Polyphonic Analog-to-MIDI Converter for Musical Applications

Senior Design 1

Updated Divide & Conquer Document

University of Central Florida

College of Engineering and Computer Science

<u>Group 12</u>

Sean Dillon	Electrical Engineering
Andrew Obeso-Silva	Computer Engineering
Colin Smith	Electrical Engineering
Noah Watts	Computer Engineering

Project Narrative

Acoustic instruments typically require manual transcription and do not integrate into electronic systems. A system that automates transcription and reacts to musical events will be useful to both musicians and electronics manufacturers. Typically, a musician would have to play a song on their acoustic instrument and then transcribe each note by hand, but this device provides a more efficient alternative. Similarly, many music hardware and software systems use the MIDI protocol, and there is no way to integrate an acoustic instrument. This would allow musicians who use acoustic instruments to streamline their recording, song writing, music notation transcribing, live performance, and more.

This device will take analog input from a music instrument or peripheral device (microphone or pickup) and convert it into a MIDI stream that can be used for transcription, virtual instrument control, etc. It will be in the format of a small portable that can be set on a desk and plugged into a computer or placed on the ground during a performance. Additional features may be added to make the project more functional for live performance or music production, such as a buffered signal path, a tuner, etc.

This device will assist musicians in being able to transcribe performances and trigger MIDI-compatible devices for live performance. This device will also be useful to programmers, hobbyists, and electronics manufacturers who wish to integrate music with their systems. This device can plug into any other device with a USB or MIDI connector that can accept a MIDI stream input (e.g. a computer).

It is expected that the analog signal will originate from a musical instrument such as guitar, piano, or voice. Through signal analysis, our device will determine the note or notes being played and use that information to create a digital MIDI signal that can be sent to other devices.

MIDI is a communication standard that is widely used in music technology to communicate information such as note frequency, duration, volume, etc. We will adhere to the MIDI standard in this project to make it compatible with existing MIDI devices. Some of these devices use a specific MIDI cable, while others use USB. To be compatible with as many of these as possible, our device should be able to output both via USB and via MIDI cable.

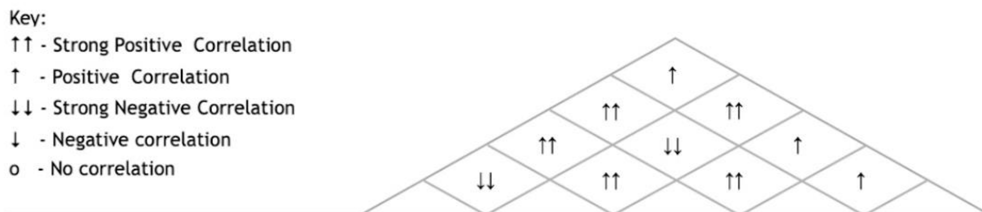
Similar devices can be found such as the Sonuus G2M which can track an instrument or microphone and output a monophonic MIDI track. This means it can only output one note at a time, which is fine for an instrument like a trumpet or saxophone that normally only plays one note at a time. Many instruments, however, can play multiple notes at once. The fact that it is monophonic is a common complaint or at least an observation that consumers make of this and other products.

We hope to make something like the Sonuus device but with polyphonic capabilities. This would allow instruments like an electric guitar or an acoustic piano to play chords and harmonies that will be converted into MIDI. We will have to develop an efficient algorithm for decomposing the instrument signal into individual notes to keep the latency low enough for use in a performance or production setting. The Sonuus and other similar devices output via either USB or MIDI connector. We plan to use both connectors to output from our device as they are the standard.

Additionally, we want the user to be able to bypass the device, sending a separate signal out of the box and into their amplifier or other audio interface so that they may easily use our device in combination with others.

House of Quality

As far as form factor, the device should be lightweight and portable for ease of use in a live setting. Additionally, to this end, it should be able to cool passively to reduce noise produced by the unit during performance. Thus, it should use less power to reduce the amount of heat produced. There should also be low latency between the arrival of the input signal and the output of the related MIDI notes to allow for timing-critical applications, such as live recording or music sheet generation.



Direction of Requirement	Engineering Requirements → / Customer Requirements - (What) ↓	Cost	Latency	Accuracy	Processing Power	Frequency Bandwidth
	Direction of improvement	↓	↓	↑	↑	↑
↓	Cost	↑↑	↓↓	↑↑	↑↑	↑
↓	Latency	↓↓	↑↑	↑↑	↓↓	↑↑
↑	Accuracy	↑↑	↑↑	↑↑	↑↑	↑
↑	Compatibility with MIDI devices	↑	↑	↓	↑	↑
↓	Power Usage	o	↓	↑↑	↑↑	↑
↑	Portability	↑	↑	↓	↓	↓
	Target	< $500	< 100 ms	Up to 6 notes at a time and up to 2 notes a second	25 MHz	82 Hz - 12.6KHz

Fig. 1: House of Quality

Block Diagrams and Device Functionality

It is shown in Fig. 2 that the device will have two inputs and two outputs for the instrument signal. The instrument signal will be sent to the device via either ¼" jack or XLR and the device will have ¼" and XLR outputs as well. While our minimum acceptable prototype version of this project will likely be designed for one specific microphone for simplicity, a final production version should be able to support any microphone either through a general algorithm or through a calibration test, and should be able to provide 48V phantom power to any microphone that requires it.

The device will be powered in one of two ways: either via USB 5V power coming from a computer it is connected to or a 9V power supply. This choice gives the user flexibility for the setting they are in, whether they are using this device in the studio or on stage.

In order to convert the instrument signal into a MIDI signal, we will have to first discretize the analog signal into a digital signal, then perform a low latency discrete fast Fourier transform to analyze the frequency content of the signal. As shown in Fig. 3, this frequency data is passed through several layers of processing.

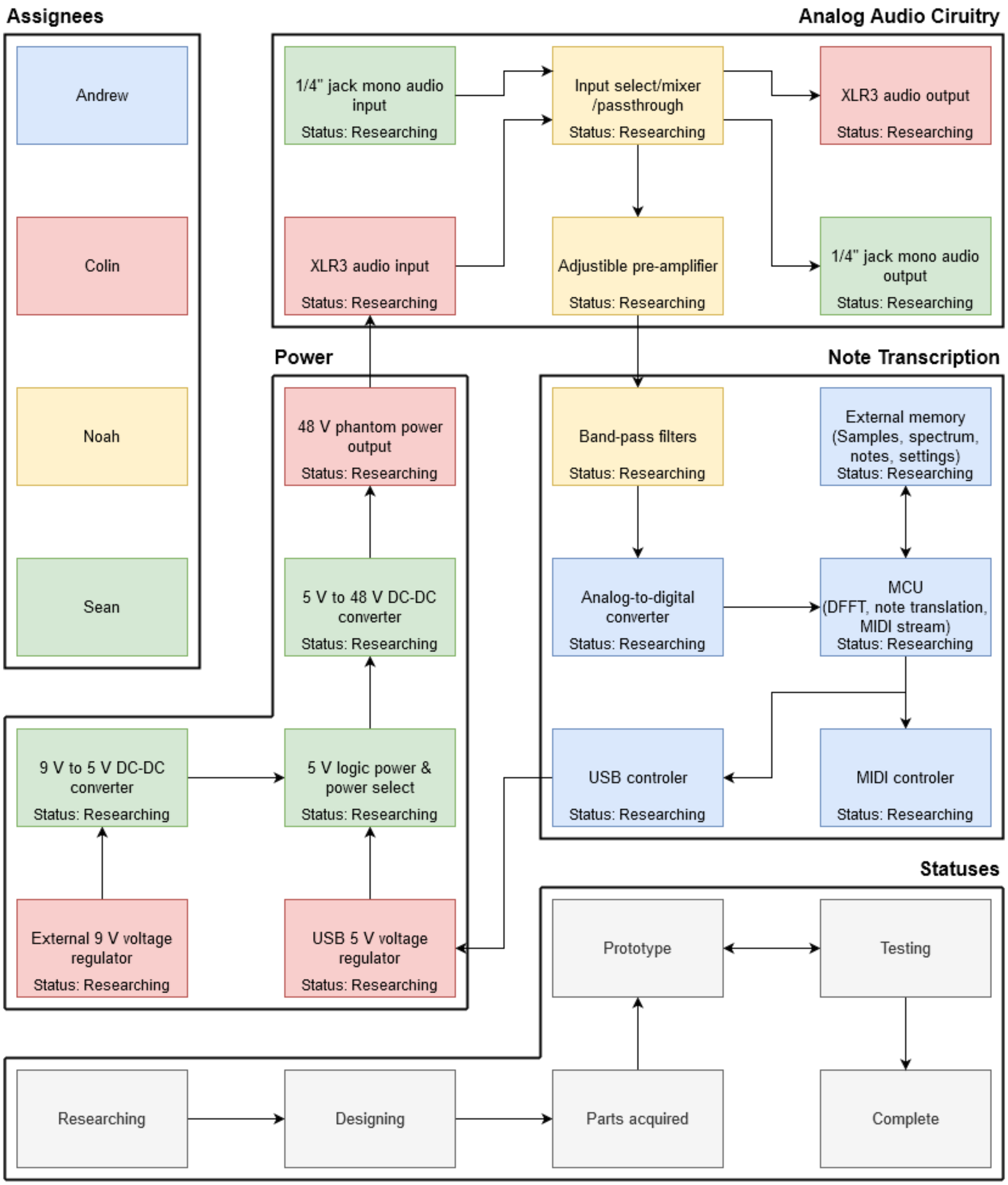


Fig. 2: Hardware Block Diagram

First, frequencies below a certain threshold of magnitude are filtered out to reduce noise, low magnitude harmonics, resonance, and overtones. Next, the magnitudes of the remaining frequencies are normalized. Finally, frequency spikes that have a low Q factor and wide bandwidth are filtered to remove resonance and noise that may have passed the magnitude threshold filter. After filtering, only the fundamental frequencies of each note should remain. We pick 6 frequencies to translate to MIDI notes and then generate a MIDI stream for the MIDI output and a MIDI-encoded USB signal for the USB output.

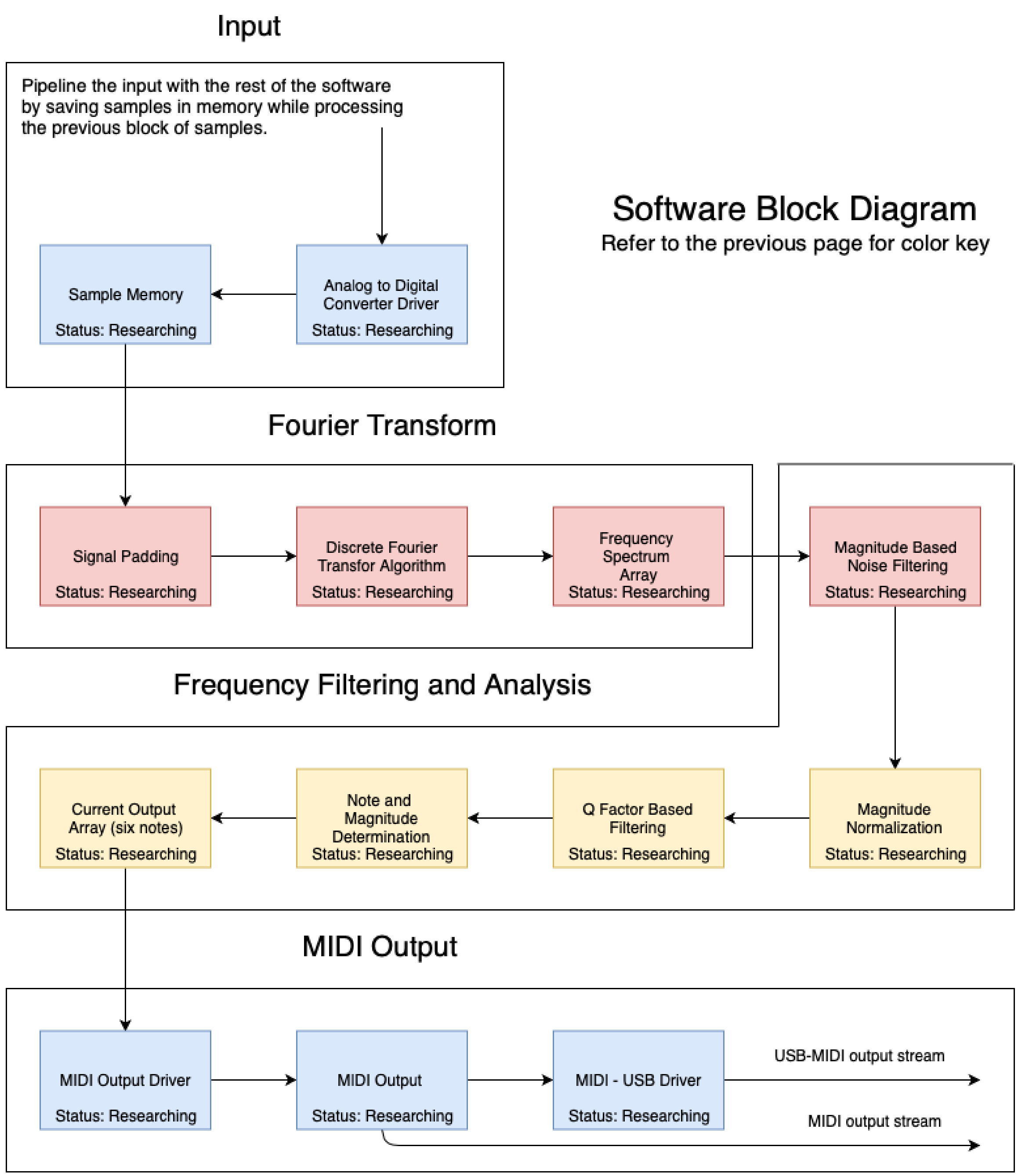


Fig. 3: Software Block Diagram (Uses same Assignees and Statuses keys as Fig. 2)

Requirement Specifications

Requirement	Description	Unit	Value
Polyphony	Minimum number of simultaneous notes converted.	N/A	6
Note Recognition Time	The maximum amount of time between recognition of notes.	s	0.5
MIDI Cable Output	Must have a MIDI cable output.	N/A	N/A
USB Cable Output	Must have a USB 2.0 output that can interface with a computer.	N/A	N/A
Note Translation Time	Maximum amount of time between signal reception and note output.	ms	100
Max. Component Temperature	Maximum long-term temperature. Must not require active cooling.	℃	Refer to parts' specs.
Length	Length of device.	in	12
Width	Width of device.	in	9
Height	Height of device.	in	6
Max. Weight	Must be lightweight for portability.	lb	0.5
¼" Input Voltage Range	Max. voltage range for this input.	V	± 2.5
XLR3 Input Voltage Range	Max. voltage range for this input.	V	± 2.5
Sample Frequency (higher voices)	Minimum sample frequency for midrange to treble range voices.	kHz	26
Sample Frequency (lower voices)	Minimum sample frequency for low midrange to bass range voices.	kHz	1
Min. Input Impedance	A high input impedance is typical for musical devices.	kΩ	5
Device Cost	Maximum device cost.	USD	500

Table 1: Requirement Specification Table

Constraints and Standards

To create a product that is compatible with the greatest number of other MIDI devices and MIDI-compatible interfaces and computers, we will adhere to the following standards.

Standard Name	Description
USB 2.0	Communication with computer, +5V power in.
MIDI 1.0	Defines physical layer and transmission protocol for musical data.
XLR3	Microphone in, signal out.
Phantom Power	Power for condenser microphones.
TS ¼" Connector	Tip + Sleeve connector for instrument cable.
Sleeve+ Tip/Center- ⅛" Barrel Jack	+9V power jack.

Table 2: Table of Standards

Constraint	Description	Unit	Value
MIDI physical layer standard	Required bit rate for MIDI physical layer standard.	kb/s	31.25
MIDI Note Range	Musical note range frequency for MIDI.	Hz	8.18 – 12500
Phantom Power Constraint	Maximum current for phantom power.	mA	10
Electromagnetic interference	PCB and enclosure design must have pathing and shielding that reduces EMI from internal and external sources on the primary signal path.	TBD	TBD
USB data rate	USB low-speed data rate standard.	Mbps	1.5

Table 3: Table of Constraints

Expected Budget and Financing

We do not expect to get a sponsor to aid us in financing for this project so the project will be self-funded. For this reason, we hope to keep the project as low budget as possible without interfering with the functionality of the design. We already own a microphone and power supply, so those are listed as free.

Component/Device	Budget	Component/Device	Budget
PCB	200$	Case	30$
Microphone	Free	Passive components	30$
Power Supply	Free	DC-DC converters	50$
Analog-to-Digital Converter	10$	Input/output interfaces	30$
48 V Phantom Power Source	25$	Voltage regulators and active components	50$
USB cable	2$	Microcontroller	20$
MIDI cable	3$	**Total**	**450$**

Table 4: Expected Budget Breakdown

Project Milestones (Both Semesters)

During Senior Design 1, most of the milestones have to do with documentation deadlines rather than actual testing, implementation, etc. In Senior Design 2 we will be implementing and testing our design.

Senior Design 1

No.	Task	Milestone Date	Status
1	Pick Project Idea, Assign roles	5/22/2020	Completed
2	Initial Project Documentation- Divide and Conquer	5/29/2020	Completed
3	Updated Divide and Conquer document	6/5/2020	In progress
4	60-page draft	7/3/2020	Not Started
5	100-page draft	7/17/2020	Not Started
6	120-page Final Document	7/28/2020	Not Started
7	Breadboard testing	7/28/2020	Not Started
8	Begin ordering parts	7/28/2020	Not Started

Table 5: Senior Design 1 Milestones

Senior Design 2

No.	Task	Milestone Date	Status
1	Implementing Note Detection & Test Software	8/18/2020	Not Started
2	Finish first draft of drivers	8/18/2020	Not Started
3	Testing Parts	8/25/2020	Not Started
4	Possible Redesign	9/15/2020	Not Started
5	Finalized Design	10/6/2020	Not Started
6	Final Prototype working	11/17/2020	Not Started
7	SD Showcase	TBA	Not Started

Table 6: Senior Design 2 Milestones

Decision Matrix

We considered one other project idea: a musical switch, activated when it “hears” a specific melody or chord. We ultimately decided on the Polyphonic MIDI Converter because it allowed for more hardware design and it has more practical use.

	Musical Switch	Polyphonic MIDI Converter
Description	Detects a melody or chord and outputs a signal or activates external device.	Converts notes and chords into a MIDI signal that can be used by DAWs or other MIDI devices.
Motivation	It is a novelty. Use it for games? Put it in an escape room?	Has many applications, including note transcription and performance effects triggering. Building something useful would be great!
Stretch Goals	● Programmable by cell phone app ● MIDI compatibility	● Identify different instruments based on harmonic structure ● Buffered / Bypass switching
Notes	Peripheral devices may require mechanical components	All electrical and computer engineering

Table 7: Decision Matrix