the audio interface product space. These run the gamut from simple one-channel budget options, to dozens of inputs to support an entire studio. In many other areas of music, communities have emerged that center around creating DIY instruments. This can especially be seen for electrical equipment such as effects pedals or synthesizers. However, there is

Author’s address: Donovan Drews, ddrews@ucsd.edu, University of California, San Diego, 9500 Gilman Dr, La Jolla, California, USA, 92023.

XXXX-XXXX/2022/6-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

significantly less activity for creating DIY audio interfaces, despite the growing popularity of home recordings and studios. There are additional technical challenges that come along with creating an audio interface that are above and beyond those found when designing and assembling a synthesizer or effects pedal. This mainly stems from the inclusion of firmware and reliance on low-noise techniques. However, if a design already exists, then modifying and assembling this design would not be significantly more challenging than these other devices. For this reason, the goal for this project is to both create a working reference design, and publish all the necessary documentation for others to recreate this device. Hopefully this will create interest in online communities to begin incorporating audio interfaces into the DIY sphere.

2 BACKGROUND

2.1 Differences Between Audio Sources

All analog audio sources function by oscillating a voltage or a current to represent sound. However, the specific differences between different types of sources requires consideration and in some cases special handling from the perspective of amplifier design. For the purposes of an audio interface, there are several categories which we are concerned with:

- Dynamic microphones
- Condenser microphones
- Pickup instruments
- Line-level audio

These can be grouped into the courser categories of balanced and unbalanced audio, with the microphones generally being balanced, and the other two unbalanced. “Balanced" audio refers to the fact that these sources use cables in which both the negative and positive signal wires are near-identical in construction, with shielding connected to ground. This topology ensures that any interference along the cable length — either from capacitive coupling or EM sources — applies equally to both polarities. This “common-mode" interference can then be eliminated by a differential amplifier on the receiving end of the cable [8]. On the other end, unbalanced audio references the positive signal to ground, by connecting the shielding to the negative terminal on the source side, and to ground on the receiving side. The shielding still provides some resistance to interference, and these sources can be used with standard amplifiers. Microphones generally use balanced cables due to the low signal levels present on their output, though this is not as much of a concern for pickup instruments or line-level audio. The prototypical balanced cable for audio purposes is the XLR cable, which has three terminals for positive, negative, and ground.

Dynamic microphones are a class of microphone that approximately functions as a speaker cone in reverse [7]. There is a sealed diaphragm which is mechanically attached to an electromagnet. Changing pressure in the air causes the diaphragm to move, which then generates electric current by moving the electromagnet relative to a permanent magnet installed in the microphone. These sources have low sensitivity but also low impedance, on the order of 100 ohms. However, due to the inductive nature of the source, the preamplifier needs a comparatively large input impedance to avoid affecting the frequency response. These microphones have low sensitivity but are also inexpensive, making them very popular for recording instrument amplifiers and drums.

Condenser microphones function by a different mechanism. They form a capacitor from a thin metallic membrane and a stationary plate. Changing sound pressure causes this membrane to move, which alters the capacitance, generating a current. This topology is very high impedance, and also requires an external bias voltage to function, where 48 volts is the industry standard level. In order to lessen the difficulties with interfacing with high impedance sources, many condenser microphones contain an integrated FET for impedance matching, which is powered off the bias voltage. Condenser microphones are far more sensitive than dynamic microphones, and

are common for recording voices. These are an extremely common type of microphone, but were not supported for this project due to the additional technical challenges associated with generating and exposing 48 volts.

Pickup instruments generate voltages based on the movements of metallic strings or tines next to an electromagnet, or from the compression of a piezoelectric material attached to a vibrating surface. Usually these transducers are referenced to ground, leading to an unbalanced audio source as discussed above. These sources are also low impedance, but must be matched with a high-impedance amplifier to preserve the frequency response. The standard connector for these types of sources is a 1/4" TS cable, and they are found in most electronic string instruments, and some drum microphones.

Lastly, line-level audio is an term which represents audio sources that are already amplified up to working voltages and currents. The range varies, but generally line-level sources have a maximum amplitude of approximately one volt. This is the easiest type of source to interface with, as no amplification or impedance matching is necessary before digitizing.

2.2 USB Audio

Universal Serial Bus (USB) is an extremely common standard for interfacing peripherals to a computer. Despite the name, USB has a packetized network topology and is not a true electrical bus, and imposes the restriction that packets can only be sent from devices to the host, or vice versa, but not from devices to other devices. The host initiates all the transactions on the network, and devices are polled at a regular interval (frame) to simulate interrupt-type functionality.

There are several different transaction types defined in the USB standard: control, interrupt, isochronous, and bulk [9]. Each transaction type is suited for a different use case, and isochronous transfers are specifically tailored for streaming applications. For the purposes of creating an audio interface, these and control transfers are the only necessary types. Isochronous endpoints are special in that they do not have any error checking or re-transmission in the protocol. This ensures the minimum latency possible, which is critically important for audio applications.

One of the major advantages of USB is the device class system. This part of the specification defines a standard set of types of devices and their associated sub-protocols. For example, there are device classes for mass storage, human interface devices (such as mice and keyboards), printers, cameras, and hundreds of others [3]. These classes cover the majority of use cases for USB. If a device is compliant with the device class specification, then it will generally work with any host operating system, making significantly less work for host driver implementers and device creators. This “plug-and-play" nature of USB devices is one of the major selling points of the standard, and has led to its wide adoption.

One of the device classes called USB Audio Device Class 2.0 (UAC) supports a wide variety of audio peripherals. By designing an audio interface to comply with this standard, no host drivers need to be written, and the device should seamlessly function with any host configuration.

2.3 Analog-Digital Conversion

The most important aspect of an audio interface is the conversion from analog to digital with the smallest amount of distortion or noise. Generally this is done using a discrete integrated circuit (IC) that continuously samples the analog lines and outputs digital representations. There are several architectures for ADCs, with the most common being successive approximation register (SAR) and delta-sigma ($\Delta - \Sigma$). These two architectures have vastly different internal structures and methods of operation, the specifics of which are out of scope [6]. It suffices to say that $\Delta - \Sigma$ ADCs are more common for audio applications, because of higher bit resolutions and less stringent anti-aliasing requirements.

Irrespective of architecture, there are two primary design considerations for using ADCs. One is that ADCs generally have a low input impedance, and require a substantial amount of current from the previous analog stage. Even if the signal is already at the correct line level, a buffer is still required to drive the ADC inputs. The second is that high-frequency content will be aliased back into the frequency range of interest after digitization, usually appearing as noise. Therefore, it is important to have a low-pass anti-aliasing filter before signals enter the ADC. For audio purposes the range of interest is 20Hz to 20kHz, with a sampling frequency of 44.1 or 48kHz, so the -3dB point of the filter is usually placed between 20-40kHz.

3 IMPLEMENTATION

3.1 Systems Design

Design for the audio interface begins from considering the user's expectations and then finding technological solutions to fit these needs. In the commercial space, feature lists and user interfaces at this budget are relatively constant, and this implementation follows suit. Generally, the main inspirations for specifications came from the *Focusrite Scarlett Solo* and the *Behringer UMC202HD*. However, due to the time and budget constraints for the project, many of the specifications and features were relaxed from their commercial counterparts.

The primary concern for a user is what devices are supported by the interface. As discussed in Section 2.1, different audio sources require different connectors, and sometimes vastly different amplifier configurations. The only source from the above list that was excluded in our design was condenser microphones, due to the difficulties in generating and supplying such a high voltage. Line-level audio and pickup instruments can be supported by the same connector and amplifier, by allowing adjustable gain down to unity. However, dynamic microphones use an XLR connector and differential amplifier due to the balanced nature of the source. Having one 1/4" TS connector with a high-impedance amplifier, and one XLR connector with a differential amplifier, allows the device to support a wide range of audio inputs. For the remainder of this document these are referred to as the instrument and microphone inputs respectively.

Another large concern for users is audio quality. There are many metrics for quantifying this attribute, as discussed in more detail in Section 4.1, though the primary goal is to reduce noise and distortion below audible levels. Based on some empirical testing we performed using generated white noise added to clean signals, 60dB is a reasonable threshold for signal-to-noise ratio (SNR) below which the noise is not audible. For this design the maximum signal amplitude is approximately 5Vpp, which translates to approximately 5mVpp as an upper limit on noise amplitude. Noise can be introduced from many sources, though the main sources of concern are power noise, environmental noise, and thermal noise. The specific strategies used for mitigating these are discussed at length in Section 3.2.

A common feature on audio interfaces is the inclusion of "clipping LEDs," which are small indicator LEDs that turn on when the signal is reaching the extremity of the supported range. Audio outside of this range will be clipped and lead to distortion. Generally there are also signal LEDs which show when there is a signal present on the channel. In combination, these allow the user to set the correct gain without having to listen to a recording. For this implementation clipping LEDs are present, but signal LEDs are not.

Lastly the interface needs some way of connecting to a host for recording. Any digital medium would work, and some interfaces use S/PDIF, though USB is far more common. As mentioned above, USB audio devices do not require any additional host drivers, and therefore is a popular choice for input and output devices alike. This necessitates both a physical USB connector on the device, as well as a microcontroller which supports USB functionality.

There are many options for powering the device. The most convenient choice for the user is to rely only on the USB 5V rail for power. This prevents the need for an external power jack as well as an additional wall outlet during use. However, upstream USB ports only need to provide up to 500mA to be compliant, so the device needs

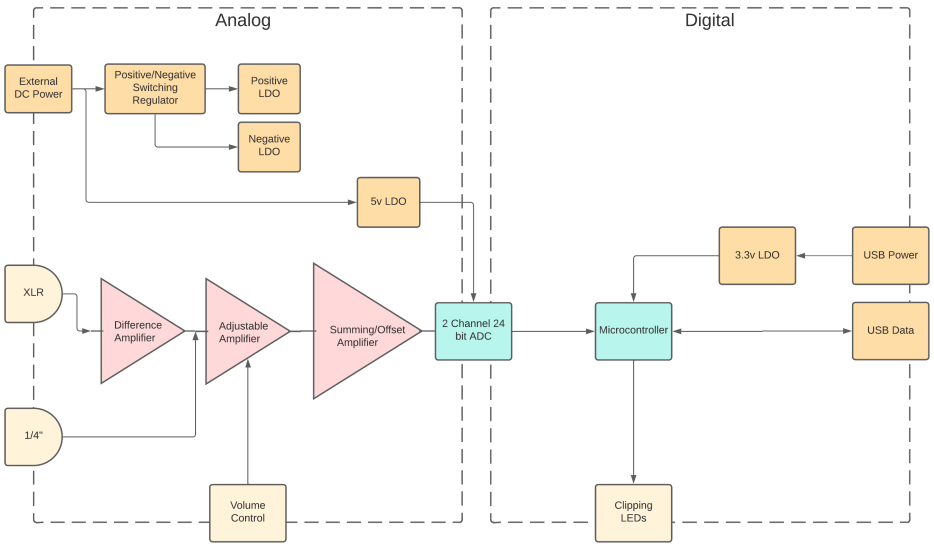

Fig. 1. The Systems Diagram for the Audio Interface

to be able to operate on only 2.5W in this configuration. Many times, the goals of low noise and low power are at odds, so for design simplification we opted for external 9v DC power through a barrel plug.

The systems diagram in Figure 1 describes the amplifier stages and power tree based upon the above considerations. The external voltage rail is assumed to be noisy, and the negative rail will contain switching noise. Therefore both are isolated from the analog supply by an LDO (low-dropout linear regulator). The balanced XLR input has an additional difference amplifier to provide additional gain as well as eliminating the common-mode interference. Both channels pass through single-ended adjustable amplifiers, where external potentiometers adjust the gain. The summing amplifier adjusts the DC bias level to bring the full wavelength above zero volts, in order to bring the input to a safe level for the ADC. This ADC is attached to a USB-supporting microcontroller which passes the data to the host. The specific details of electrical, firmware, and mechanical design which enable this system are described in the following sections.

3.2 Electrical

The electrical design of this audio interface can be broken into three separate categories: analog, digital, and power. For analog, the primary considerations are providing enough gain, and reducing the amount of noise. The microphone used for testing this project was the *Shure SM57*, an extremely common and prototypical dynamic microphone. According to the specifications, this microphone has a sensitivity of 56mV/Pa. A more familiar metric would be that a standard speaking volume of 55dB SPL produces an output of 0.6mV. Therefore approximately 80dB of gain would be sufficient for the XLR signal path. This was split evenly across the differential and adjustable stages. For the instrument input, a unity gain is necessary to support line-level audio, and an electric guitar pickup produces approximately 100mVpp output. For consistency, both adjustable stages were designed for 40dB maximum gain and 0dB minimum gain.

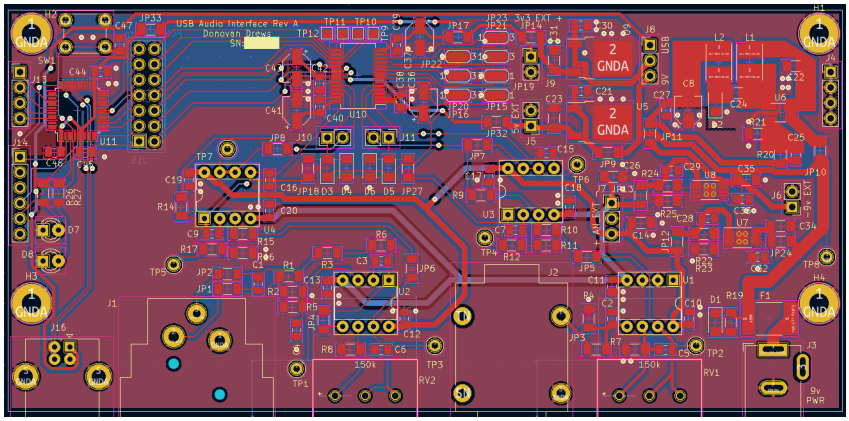

Fig. 2. PCB Layout for the Audio Interface

In amplifier design, the intrinsic thermal noise generated by resistors can be a significant contributor to the overall noise profile. This thermal noise has energy that is linearly proportional to heat as well as resistance [5]. Therefore, it is best practice to reduce the resistances used in the amplifier feedback loop and filters to the minimum level without loading the op-amps. This is especially important for the resistors in the difference amplifier, since the signal amplitudes are very low at this stage. The higher currents resulting from this choice has an additional noise benefit, as it reduces the effect of shot noise as well as capacitively or inductively coupled noise from the environment [4]. The amplifiers were constructed with 4 *TI OPA2134* op-amps, which were chosen for low noise and high power supply rejection ratio (PSRR).

The digital design of the interface hinged on the choice of microcontroller and ADC. For the microcontroller, the *STM32F042* was chosen due to the hand-solderable package, USB support, and price. In addition this microcontroller had an I2S peripheral for interfacing with the ADC. The ADC chosen was the *TI PCM4202*, which provided two channels at 24 bits each, and a total harmonic distortion + noise (THD+N) of -105dB. This ADC is designed specifically for audio purposes, which brings several important features such as DC bias removal through an internal digital high-pass filter.

Delivering power to all these components required multiple different voltage rails. All op-amps were powered off the linearly regulated +7v and -7v rails, generated from low-noise LDOs. The negative voltage were generated by a switching regulator in Cuk configuration, again to reduce noise [2]. The analog and digital 5v and 3.3v rails were both generated with higher-power LDOs. Usually a digital rail would be generated with another switching regulator, but a linear regulator was chosen to reduce to total amount of switching noise on the board, as well as reduce design complexity.

The high-level layout of the printed circuit board (PCB) brought all connectors to the front edge to simplify mechanical design. In addition, digital and switching circuitry was spatially isolated from sensitive analog circuitry, which can be seen in Figure 2. A significant number of test points and jumpers were added to aid testing and verification of the board after assembly.

3.3 Firmware

The firmware running on the microcontroller has 3 main tasks:

- Gather samples from the ADC, and move them into USB packets for transmission to the host.
- Handle USB control and enumeration signals.

Fig. 3. The CAD for the faceplate (left), and the assembled enclosure (right)

- Control the clipping LEDs.

The first of these tasks is accomplished through the use of the I2S peripheral and direct memory access module (DMA). The I2S peripheral generates control signals for the ADC and receives the samples in a bit-serial fashion. The DMA is then configured to move these samples into a circular buffer located in main memory upon each reception. This circular buffer is the hand-off point to the USB side of the firmware stack.

The tinyUSB [10] library was used to implement most of the USB functionality. This is a middleware library that abstracts hardware-specific USB functionality into a generalized software API. USB device, configuration, and endpoint descriptors were created and made accessible to the host through tinyUSB. These allow the host to determine the name, manufacturer, device class, and specific audio attributes of the interface. Isochronous endpoint support in not included in the driver for stm32 in tinyUSB, so support for this endpoint and transfer type was added to a local fork of the library.

After each 1ms USB frame, the last millisecond of audio data is parsed in the circular buffer to determine if the clipping threshold has been exceeded, and to fix endianness issues created by the DMA copying. If the audio is clipping, the LED is turned on for the next 200ms to notify the user. This parsed data is then moved into a tinyUSB FIFO, where it is eventually placed into the stm32's packet buffer memory region. On the stm32 architecture, isochronous endpoints use double buffering to achieve maximum throughput for streaming applications such as audio.

3.4 Mechanical

An enclosure for the PCB is essential to prevent both mechanical damage as well as ESD. The primary consideration in the mechanical layout was ease of assembly and disassembly. For this reason, all connectors are attached to only the front plate of the enclosure, so that the PCB can be entirely removed by only removing one face of the enclosure. In this same vein, the front and back plates are connected with M3 hex bolts to allow for many cycles of attachment and detachment without mechanical failure. For durability the sides of the enclosure are constructed with 1/4" oak boards, and the front and back plates with 1/8" brushed aluminum sheet metal. Both front and back plates were modeled in the CAD program *Fusion360* before being CNC routed and attached to the wooden side panels, as seen in Figure 3.

4 RESULTS

The interface currently is able to be recognized by Linux, MacOS, and Windows as a valid USB audio device, and produce acceptable quality audio from both inputs. The clipping LEDs also light up for their respective channel when the threshold is reached. The interface has been tested with dynamic microphones, pickup instruments,

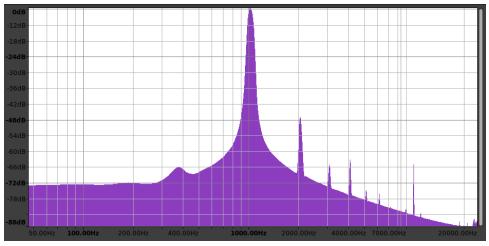

Fig. 4. Frequency plot of the signal received digitally after inputting a 1kHz sine wave on the instrument input

and line-level audio for correct operation. Many useful metrics for this device are specific to audio analysis, and explained in more detail in the following section.

4.1 Measurement Techniques

SNR measures the ratio of the difference in amplitude between the signal when at full scale, and the noise floor. It is a common way of measuring the amount of useful information that can be derived from a signal. Usually, the measurement is specified in dB and has an associated bandwidth over which the noise is collected. For audio, this band is assumed to be 20Hz-20kHz. However, this is a poor proxy for audio quality, as many other effects can lead to audible degradation, such as distortion. Total harmonic distortion (THD) is another measurement that quantifies the amount of distortion in a signal. The measurement is performed by inputting a sinusoid of a known frequency into a system, and then recording the output. Then, Fourier analysis is used to compare the power present in the fundamental frequency as opposed to the sum of all integer harmonics of that frequency. This is a very common measurement in power electronics, RF engineering, and amplifier design.

One of the most useful measurements of audio quality combines these two metrics into “total harmonic distortion and noise" or (THD + N). Sometimes it is also referred to in the inverse ratio form as “signal to noise and distortion ratio" (SINAD) [1]. This is performed by inputting a sinusoid into the system, and then applying a notch filter at that frequency to the output. The ratio between the total power in the unfiltered and filtered signals is the SINAD, and is a good proxy for the audible quality of the system. In order to provide a closer representation of perceptible audio quality, sometimes "A-weighting" is applied to the output, which scales the power at each frequency based on the perceived loudness by the human ear. Generally, this means attenuating the very high and very low frequencies, while amplifying the mid-range.

For the purposes of this project, SINAD is used as the primary measurement of quality, but without A-weighting. Further work is required to test using that specific metric.

4.2 Measurement Results

The audio interface was measured to have a SINAD of 45dB on the instrument channel and 27dB on the microphone channel. When viewing the frequency plot for the instrument input in Figure 4, it can be seen that most of the SINAD reduction is due to the harmonic distortion and not the noise. These measurements fall short of the specification, and there are several theories as to the cause. One of the most likely candidates for the poor performance of the microphone input centers around the crosstalk between channels, which is approximately 30dB. When the instrument input is floating, the high impedance source amplifies EMI to nearly full-scale, and this carries over to the microphone input during testing. A fix was applied to resolve this issue, but additional testing did not take place after this change was made. Another possible cause for both channels is the mismatch between the real and expected ADC sampling frequency, which is 46.7kHz as opposed to 48kHz. Work is ongoing to integrate an external oscillator to resolve this issue and hopefully bring the measurements closer to specification.

4.3 Open Source

All project files necessary to recreate the device are located on the project repository on GitHub [1]. This includes the electrical schematic and the PCB layout, both of which were designed in *KiCAD*. This repository also contains all the code to build the firmware, the local fork of tinyUSB, and the bill of materials containing all the components present on the PCBA. Lastly, the repo contains some minimal instructions for how to utilize these documents to build the device.

5 MILESTONES

The milestones for this project at the beginning were, in brief:

- April 15: The schematic and layout are on the Github repo, with all necessary components present, and have passed through multiple rounds of peer review.
- May 1 (Firmware): The test firmware running on the development board should pass a clean and undistorted sine wave tone to the host computer, which is recorded and spectrum analyzed. All the descriptors should make sense from both a technical and user perspective. Results from these tests should be present on the Github repository.
- May 1 (Electrical): The test log document should contain both expected and measured values for every listed test on every functional block.
- May 19: A recording of analog signal on the host should be made, and accessible from the Github repository. Analysis on the recording to determine quality should be in the same location.
- June 7: Recordings with an actual dynamic microphone and electric guitar should be collected, and analyzed. These items will make it into the final presentation as well as the Github repository. The device should be contained within a metallic enclosure.

Most of these milestones remained unchanged throughout the quarter, though there were some revisions made. The primary change was the addition of another milestone at May 25 for adding the external oscillator to attempt to fix the distortion issue. Unfortunately, many issues arose while integrating this change into the firmware, and the effort was not able to be completed as of June 9. The current status of this effort is that running the CPU clock off of the ADC clock to prevent the mismatch causes the stm32 to miss real-time deadlines and fail, though many optimizations are possible to reach this performance on the slower clock. Another change to the milestones was the requirement for a “metallic enclosure" for the PCB. After some discussions it was realized to be unnecessary, and the design was changed to it’s current form as described in Section 3.4. All of the milestones except the oscillator integration were reached, though some of the documentation on the Github repository is sparse.

6 CONCLUSION

There is a distinct lack of online communities centered around custom DIY audio interfaces, despite similar interest for DIY synthesizers and effect pedals. One of the causes could be the additional difficulty of creating audio interfaces beyond these other devices due to the mixed-signal component. Our goal was to create a reference design which could provide a starting point for these DIY communities to begin exploring this design space.

Towards this goal, an analysis was performed on existing commercial solutions, and these specifications we used to guide the design of a custom PCB containing amplifiers, an ADC, and a microcontroller. Firmware was created for the microcontroller to allow analog inputs to be amplified, sampled, and then sent to a host for digital recording. Once the electrical and firmware aspects were verified, a mechanical enclosure was designed and integrated. This final design was tested using a variety of quantitative and qualitative metrics, and found to have acceptable quality, but not fully reaching the specifications of the design.

[1] https://github.com/DonDrews/audio_interface_cse145

6.1 Future Work

As mentioned in previous sections, several important features were excluded to simplify the first revision of the design. Two of the most important features are support for condenser microphones, and a direct monitoring system. Further revisions of the audio interface could include these features to move closer to the functionality of commercial equivalents. In addition, work is ongoing to fix issues with distortion found in the current revision by installing an external oscillator to provide the ADC clock.

ACKNOWLEDGMENTS

This project was aided by many of my colleagues, who I would like to thank and give credit where it is due. Ravi Johnson reviewed the electrical schematics and layout through multiple cycles of revision, and assisted with some soldering. Patrick Paxson performed with multiple instruments to create demos, and additionally mixed these demos. Kunal Singla operated the CNC router to create the front and back plates of the enclosure. Devanshi Jain assisted with testing and verification of the PCB. Lastly Nathan Hui and Christopher Crutchfield advised this project throughout the previous 3 months.

REFERENCES

[1] 2011. IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters. *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)* (2011), 1–139. https://doi.org/10.1109/IEEESTD.2011.5692956

[2] S. Cuk. 1983. A new zero-ripple switching DC-to-DC converter and integrated magnetics. *IEEE Transactions on Magnetics* 19, 2 (1983), 57–75. https://doi.org/10.1109/TMAG.1983.1062238

[3] USB Implementers Forum. 2022. *Defined Class Codes.* Retrieved June 9, 2022 from https://www.usb.org/defined-class-codes

[4] Paul Horowitz and Winfield Hill. 2015. *The Art of Electronics, Third Edition.* Cambridge University Press.

[5] Art Kay and Tim Green. 2019. *Analog Engineer's Pocket Reference.* Texas Instruments.

[6] Walt Kester. 2005. Which ADC Architecture Is Right for Your Application? *Analog Dialogue* 39, 06 (2005).

[7] Carl R. Nave. 2001. *Hyperphysics: Microphones.* Retrieved June 9, 2022 from http://hyperphysics.phy-astr.gsu.edu/hbase/Audio/mic.html

[8] Sathybama Institute of Science and Technology. 2021. *Audio Production Theory.* Retrieved June 9, 2022 from https://sist.sathyabama.ac.in/sist_coursematerial/uploads/SVCA1301.pdf

[9] Craig Peacock. 2010. *USB in a Nutshell.* Retrieved June 9, 2022 from https://www.beyondlogic.org/usbnutshell/usb1.shtml

[10] Ha Thach. 2022. *TinyUSB.* Retrieved June 9, 2022 from https://github.com/hathach/tinyusb