Facilitated Instrument Learning Design Document

Jiajun Xu, Theodore Lao, Chris Chen

02/18/2018

1 Introduction

1.1 Objective

Problem Statement

Musicians spend a substantial amount of time learning the positions of chords and notes on new instruments they are interested in learning. Facilitated instrument learning will allow one to sing, hum, or play another instrument and the currently played notes will be mapped onto the new instrument in real time. This allows a beginner, with little musical background, to sing a melody they wish to play and learn it on a new instrument and also allows a professional, with an extensive background on musical theory and other instruments, to compose music on new instruments.

Proposed Solution

The proposed solution is for the acoustic input recorded into an analog MEMS microphone and the signal will go through an ADC. This will connect to a DSP chip which handles the frequency analysis. This will connect to a microprocessor which controls the LEDs on piano keys. The DSP chip filters noise, additional signals from the instrument, and harmonics to determine the current note being played. The frequency of the notes will correspond to positions on an instrument which can be indicated by LEDs.

1.2 Background

The software complexity will be within the pitch detection algorithm. Noise, additional harmonics, and overtones first come to mind when filtering. Then there are frequencies which come from striking strings and frequencies which come from resonances in instrument bodies. The pitch

detection algorithm will contain signal processing and machine learning algorithms such as autocorrelation and k clustering to accurately determine the fundamentals.

The hardware complexity of this project will be the system integration of the power, microphone, controls, and microprocessor.

There are other products in the market which help people earn instruments but they have different limitations which this project overcomes.

SCI V9000 KEY/NOTE VISUALIZER. This $2000 product is limited because notes must be pre-selected on the piano for students to learn from later. With the proposed project, users produce the melodies with their voice or an instrument and the notes on the destination instrument will be shown in real time.

Synthesia This piano learning subscription service is limited to select songs the company has pre-transcribed for its users, so customers are unable to write songs with melodies they have come up with.

1.3 High-level Requirements List

- Detect pitches accurately at least 80 percent of the time,
- Record at least two instruments, voice and guitar, and accurately map the notes to the piano.

2 Design

2.1 Block Diagram

Successful operation of this system relies on three units: a power supply unit, control unit, and audio unit. The audio unit listens to the acoustic sounds the signal is amplified into the ADC of the control unit. The signal processing unit contains the algorithms to determine the fundamentals of the acoustic signal. All modules in this unit are supplied by 3.3V and 5.0 V which are output by the voltage regulator in the power unit. The power unit contains a USB/USB-C connection for wide charging accessibility.

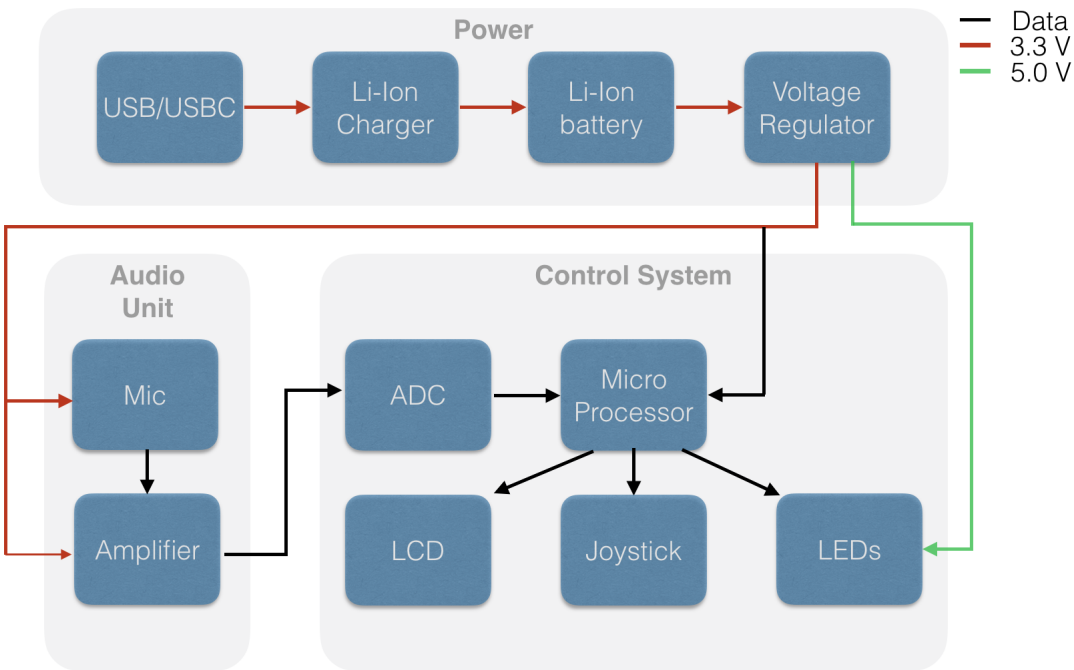

Figure 1: Project Block Diagram

2.2 Power Supply Unit

The Power module connects two 3.7V 18650 li-ion batteries to voltage regulators and a charging circuit. The voltage regulators step down to two 5V rails, each separately running a 5V to 3.3V LDO. These will supply power to other components in our system. The batteries will be charged via USB to ensure wide compatibility with current chargers.

Our design incorporates TI's BQ24103 battery charging IC which allows for detection of high current charging sources. The system is designed around a nominal voltage of 7.4V, implementing the use of two configured buck converters to step down to 5V for the external IO and a separate 5V rail for the micro-controller. The microcontroller is always hooked up to the battery, and draws a total of 20nA in energy mode 4. The second 5V buck and 3.3V LDO is enabled through microcontroller GPIO, allowing us to turn on the LEDs and audio unit only when they are needed.

Requirements	Verification
USB Input The USB input should be a 5V ± 0.3V supply	• collect an array of 5v USB charging sources • measure voltage across VBUS before and after connection to battery charging circuit
Li-Ion Charging IC Battery charger should follow CC/CV charging	• deplete battery and attach a power source to the IC and battery pack • record battery charging current and voltage until fully charged • verify circuit follows CC/CV charging scheme
Li-Ion Battery The battery pack should be 8.4V±0.1V fully charged and 5.8V ±0.2V fully discharged	• measure battery voltages at full and empty, and verify it falls within battery constraints
DC/DC conversion 5V rails should be 5V ± 0.1V 3.3V rails should be 3.3V ± 0.05V	• measure open voltage and verify correct voltages • simulate resistive load, and verify output voltages

2.2.1 Power Unit Schematic

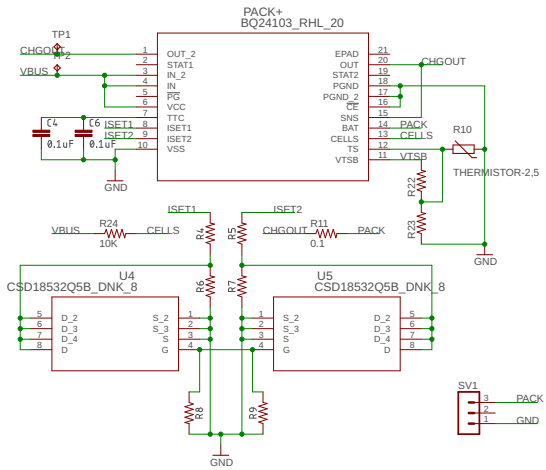

Figure 2: Battery Charger Circuit Schematic

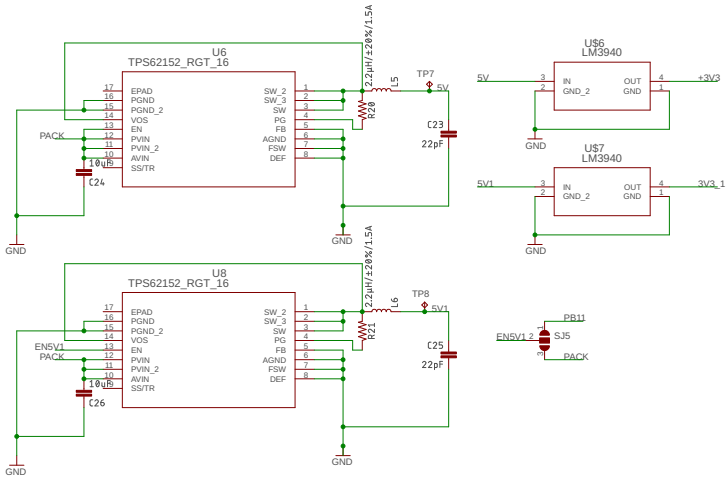

Figure 3: Power Regulator Schematic

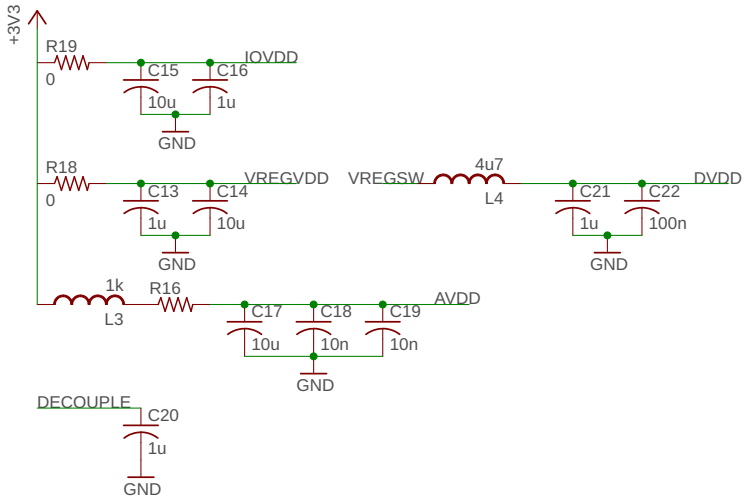

Figure 4: Microcontroller Power Schematic

2.3 Signal Processing Unit

The Signal Processing module takes the analog signal as an input, and outputs the frequencies of the signal by taking the Fourier transform of the analog input data. It contains the microprocessor and LEDs. The microprocessor will perform algorithm work on the frequency domain data. After determining the fundamental frequencies of the audio, the corresponding LEDs on the piano keys will turn on.

At the heart of the unit, we included an EFM32 Jade Gecko processing unit, based on the ARM Cortex M4 core. We found this processor to be the best for our application due to its extremely low power consumption, flexibility with power requirements, and the ARM M4's inclusion of enhanced DSP libraries. For the design of a portable device, the energy usage is relatively restricted, yet we still have to perform additional processing on received input signals.

The processor is programmed through SWD break out headers, and we have included additional headers for potential I2C and GPIO devices. Two dual color LEDs are included for status indication, and headers for an external LCD display and push buttons have been included.

Requirements	Verification
ADC Discretize analog signal from 0-5 V to 2^{16} values	• Connect pin pc10 of microcontroller to a DC power supply • Monitor output readings from ADC by supplying 0.0 V from the DC supply • Repeat previous step for 1.0 V, 2.0V, 3.0 V, and 4.0 V • Values for each voltage value should be separated 13000 - 14000 digital units apart.
MicroProcessor Perform FFT on digital data from ADC	• Connect a waveform generator's output to microcontroller's digital input • Generate 50 Hz sine wave • FFT results should indicate peaks at n*50Hz where n = 1,2,3...
Determine single fundamental in monophonic digital data	• Play an A4 on the guitar and the algorithm should find the fundamental at 440 Hz ± 10 Hz • Repeat for different notes B4, C4, D4..
Determine fundamentals of polyphonic sound at 65 Hz, 164 Hz and 196 Hz	• Play an open C chord on the guitar • Algorithm should find fundamentals at 65 ± 10 Hz, 164 ± 10 Hz, and 196 ± 10 Hz) • Repeat for different chords and vocal recordings
LEDs Microcontroller programmable	• Supply 5.0V to the LED's Vcc with 0.1 microFarad bypass capacitor • Program microcontroller to address the first LED to dataIn of the LED chain • First LED should turn on • Repeat for other LEDs on the chain

2.3.1 Signal Processing Unit Schematic

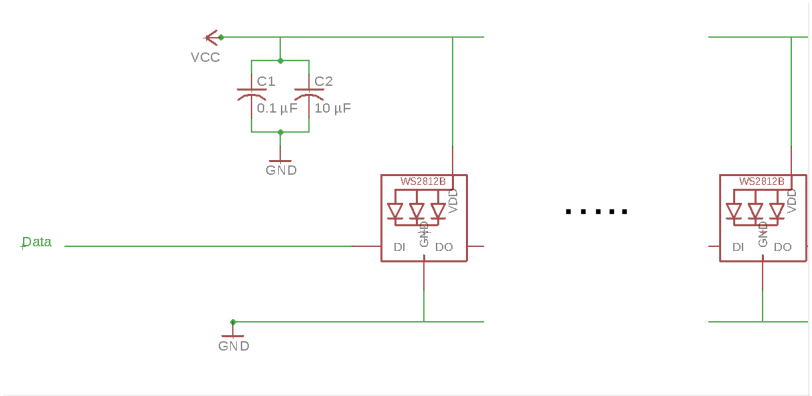

Figure 5: LED Schematic

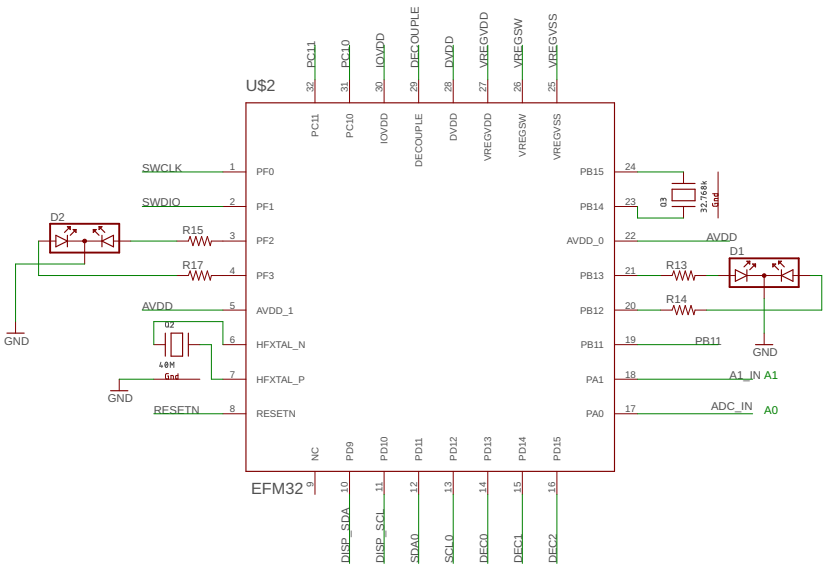

Figure 6: Microcontroller Schematic

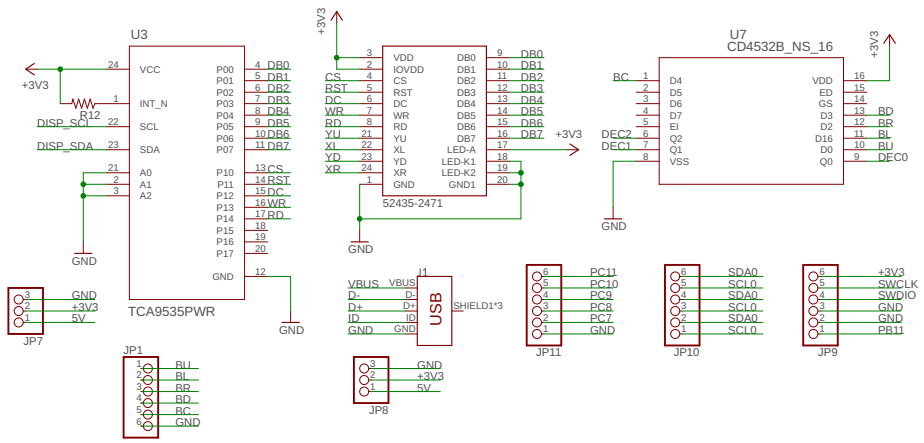

Figure 7: Peripherals Schematic

2.4 Audio Unit

The audio unit contains an electret condenser microphone and an amplifier. The microphone outputs voltage variations due to acoustic vibrations. These voltage variations are then amplified through the MAX9814 amplifier with automatic gain control. The automatic gain control controls the amplified signal so it does not surpass the max voltage output of the amplifier. This ensures the signal sent to the control/signal processing system does not surpass the 2^12 maximum value for the microcontroller. Since we care more about the frequency of our sound and not the amplitude, obtaining a clear signal is more important than capturing the true sound.

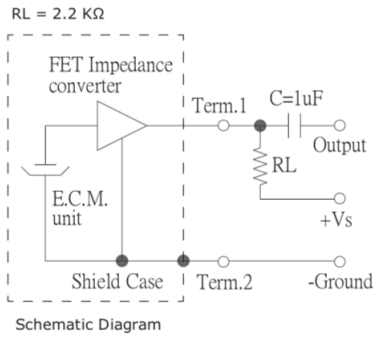

Figure 8: Microphone Measurement Circuit

Requirements	Verification
MEMS Mic	Using the measurement circuit in figure 8, Supply 3.3 VDC to + Vs, Generate a -10.0 dB sound Monitor the output voltage modulation Repeat for -15.0, -20.0 dB
Amplifier	Attach the amplifier to the MEMS Mic, and monitor the output voltage of the amplifier

2.4.1 Audio Unit Schematic

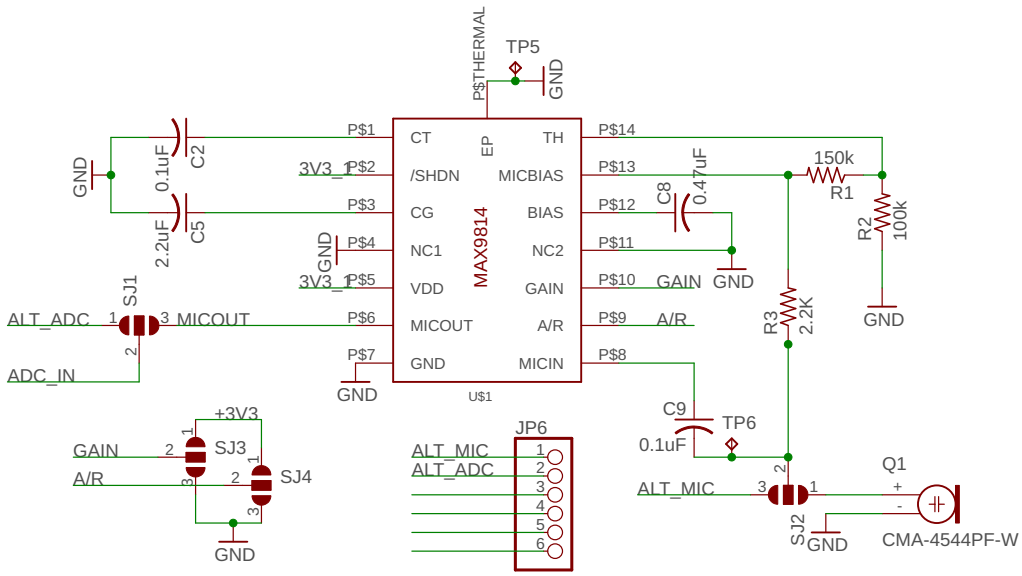

Figure 9: Audio Unit Schematic

2.5 Algorithm

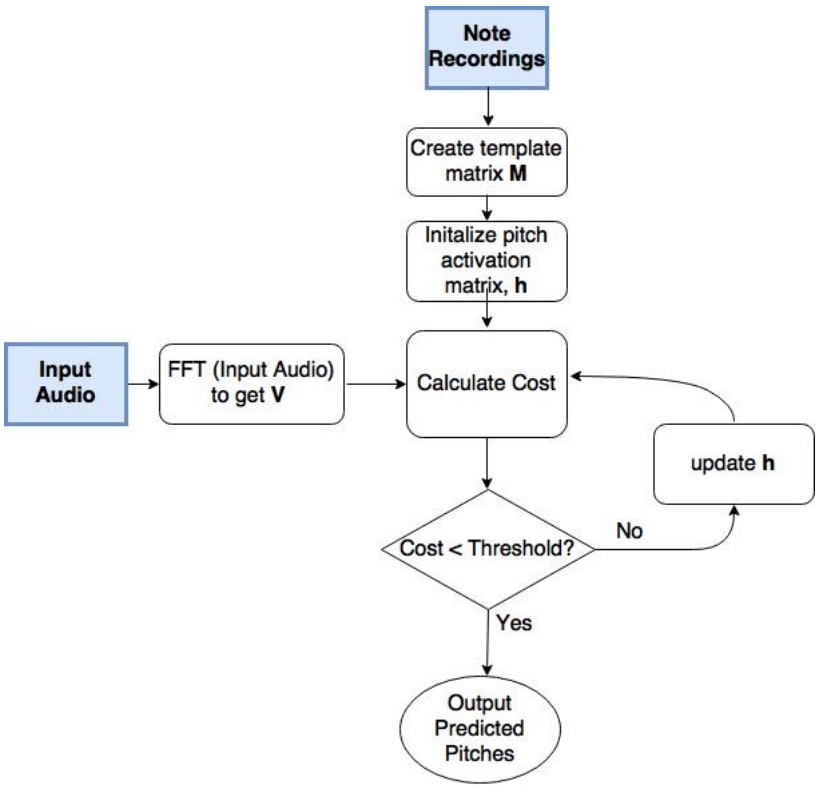

Figure 10: NMR algorithm[4][5]

2.6 Tolerance Analysis

The most critical aspect of our project is its ability to identify the note of the input waveform. The waveform travels through the microphone, amplifier, and the algorithm running on the microprocessor before the note is detected. Noise from the environment and the circuit will affect the waveform as it enters the microprocessor. For the tolerance analysis, we will investigate how the input sound quality affects the error rate of the monophonic pitch detection algorithm.

2.6.1 Input Signal SNR

The goal of this analysis is to determine the lowest SNR of the input signal such the average detection accuracy is above 80 percent for monophonic inputs.

To determine the SNR our algorithm can handle, we added a gaussian noise

to a prerecorded input signal. A signal's power is calculated by equation 1.

$$SNR = 10log_{10}\frac{P_{signal}}{P_{noise}} \tag{1}$$

SNRs of 0, 10, 20, 30, 40, 50, 60 dB were used in this analysis. The test was ran on thirty recorded guitar pitches. The average detection accuracy is the percentage of correctly identified pitches.

SNR	**Detection Accuracy**
60 dB	93.33 %
50 dB	86.67 %
40 dB	50 %
30 dB	16.67 %
20 dB	10 %
10 dB	6.67 %

Table 1: Tolerance Analysis on SNR

It is determined that increasing input signal noise affects the pitch detection accuracy. Above eighty percent accuracy occurs when the SNR is around 50 dB, and above 90 percent accuracy occurs when the SNR is around 60dB. The maximum noise generated from the circuit is first determined to ensure the algorithm is robust to circuit noise.

Audio Components	**Noise Levels**
CMA ECD Mic	60 dBA
MAX9814 Amplifier	61 dB
ADC Error	0.6 mV

Table 2: Circuit Component Noise Levels

The SNR for the microphone was given in dBA which needs to be converted to decibels. The maximum attenuation from the A-Weighted filter occurs at low frequencies. For this project, the lowest frequency produced by a guitar and voice is 80 Hz [6]. The worst case SNR in dB was calculated by observing the A weighted filter spectrum given by the equations [7] below.

$$R_A(f) = \frac{12194^2 \cdot f^4}{(f^2 + 20.6^2) \quad \sqrt{(f^2 + 107.7^2)\,(f^2 + 737.9^2)} \quad (f^2 + 12194^2)} \tag{2}$$

$$A(f) = 20\log_{10}(R_A(f)) + 2.00 \tag{3}$$

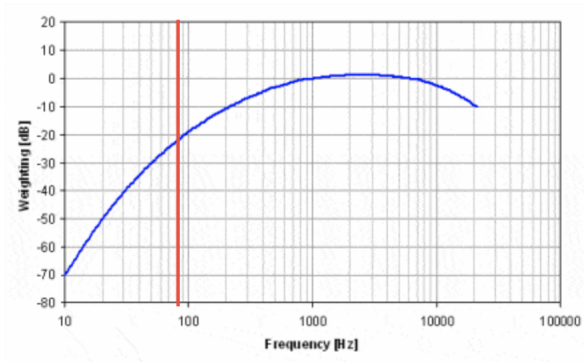

Figure 11: A Weighted Filter Response[7]

In the bandwidth of interest (beyond 80 Hz), the A filter attenuates the original signal by a maximum of 22.5 dB at 80 Hz. 22.5 dB is subtracted from the SNR of 60 dBA to obtain the SNR in dB, which is 47.5 dB.

$$dBA[f] = dB[f] + A_{filter}[f] \tag{4}$$

The components noise levels are summed together to find the typical noise generated from the circuit. SNR is on a log scale so the voltage values must be recalculated and summed together before recomputing the total SNR.

$$SNR_{total} = 10log_{10}(\sum_{i=1}^{n} 10^{\frac{SNR_i}{10}}) \tag{5}$$

The total SNR of the circuit is 60.1 dB. To ensure total SNR greater 50 dB, the environmental SNR must be greater than 59.5 dB.

2.6.2 Speed Performance

The factors affecting the system's input to output time are the FFT hop length,algorithm run-time, frequency resolution and signals traveling through the wires. This section discusses the first two factors and how they affect the speed.

FFT Hop Length The maximum delay between a note being struck and the FFT being calculated is determined by the FFT's hop length. The hop length is the number of time samples collected to perform an FFT on the new data. The length must be long enough so that the algorithm can output the predicted notes before another FFT is performed to prevent delays from data being obtained faster than can be analyzed.

$$fftdelay = \frac{N\_Hop}{Fs} \tag{6}$$

Algorithm run-time Two algorithms have been investigated so far for this project: Yin-Autocorrelation algorithm and Non-negative Matrix Factorization.

The NMR algorithm's average run-time across 90 runs of time series data is around one second in python. This can be adjusted by varying the maximum number of iterations NMF runs on each time series and by reducing the convergence criteria.

The Yin algorithm's average speed is twice the NMR algorithm. However, autocorrelation is unable to accurately identify polyphonic pitches.

Frequency Resolution Sufficient frequency resolution is necessary to distinguish different fundamental frequencies. The sampling rate Fs and the FFT window length N are the variables we set to acquire a high enough resolution. As N increases, the resolution increases proportionally but time resolution decreases. N is selected to provide a high enough frequency resolution for lowest detection error over having immediate temporal response.

Music is on a log scale, so adjacent pitches are spaced closer in frequency the lower they are. Based on the rayleigh criterion to resolve two peaks, the frequency resolution must be less than half the distance between the two peaks.

The difference between the lowest note on the guitar, E2 and its adjacent note, F2, is 4.9 Hz. The rayleigh criterion requires that the resolution be less than 2.45 Hz. The sampling rate is set to 16kHz to prevent aliasing from higher frequency harmonics which come from the highest pitch 1618 Hz (E5) of the guitar.

$$f_{rayleigh} = \frac{\triangle f}{2} \tag{7}$$

$$\triangle f_{resolution} = \frac{Fs}{N} \tag{8}$$

A large window size decreases the temporal resolution. With a sampling frequency of 16 kHz and window length of 8192, a played note can last for half a second longer than it was actually played.

Table 3: Fs and FFT Frame on Frequency Resolution

	8 kHz	***10 kHz***	***16 kHz***	***24 kHz***	***44.1 kHz***
1000	8.0 Hz	10.0 Hz	16.0 Hz	24.0 Hz	44.1 Hz
4000	2.0 Hz	2.5 Hz	4.0 Hz	6.0 Hz	11.0 Hz
5000	1.7 Hz	2.0 Hz	3.2 Hz	4.8 Hz	8.8 Hz
8000	1.0 Hz	1.3 Hz	2.0 Hz	3 Hz	5.5 Hz
10000	0.8 Hz	1.0 Hz	1.6 Hz	2.4 Hz	4.4 Hz
12000	0.7 Hz	0.8 Hz	1.3 Hz	2.0 Hz	3.7 Hz

3 Cost and Schedule

3.1 Cost Analysis

3 persons x 15 hrs/person/week x $50/hr x 13 weeks = $29,250

Parts And Labor	**Price**
Labor	$29,250
Microcontroller EFM32PG1	$3.3
CMA-4544PF-W ECD Mic	$0.82
Mic Amplifier MAX9814	$1.51
LEDs	$40.50
Batteries	$13.00
Stepdown Convertors	$6.31
Power MOSFET	$5.20
Battery Charger	$4.80
Voltage Regulator	$3.3
Button JS5208	$3.03
LCD screen LS013B7DH03	$21.62
Various Resistors, Caps, etc3	$14.00
Additional ICs and Components	$15.23
Total	$29,353.39

Table 4: Cost Table

3.2 Schedule

Time	Tasks	Ted	Chris	Jiajun
02/22	Finish the design document	x	x	x
03/02	Board Layout		x	
	Develop pitch detection algorithm on	x		x
	instrument recordings			
	Order all parts	x		
03/09	Test pitch detection algorithm	x		x
	Verify power supply components		x	
	Verify ADC, FFT, and LCD on board and LEDs	x	x	
03/16	Finalize/order PCB Design Rev1		x	
	Verify mic and amplifier			x
	Make algorithm process in real time	x		x
	Stream battery status and current instrument on LCD		x	
03/23	SPRING BREAK			
	Solder components on board	x	x	x
	Verify PCB & Debug	x	x	x
	Port algorithm to C and reduce run-time	x		x
03/30	Order final PCB		x	
	Test algorithm on hardware	x	x	x
04/06	Debug and finish remaining items	x	x	x
04/13	Finish Technical Work	x	x	x
04/20	Finish Presentation	x	x	x
04/27	Finish Paper	x	x	x

Table 5: Time Table and Task Distribution

4 Ethics and Safety

4.1 Ethics

To follow the #7 of IEEE code of ethics, "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others"[2]. We will be designing our own hardware module and software algorithms. If we come upon existing solutions that are available for us to use in our design, we will properly cite them. We will offer constructive advice to our team members and consider the advice given to us by other students, professors, and TA's to improve the project.

This project doesn't illegally use other people's musical work, so there shouldn't be any issues related to copyright infringement

The project's use case is pretty limited for personal instrument learning, we don't think there will be misuses of our project.

4.2 Safety

Li-ion batteries can explode when they are overcharged or overheated[3]. To prevent that, we need to make sure the battery is correctly charged and properly handled. The battery charging IC handles the voltages when charging, however we should always be present when handing a connected battery. We also need to make sure no water gets near the battery, and make sure the battery isn't short circuited or handled near a fire. We will monitor our battery temperature, and in software, implement emergency shutdown when the battery is overheated.

References

[1] Anssi Klapuri. *Signal Processing Methods for the Automatic Transcription of Music.* Tampere University of Technology, Tampere, Finland, 2004. Available: https://pdfs.semanticscholar.org/1970/5f4c41c2ee8411aa7b940dbedc5b864070a6.pdf

[2] IEEE.org, *IEEE Code of Ethics.* Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[3] Battery University. *Safety Concerns with Li-ion* . Available: http://batteryuniversity.com/learn/article/safety_concerns_with_li_ion

[4] John Hartquist. *REAL-TIME MUSICAL ANALYSIS OF POLYPHONIC GUITAR AUDIO* .California Polytechnic State University

[5] Arnaud Dessein, Arshia Cont, Guillaume Lemaitr *REAL-TIME POLYPHONIC MUSIC TRANSCRIPTION WITH NON-NEGATIVE MATRIX FACTORIZATION AND BETA-DIVERGENCE.*Paris, France

[6] Titze, I.R. (1994). Principles of Voice Production, Prentice Hall (currently published by NCVS.org) (pp. 188)

[7] IEC 61672-1:2013 Electroacoustics - Sound level meters - Part 1: Specifications. IEC. 2013.