Multichannel Audio Synthesizer

Joseph Farnham, B.S. Computer Engineering, jfarnham3, jfarnham3@gatech.edu
Moradeke Olumogba, B.S. Electrical Engineering, molumogba3,
molumogba3@gatech.edu

Submission Date: 06/02/2020

Table of Contents

Change numbering

Executive Summary

The goal of our project is to develop a device that records analog voltages on multiple channels and convert that data into audio files. Devices that sample audio simultaneously across several channels are too expensive for most developers and hobbyists. Our solution, which will cost $104.93, will make this technology accessible to a wider range of people.

With guidance from Dr. Smith [7], we have chosen several technical constraints. Our device needs to have four stereo speakers that feed into eight channels of audio, and each channel needs to sample at rate of 196,000 samples per second or greater. Our device will also need to convert the digitized audio signals into a format that can be read via USB. The device then needs to connect to a USB 2.0 port on a laptop, where we will have written code that is capable of decoding the serial data and converting it into an uncompressed audio file, or a .wav file. Thankfully, there are inexpensive chips on the market that perform most of the hard work. Our jobs involve figuring out how to interface with those chips to provide an end-to-end solution.

We will use CAD software, such as National Instrument's Multisim, to design a circuit that can effectively communicate with these chips. We will also be utilizing the USB 2.0 interface, for which we are provided a functioning driver by FTDI, the company that makes the USB chip we will be using. From there, we will write Python code that will interface with this driver and convert the audio data sampled on each channel into an audio file.

Lastly, our project is based on the work of a Senior Design project that students undertook in 2014. We will be referencing their project throughout this proposal.

Nomenclature

Codec -- Encoder and Decoder in the analog to digital sense

LT Spice – Linear Technologies Spice Simulation tool

Multisim – National Instruments Multisim Simulation tool

Ultiboard – National Instruments Ultiboard Simulation tool

USB – Universal Serial Bus

Multichannel Audio Synthesizer

1.   Introduction

Team Audio Synthesizers is requesting $104.93 to develop a multichannel audio synthesizer. We intend to prototype and build a cost-efficient device that can sample audio simultaneously on multiple channels and convert that audio into uncompressed files that are playable on a computer. This project builds on the work done by a group of Georgia Tech ECE students in 2014 for their senior design project. There is not a device on the market that offers the described functionality at a price that is affordable to the average developer and hobbyist [1]. In conversations with Dr. Smith, we were told that there are a variety of projects could use a device with this functionality [8]. For example, there were a series of student submarine competitions at Georgia Tech where one goal was for a team's submarine to move itself to an underwater device emitting a series of ultrasonic signals [3]. Because the students did not have the expertise to make a device from scratch, and were operating within a budget, they would have to outsource parts of the project. We hope to make a device that can offer similar capability to devices that already exist, but at a lower cost so that the device is available to a wider range of people.

Our main motivation is to complete a senior design project with the constraints of COVID-19 and not being collocated. In addition, hobbyists and developers need more cost-effective technology, and we aim to provide one to them.

Getting into the technical details, we will be sampling eight channels of audio simultaneously. That audio will then be connected to an audio codec chip, which will then interface with an FTDI USB chip. We will also be writing Python code that can convert the streamed serial data into uncompressed audio format. FTDI supplies us with a USB2.0 driver that can interact with their chip, so we will be leveraging that technology in our solution. Technical challenges include learning how to properly interface the codec chip with USB chip, writing Python code to communicate with the USB chip, and demultiplexing the eight channels of audio in software. Key performance aspects are listed below in the Technical Specifications section.

2.   Project Description, Customer Requirements, and Goals

The consumers of this device want a technology that can sample multiple channels of audio simultaneously and output files that replicate what was heard through the stereo speakers [1]. Ideally, this device will be easy to use; the end consumer will not have to care about the internals of the device. The end user will need to tell the device to listen to audio for a certain length of time, and subsequently listen to the file produced by the Python code. The device will be cheaper than the alternatives that already exist, which enables more people to purchase this device.

This project involves utilizing an audio codec chip, an FTDI USB 2.0 chip and driver (see figure 2), and speakers for the final demonstration. The audio codec chip will convert audio data sampled across multiple channels to digital signals. Those digital signals will then be fed to the USB chip, which will convert those signals into serialized data that can be transmitted over a USB interface. The device will then plug into an USB 2.0 jack on our computer. The computer will also be running Python code that interacts with these chips. The Python code will convert the serialized audio data into uncompressed audio files that are playable with a music application such as Audacity.

The device will be sampling eight channels of audio simultaneously. It will also be using the USB 2.0 driver that FTDI provides with their chip. This significantly reduces the time and effort needed to build a functioning device.

3. Technical Specifications

Multichannel Audio Synthesizer		
2020 June 1		
Technical Specifications		
	Proposed	Quantitative
Specification	Quantitative	Goal
	Goal	Demonstrated
Sample Rate	>= 192k Samples/Second	
Sample Bandwidth	25-80 kHz	
Channels	8	
Bits/Sample	24 or 32	
File Format	File must be a .wav file	

4. Design Approach and Details

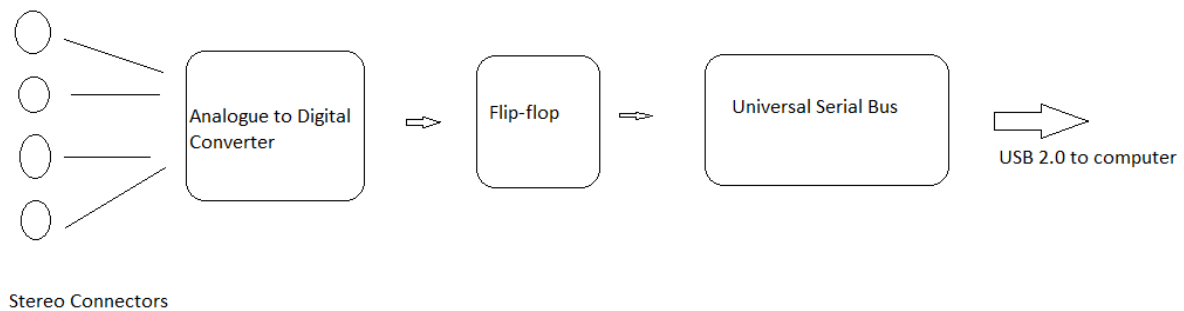## 4.1 Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

Figure 1. Circuit proposed by 2014 cohort.

Much of the design (see figure 1) and objectives for this project are taken from the Senior Design cohort of 2014. We will have to design and fabricate a printed circuit board, as well as convert each channel of serialized audio data into a separate .wav file. Speakers will be connected to an audio codec chip which will then be connected to a USB integrated circuit (see figure 1). This circuit will then use a USB 2.0 cable to feed the audio data into a Windows computer with a USB 2.0 jack. That computer will then run Python code that will convert the USB stream to a wav file. The tools and methods to guide the work include National Instruments Multisim Simulation tool, National Instruments Ultiboard Simulation tool, Python, the FTDI USB2.0 driver and USB 2.0.

The design needs to meet the following specifications:

- Sample each channel at 196k samples/sec
- Sample 24 bits/sample
- Be able to convert analog signals to digital signals
- Serialize the digital signals
- Send the serialized data to a computer
- Convert serialized data to wav file

When digitally sampling a signal, the Nyquist Frequency must be considered. The Nyquist Frequency says that we must sample at a rate of at least twice the highest frequency to mitigate aliasing effects under digitization. As the resolution is 24 bits per sample given that there are 8 channels, the codec must be able to sample at a rate of

$$2 \times \frac{samples}{second} \times bit\ resolution \times number\ of\ channels$$

which necessitates a codec of approximately 10 Mbps to the nearest whole integer rounding up [9].

The technical trade-offs are bulleted below:

- USB 2.0 is not as fast as USB 3.0, but is cheaper. We are aiming to create a low-cost solution, so we chose the cheaper option.
- Python is slower than C, but it is easier to write. The Python code may also be calling C libraries under the hood.
- We are using a prewritten USB driver instead of writing our own. This will save us a ton of time and effort.

4.2   Preliminary Concept Selection and Justification

Dr. Smith presented us with several projects to complete our senior design course [8], while also giving us the option to define our own project. The chosen project allows us to use our complementary skillsets and is possible to complete with the constraints of COVID-19 and not being collocated.

4.3   Engineering Analyses and Experiment

Team Audio Synthesizers

Prototype testing will be done on a breadboard. The FTDI USB chip is affixed with pins, as is the audio codec chip. Once a design has been tested and agreed upon, it will then be assembled onto a printed circuit board. It is planned to undergo a Mathcad, LT Spice, Multisim, and Ultiboard simulation that would ensure feasibility and operability of circuit before layout and build. The Mathcad and LTSpice simulations may not be possible to undergo due to COVID-19 coupled with Summer Semester constraints on time. Mathcad is the tool to convert circuitry equations into graphs that can depict frequency. If undertaken, frequency plots shall be ascertained.

4.4   Codes and Standards

The driver will be written in Python3 and will comply to Python3 standards. We will be using the USB 2.0 protocol to transfer data over the USB cable.

5.   Project Demonstration

We will play audio near the speakers. We will then listen to the wav files generated by the Python code and see if it matches what we played.

6.   Schedule, Tasks, and Milestones

	Proposed	Check Here
Tasks and Milestones	**Goals**	**When Complete**
Printed Circuit Board		**And Include Date**
Schematic	Yes	
Copper Layout	Yes	
Fabricated	Yes	
Populated	Yes	

Team Audio Synthesizers

	Tested	Yes	
Development Software			
	Confirm USB termination	6/15	
	Confirm USB IC I/O	6/15	
	Confirm Codec Interfacing	7/1	
	Confirm Codec Data	7/1	
User Python Interface			
	Start	6/25	
	Stop	6/25	
	Set USB Variables	6/30	
	Get USB Variables	6/30	
	Set Codec Variables	7/5	
	Get Codec Variables	7/5	
	Set Data Sink	7/10	
Output to File		7/15	
USB Interface			
	FTDI	6/25	
	PC	6/30	

7.   Cost Analysis

7.1   Cost Analysis

We believe this project will cost £84.01 excluding the cost of software amenities and

borrowed tools. The possible manual labor if assembled for us gives a codec cost of £41.69,

Team Audio Synthesizers

the FTDI USB circuit costs £17.32, five breadboards cost £5, and the estimate cost for layout fabrication is £20.

Current Status

8.   Leadership Roles

Joseph Farnham – Web Master

9.   References

[1] A. Ward, Y. Lu, E. Patterson, M. Yang, F. Xie "Multichannel ADC Device with Simultaneous Conversion" Presented at ECE4011 Senior Design. [PDF]. Available: http://ece4012y202002.ece.gatech.edu/14spring/ECE4012L1A/ws2/

[2] W. Smith. (2020). Analog Devices (email of 5/29/2020)

[3] W. Smith (2020), AUVSI submarine vehicle competitions (email of 5/28/2020)

[4] W. Smith (2020), Initial Challenges (email of 5/27/2020)

[5] W. Smith (2020), Tasks and links to suppliers (email of 5/21/2020)

[6] W. Smith (2020), Prior Project's website (email of 5/19/2020 3:41 PM)

[7] W. Smith (2020), Suggestion of parts (email of 5/19/2020 1:38 AM)

[8] W. Smith (2020), Project description (email of 5/14/2020)

[9] W. Smith (2020), Expected serialized bits (email of 5/29/2020 2:13 PM)