#71 Digital Theremin with LED

ECE 445 Design Document- Spring 2017

Team: Phong Che (pche3), Juan Chapa (jachapa2), Chawakorn Wichulada (wichula2)

TA: Jose Sanchez Vicarte

2/24/2017

1 Introduction

1.1 Objective

The Theremin is a musical instrument that a player can play without ever having physical contact between the player and the device. It requires two hands to control the pitch and volume by changing the charge in its antennas. The antennas work the same way as a capacitor. After its invention, another form of Theremin was released, which is known as the Digital Theremin. Digital Theremin works differently because it receives an input and converts the input to digital format by sampling the input before processes the input by using digital signal processing techniques. After that, it converts the digital output to an analog signal before it goes to the speakers.

We believe that a lot of manufacturer of Digital Theremin are focusing on the wrong aspect. After examining current Digital Theremin in the market, we notice that manufacturers push for fancy features or sounds rather than improving the quality of the sound itself and striving for real innovation to the product. Therefore we proposal to implement a digital processing algorithm and apply some other signal processing technique which will improve the sound by making it sound closest to the original Analog Theremin. Furthermore, we would like to have the foot control a pedal in order to modify the volume instead of using a hand. In doing so, we can now use both hands to play two unique notes simultaneously. Another unique feature we would like to add is having a specific LED light up to represent the note being played. This will allow viewers to follow along and provide another form of entertainment besides sound.

1.2 Background

The Theremin is a classic electronic instrument made in the 1920s. It is a dynamic instrument with a unique sound and an even more unique way to play the instrument. The Digital Theremin needs more rich and distinct sounds with broader frequencies and dynamics. In ECE 395, one of our group members has created a working prototype Digital Theremin (a sensor, audio output, and microcontroller). However, it is far from sounding instrumental and lacks unique features. A company called Moog has already put a Digital Theremin on the market. Moog design focused on very different aspects than our group priority in innovating the instrument. Their Theremin focuses more on a lot of sound effects as well as automatic pitch correction.

We will revise the design of the Theremin to have two distance sensors for detecting the left and right hand's distance away from the sensors. In doing so we can play two different notes at the same time which opens to a wider range of different sounds it can play. To further that range, we will also use a waveform generator to transform the square wave to other waveforms like Triangle, Sine, and Sawtooth waveforms. Not only do we want it to expand the range of the sound, we want to provide a smoother volume control to the instrument. By adding the foot pedal that adjusts the sound, it achieves a smoother volume control. This mechanic is found in a piano for the same purpose. Lastly, we all agree it would be fun to see the notes being played through the display of LED array manipulation.

1.3 High-Level Requirement

Quantitative characteristics this project must exhibit the followings:

- It can produce frequencies within the range 65.41-261.63 Hz. ▯
- It can play square, sawtooth, triangle, and sine waveforms. ▯
- The LED array accurately display the different notes being played. ▯

2 Design

2.1 Block Diagram

In order for this instrument to operate successfully, it must include a power module, control module, sensor module, waveform module, and LED module as depicted in Figure 1. The Power module supplies voltages to other modules in order for them to perform their operations. The Sensor Module is the one that perceives information about the player's hand position relative to the sensors. The information is then processed by the control module and generates a waveform to be played. The waveform module will manipulate and utilize the waveform in order to generate a sound representation of the note to be played. The LED module is the visual representation of the note to be played by lighting the corresponding LED representing one of the notes in the octave.

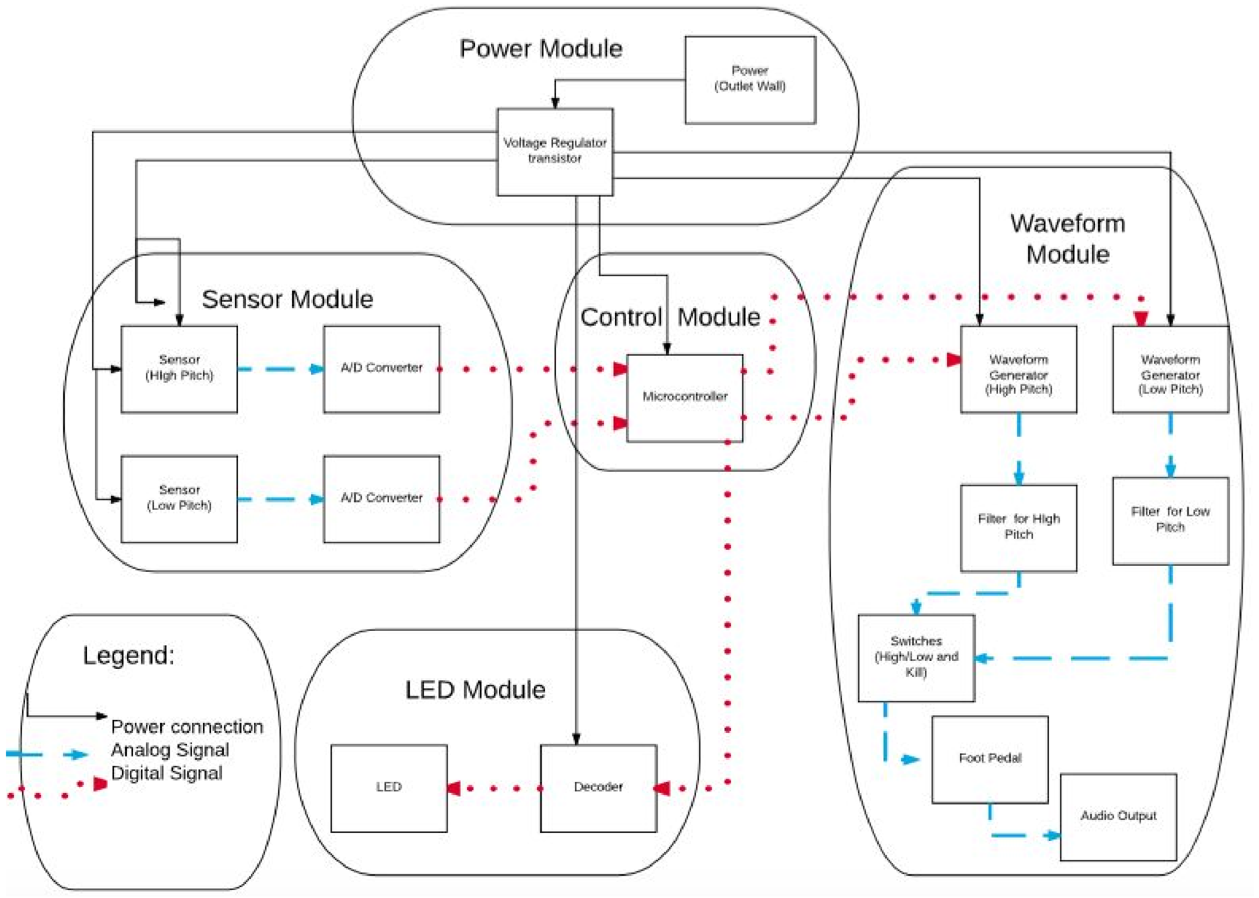

Figure 1: Block Diagram

2.2.1 Infrared Sensors

We will use two infrared Sensors for our instrument. The infrared sensors are Sharp GP2Y0A21YK0F sensors. They work like the antennas of original the Theremin. Each sensor detects the distance between the sensor and each of player's hand in order to generate high and low pitches. Both infrared sensors require 5 volts to function and output analog voltages vary between -0.3 to 0.3 volts. They accurately measure distinct distances from 10 to 80cm. Hardware-wise, both sensors are identical. However, one sensor will send data for generating higher pitches while the other sensor sends data for generating lower pitches. The pitches being played will be determined by the microcontroller. One sensor will control the higher frequencies of 130.81 to 523.25 Hz (C3 to C5 or two octaves) while the other sensor controls the lower frequencies of 65.41 to 261.63 Hz (C2 to C4 or two octaves). Notice that there is overlap in the frequency ranges.

Requirement	Verification

1. The sensor must be able to measure distances between the sensor and player's hand from 10 to 80 cm. with 8% of acceptable deviation 2. Update period is approximately 30 ms with 10% of acceptable deviation	1. A ruler will be used to measure distances from the sensor while changes in output voltage will be tested with a voltmeter. Voltage fluctuations must detected at desired maximum and minimum distances. 2. The update period will be tested using an oscilloscope to measure the period between changes in information.

2.2.2 12-bit Analog/Digital Converters

The Analog/Digital converters will convert analog signal from each infrared sensor to digital signal, which will be feed to the microcontroller because the microcontroller can only take digital information. It is designed to read input voltages in the range -0.3 +/-0.1% to 0.3 +/-0.1% volts, which makes it perfect for using alongside the distance sensors. The reason our group decided to convert the analog voltage into 12 bits is because 12 bits is the standard amount of bits used for MIDI. With 12 bits, MIDI can accurately represent thousands of distinct tones, which is perfect for this project because our digital theremin needs to be able to play thousands of semitones. A semitone is a tone that is in between two notes that are a half-step apart. If two neighboring keys on a piano represent two notes a half step apart, a semitone is a frequency that is in between those of the neighboring notes. The reason including semitones is important for this project is because a theremin gradually raises and lowers pitch until the next note is reached. It doesn't abruptly shift notes when a certain distance is sensed. This is one aspect that gives it its characteristic sound. With 12 bits, plenty of semitones can be represented without causing any major latency issues due to extended periods of processing time. The TLC2543C has a maximum delay time of 2.2 microseconds, which should cause no latency problems.

Requirement	Verification
1. The output signal of the ADC is 12 bits in length 2. The ADC has a delay time of no more than 3 microseconds with 20% of acceptable deviation	1. Check datasheet to ensure that the ADC outputs 12 bits 2. The amount of bits and delay time will be verified using a software such as modelsim to analyze the output.

2.2.3 Microcontroller: LPC1114FN28/128

The microcontroller is the main component for processing information because it has to process information from the two sensors in order to generate correct pitches, then choose a correct LED to light up based on current pitch. The “Algorithm for Microcontroller” section below explains how the algorithm computes the pitches to be played as well as how to determine which LEDs have to light up. Our criteria for finding a microcontroller were simple. We needed one with an adequate amount of general purpose input/output pins. In our case, we need at least 14 GPIO pins. The microcontroller also has to run our algorithm with a very minimal probability of creating delayed responses between the distance readings and the note being heard from the speaker. The LPC1114FN28 microcontroller is a 32 bit microcontroller with 22 GPIO pins and 4kB of SRAM. These specs make it extremely unlikely that our algorithm will cause excessive delay of the sound being produced.

Requirement	Verification
1. The microcontroller runs the algorithm quick enough that the sound being produced from the speaker changes no later than 0.15 seconds after distance reading changes with 0.5% of acceptable deviation	1. An oscilloscope will be set up to read signals from the Sharp GP2Y0A21YK0F sensor as well as the audio output after the foot pedal. The difference in start and stop times in the signals can then be measured. 2. Modelsim will be used to time the digital signal generated by the analog/digital converters as well as time the digital outputs of the microcontroller that lead to the waveform generators.

2.2.4 Waveform generators

Waveform generators will be used to generate different kinds of wave because the microcontroller can only produce square waves. However, classical musical instruments can generate many different kinds of waveform. There will be two waveform generators for each low and high pitch. This component will make the instrument be able to generate various kinds of sound because each type of waveforms can make different sound.

Requirement	Verification
1. The waveform generator must be able to produce sine wave, triangle wave, and sawtooth wave. 2. Maximum output amplitude should be 3	1. Test the component with an oscilloscope to see what waveform can this component produce. 2. The maximum amplitude can be tested by

+/- 5% volts peak-to-peak because we want to let it output to an audio jack.	using an oscilloscope. Make sure that if supply voltage is 3 +/- 5% volts, then the peak amplitude should be 3 +/- 5% volts also.

2.2.5 Decoder/Demultiplexer

This component will decode radix-2 data line to select which LED to light up. We need the decoder because we can reduce data line to be 4-bit data line instead of 12-bit data line. This way, the microcontroller does not have to waste its pins. Only one LED will be lit up at a time, so decoder is a good choice. The supply voltage is from -0.5V to 7V which is enough for lighting up each LED.

Requirement	Verification
1. It has to be able to decode to select at least 12 different output. The closest minimum number of outputs is 16 because 4-bit data line represents 2^4 = 16 different possible values. 2. The outputs have to have at least 3 +/- 5% volts in order to supply each LED.	1. The outputs and the selector can be tested by constructing a simple LED circuit. Make sure that LED lit up corresponds to the selector. 2. The maximum supply voltage is 7 volts, so 3 +/- 5% volts should work perfectly. It can be tested by using voltmeter to see if the supply voltage is 3 +/- 5% volts, do outputs also have 3 +/- 5% volts.

LED number	Note	Frequency range (Hz)
1	C	130.81 - 141.71 OR 261.63 - 283.43
2	C# / Db	141.71 - 152.61 OR 283.43 - 305.23
3	D	152.61 - 163.52 OR 305.23 - 327.04
4	D# / Eb	163.52 - 174.42 OR 327.04 - 348.84
5	E	174.42 - 185.32 OR 348.84 - 370.64
6	F	185.32 - 196.22 OR 370.64 - 392.44
7	F# / Gb	196.22 - 207.12 OR 392.44 - 414.24

8	G	207.12 - 218.02 OR 414.24 - 436.04
9	G# / Ab	218.02 - 228.93 OR 436.04 - 457.85
10	A	228.93 - 239.83 OR 457.85 - 479.65
11	A# / Bb	239.83 - 250.73 OR 479.65 - 501.45
12	B	250.73 - 261.63 OR 501.45 - 523.25

Figure 2: Loop-up Table for decoder

2.2.6 LEDs

LEDs will display which note that current player is playing. The note corresponds to the higher pitch (right hand) only. There are 12 different notes in one octave. Therefore, we need 12 LEDs. Seven of them are green color to represent normal notes, and the remaining 5 of them are red color to represent sharp and flat notes. The required voltage for each LED to light up is 3 volts.

Requirement	Verification
1. They have to light up and be at least visible 3 +/- 5% meters away	1. This can be tested by lighting up the LED and observe its intensity from that distance for visibility.

2.2.7 Algorithm for the microcontroller

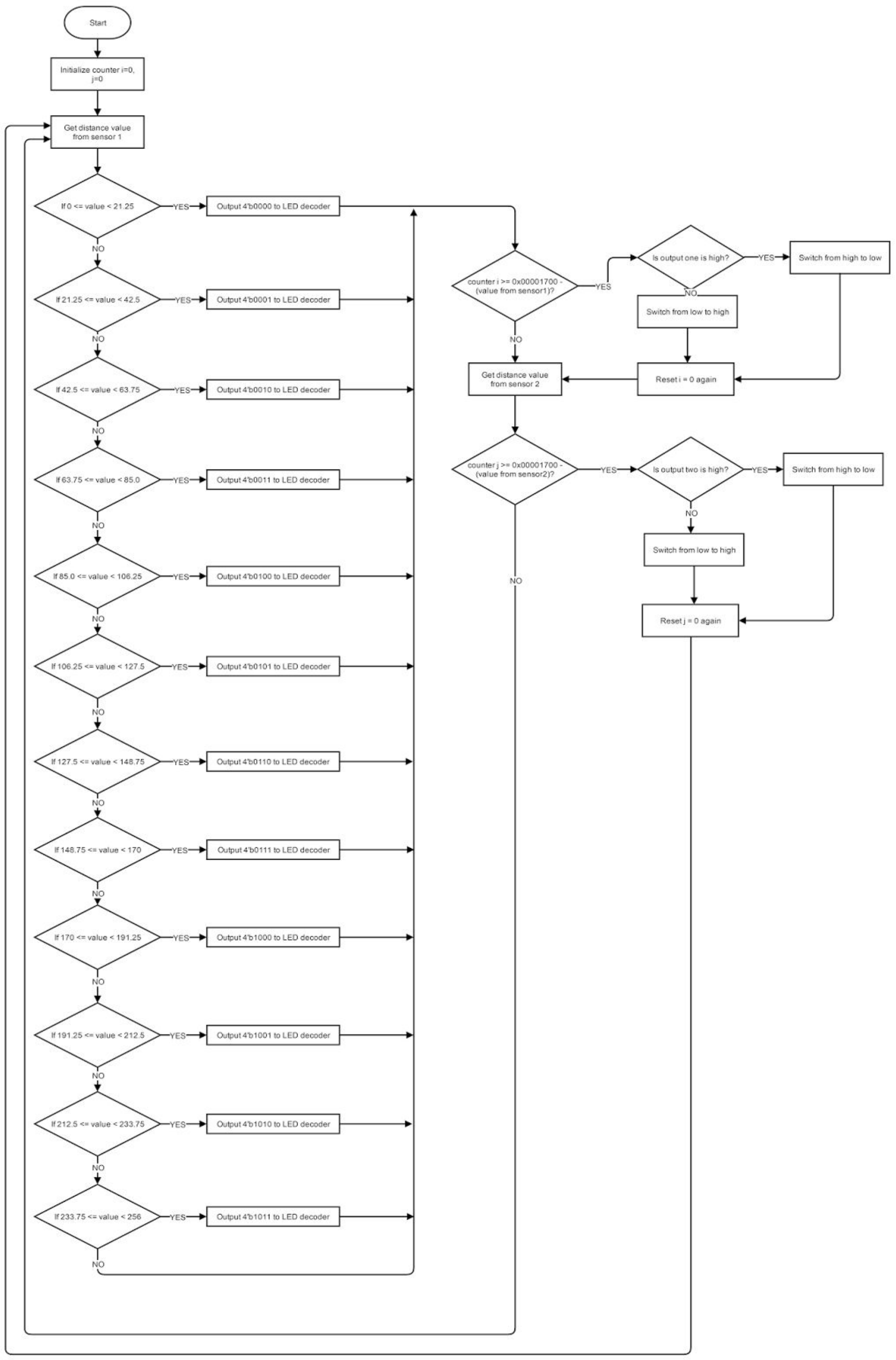
Start
Initialize counter i=0, j=0
Get distance value from sensor 1
If 0 <= value < 21.25
YES
Output 4'b0000 to LED decoder
NO
If 21.25 <= value < 42.5
Output 4'b0001 to LED decoder
If 42.5 <= value < 63.75
Output 4'b0010 to LED decoder
If 63.75 <= value < 85.0
Output 4'b0011 to LED decoder
If 85.0 <= value < 106.25
Output 4'b0100 to LED decoder
If 106.25 <= value < 127.5
Output 4'b0101 to LED decoder
If 127.5 <= value < 148.75
Output 4'b0110 to LED decoder
If 148.75 <= value < 170
Output 4'b0111 to LED decoder
If 170 <= value < 191.25
Output 4'b1000 to LED decoder
If 191.25 <= value < 212.5
Output 4'b1001 to LED decoder
If 212.5 <= value < 233.75
Output 4'b1010 to LED decoder
If 233.75 <= value < 256
Output 4'b1011 to LED decoder
counter i >= 0x00001700 - (value from sensor1)?
Is output one is high?
Switch from high to low
Switch from low to high
Reset i = 0 again
Get distance value from sensor 2
counter j >= 0x00001700 - (value from sensor2)?
Is output two is high?
Switch from high to low
Switch from low to high
Reset j = 0 again

Figure 3: Algorithm of microcontroller

The flowchart describes the algorithm which controls LEDs and frequency of the output square wave. The output for LEDs will be encoded in radix 2. Since we have 12 LEDs, the decoder will take 4-bit wide selector because it has to be power of 2 which 2^4 = 16 is enough for 12 distinct LEDs. We chose to have the index of LEDs from 0 to 11, so the output to decoder is as follows:

· output[3:0] = 0000 for lighting up the first LED	(C)
· output[3:0] = 0001 for lighting up the second LED	(C# / Db)
· output[3:0] = 0010 for lighting up the third LED	(D)
· output[3:0] = 0011 for lighting up the fourth LED	(D# / Eb)
· output[3:0] = 0100 for lighting up the fifth LED	(E)
· output[3:0] = 0101 for lighting up the sixth LED	(F)
· output[3:0] = 0110 for lighting up the seventh LED	(F# / Gb)
· output[3:0] = 0111 for lighting up the eighth LED	(G)
· output[3:0] = 1000 for lighting up the ninth LED	(G# / Ab)
· output[3:0] = 1001 for lighting up the tenth LED	(A)
· output[3:0] = 1010 for lighting up the eleventh LED	(A# / Bb)
· output[3:0] = 1011 for lighting up the last LED	(B)

It will first get detected value from the first sensor (for the right hand), then it will go to switch-case statement to choose which LED to light up. Then it will determine how long to wait until switch the state (high to low/low to high) of the square wave output for the higher pitch. This will alter frequency of the output wave. After it is done with the first sensor, it does the same with the other sensor (for left hand) by getting a value from the sensor. After that, it determines how long it should wait until it switch the output state (for left hand) in order to alter the frequency of the square wave that come from the left hand. After that, it goes back to the first process (the first sensor) again.

2.2.8 Schematic

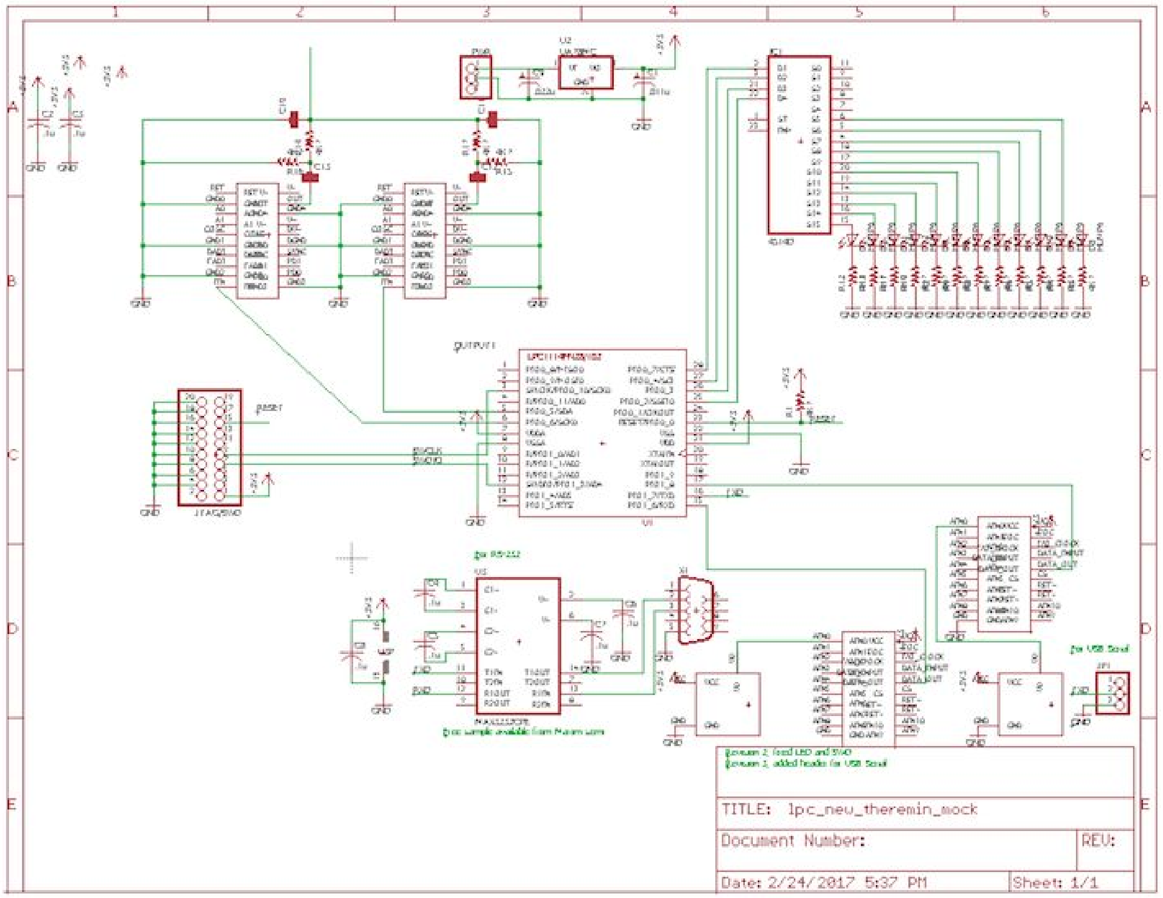
Revision 2, fixed LED and SWD
Revision 1, added header for USB Serial
TITLE: lpc_new_theremin_mock
Document Number:
REV:
Date: 2/24/2017 5:37 PM
Sheet: 1/1
JTAG/SWD
for USB Serial
for RS232

Figure 4: Overall schematic

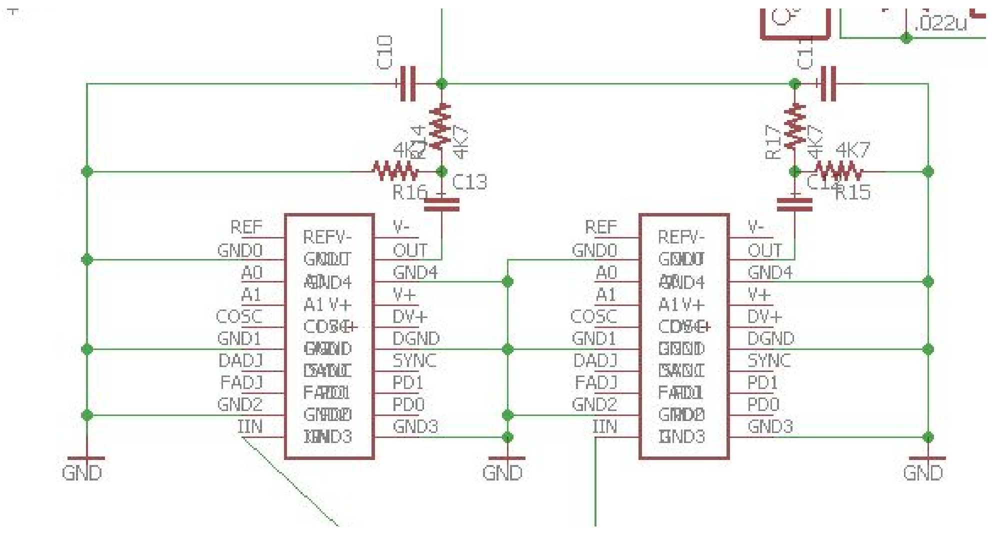
Figure 5: Waveform generator and bandpass filter (Waveform Module)

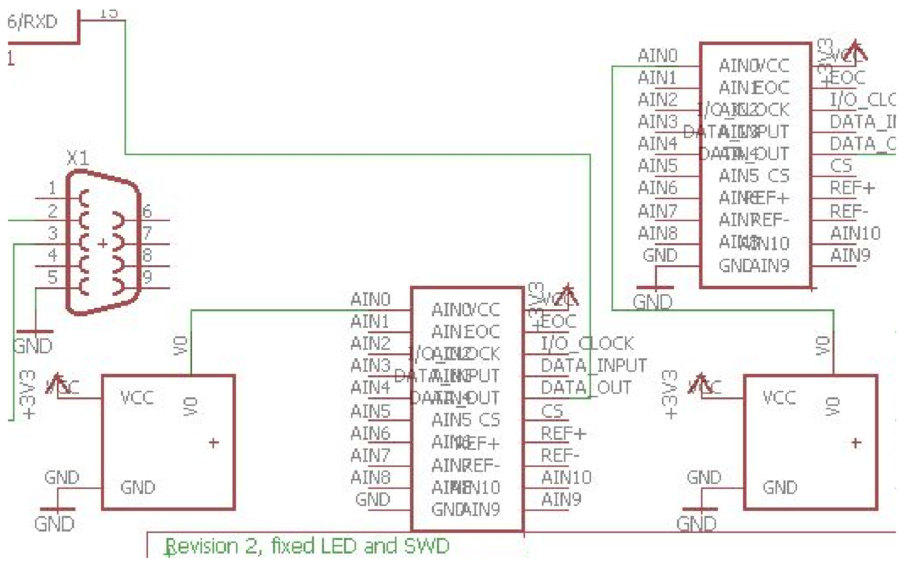
Figure 6: Sensor and ADC (Sensor Module)

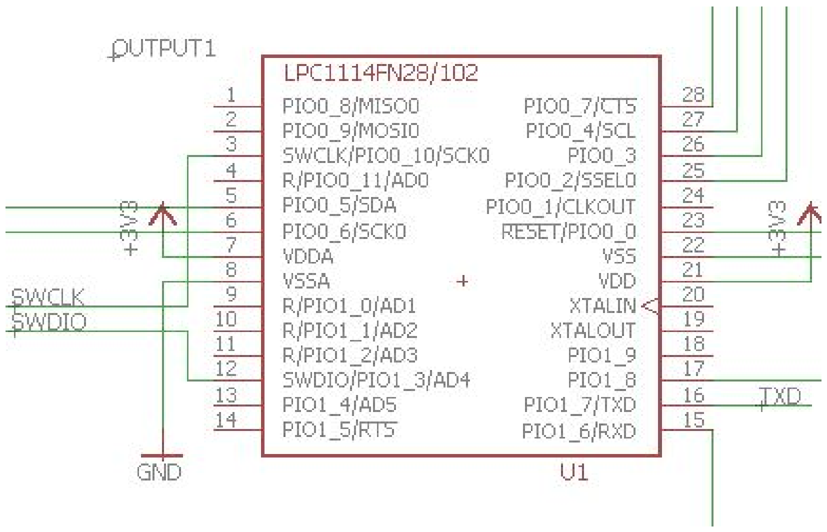

Figure 7: Microcontroller (Control Module)

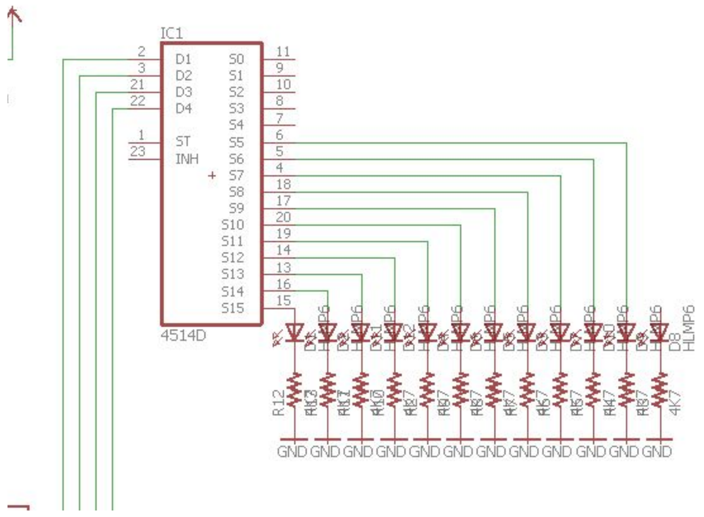

Figure 8: Decoder and LEDs (LED Module)

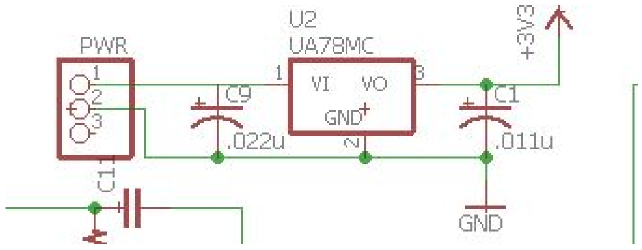

Figure 9: Voltage regulator (Power Module)

2.3 Tolerance Analysis

An important aspect of the infrared sensors to explore is the sensitivity of the sensors. The total voltage range of the sensor output is:

0.3-(-0.3) = 0.6 volts

This means that to accurately represent 12 bits, the sensors must be able to produce 4096 distinct voltages, which requires quite a bit of precision. This means that in the 0.6 volts range, the difference in voltage between two consecutive unique distance readings is:

0.6/4096 = 0.000146484375 volts = 0.14648 mvolts

An important aspect of the analog to digital converters to explore is the amount of individual frequencies that can be played with 12 bits. With 12 bits of information, 4096 unique frequencies can be represented. One octave has 12 different notes (seen below), so to play every single note requires representing at least 24 distinct pieces of information. However, the goal is to transition from note to note as smoothly as possible, which means adding semitones. The goal then becomes to represent as many notes as possible. This is essentially done by adding enough semitones that the human ear can't hear a difference between the different states. This can be done by first choosing the number of bits used to represent distinct distances and therefore distinct frequencies. MIDI uses 12 bit information since anything below 12 bits would make it difficult to create smooth transitions between notes as they rise and fall. Below are calculations for finding distinct semitones with 12 bits of information and 8 bits of information respectively.

Equation 1: Resolution Calculation

Bit representation / 24 - 2 = resolution

4095/24 - 2 ~ 167	With 12 bits representation
256/24 - 2 ~ 9	With 8 bits representation

The -2 subtracts the two real tones.

Equation 2: Find frequency range

Frequency range = Fmax - Fmin

523.25 - 130.81 = 392.44 Hz
261.63 - 65.41 = 196.22 Hz.

Notice that frequency span of the higher octaves is twice that of the lower octaves, yet they both represent the exact same amount of notes. The reason for this is simply due to the way human perceive hearing. As you can see, the frequency range for the higher frequencies is twice that of the lower frequencies, which is characteristic of the logarithmic scale that models human hearing. The way in which humans perceive hearing is modeled by the following equation:

Equation 3: Volume amplitude respect to voltage

$$N_{dB} = 10\ \log_{10}\left(\frac{V2^2}{V1^2}\right)$$

This will be important to model for our algorithm. NdB is the volume amplitude in decibels. V2 is the voltage amplitude of the output while V1 is a reference voltage many of the components must be set to.

One important aspect of the microcontroller is that both outputs must be in the form of clean, uniform square wave such as in figure 10.

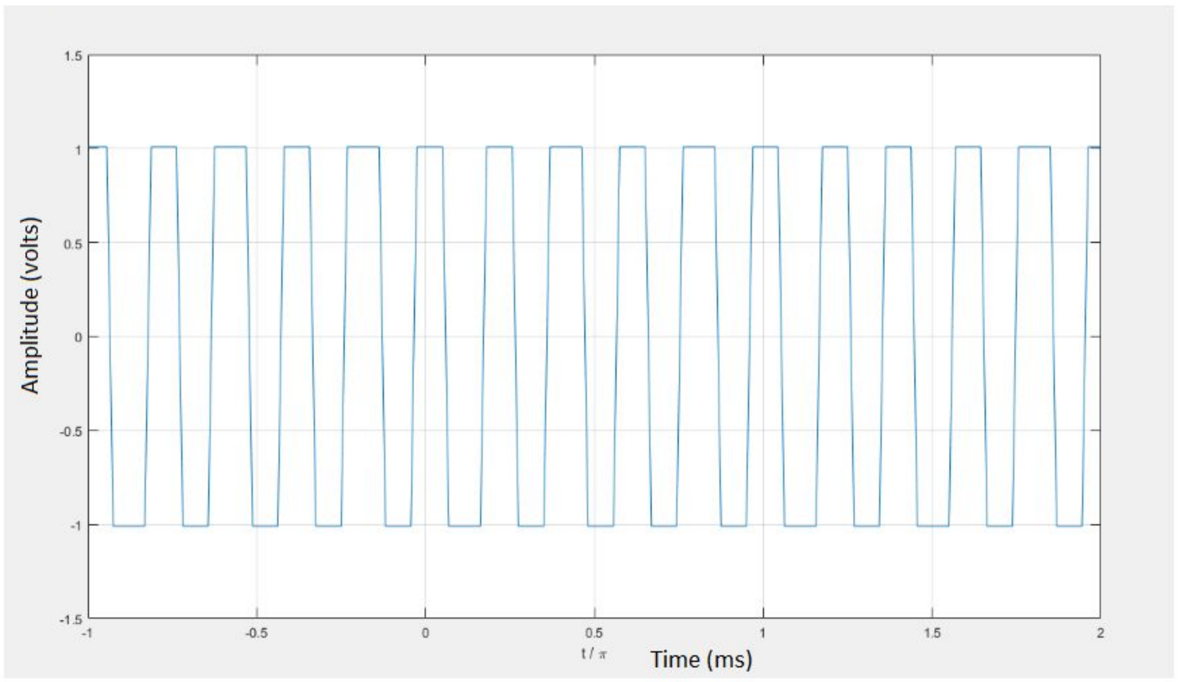

Figure 10: Middle C note (261.63 Hz) square wave

However, by the end of ECE 395, the output of the microcontroller was something completely different. Figure 11 is an oscilloscope image of the main output of the microcontroller from previous tests. Although the image below shows a function with a uniform period able of holding a pitch. It is far from the uniform square wave desired from the microcontroller output. This was most likely due to erroneous values from incorrectly implementing UART protocol for the distance readings. UART protocol requires a start bit and a stop bit. Since the old algorithm never properly defined where the stop and start bits were located, the resulting output was the distorted image seen in figure 11.

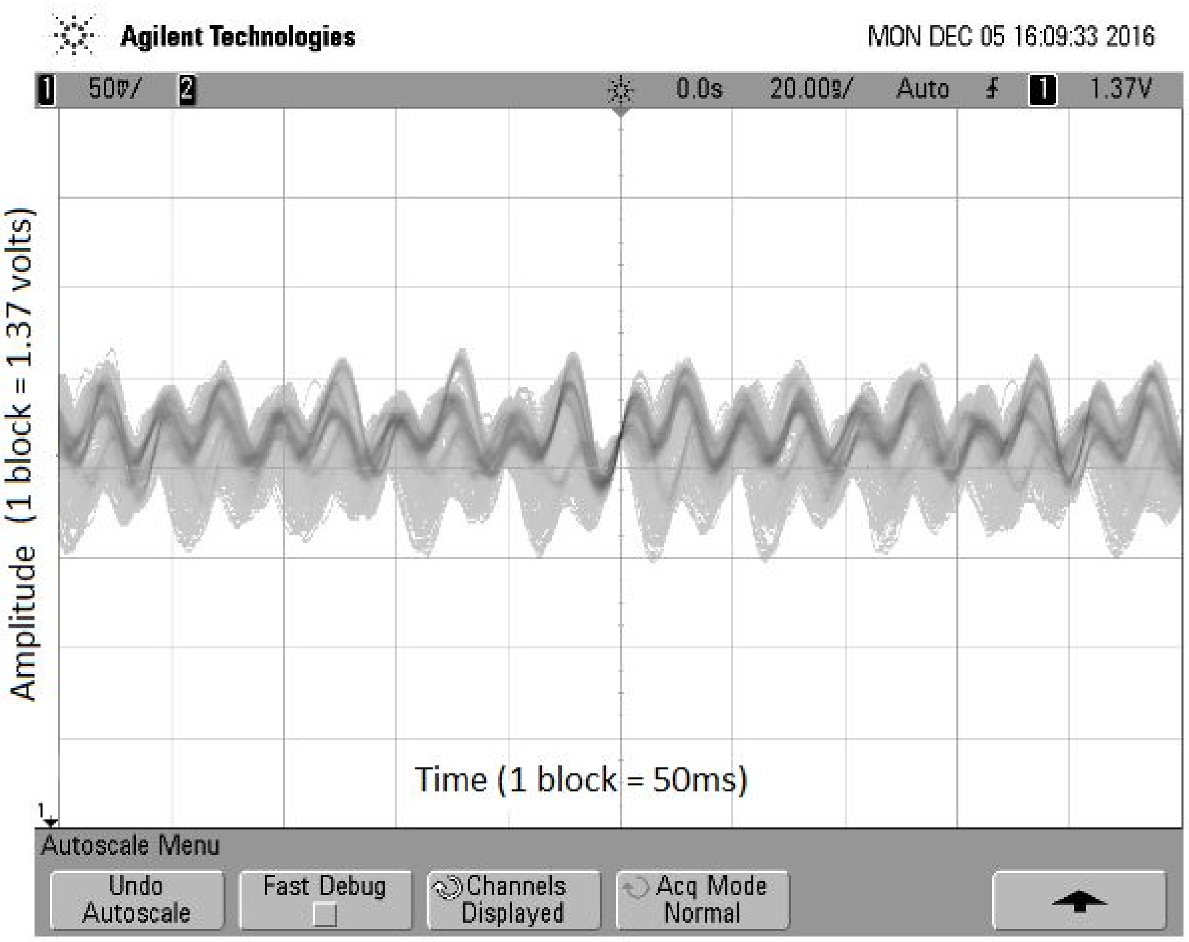

Figure 11: G sharp (103.83 Hz) distorted output from microcontroller

3 Cost and Schedule

3.1.1 Labor

The fixed cost for development would be (\$35/hour) with (10 hours/week) for each member on the team. There are three people on the team working over the course of the semester mainly (13 weeks)

Equation 4: Labor Cost

Labor = 3 people * \$35/hour * 10 hours/week * 13 weeks * 2.5 = \$34,125

3.1.2 Parts

Quantity	Part	Cost of single unit	Cost of single in bulk (100)
2	Infrared Sensors (Sharp GP2Y0A21YK0F)	\$9.95	\$6.95
2	A/D converter (TLC2543C)	\$9.87	\$7.37
1	Microcontroller (LPC1114FN28/128)	\$27.89	\$10.00
1	Decoder (generic 4 to 16)	\$3.25	\$2.75
12	LEDS (green, red)	\$0.15	\$0.12
3	Toggle switches (generic)	\$1.55	\$1.15
2	Waveform Generator (MAX038)	\$15	\$11.00
2	Filter (Resistors, capacitor)	\$1.00	\$0.60
1	Foot Pedal (generic)	\$62.00	\$47.00
1	Audio output (generic)	\$2.27	\$1.59
1	Voltage Regulator (UA78M33)	\$0.55	\$0.30
1	Power transformer (Generic)	\$6.00	\$5.50
1	PCBs (Custom print)	\$2.00	\$1.50
	Total	\$182.05	\$125.37

Figure 12: Table of Parts costs

We estimated that the cost of the parts and manufacturing the prototype would be $200.00 each. During the development phase we will be creating at least ten prototypes.

Equation 5: Cost of Parts

$$\text{Cost of Parts} = 10 \text{ prototype} * 200/\text{prototype}$$

Equation 6: Grand Total

$$\text{Grand Total} = \text{Labor} + \text{Cost of Parts}$$

Therefore the total development cost will be estimated to $36,125.

3.2 Schedule

Week	Phong	Juan	Chawakorn
1/30/2017	Develop the basic block diagram	Prepare the project approval	Edit the project approval
2/6/2017	Design the LED Display and Control Module	Design the Sensor Module and Power Module	Design the Waveform Module
2/13/2017	Redesign block diagram	Work on foot pedal design	Prepare the Mock Design
2/20/2017	Calculate cost and create schedule	Create circuit schematics	Select specific components to use
2/27/2017	Prepare the Design Review	Assemble the first prototype	Purchase components needed
3/6/2017	Create a lookup table for LED display	Test the sensors correlation to distance	Implement algorithm for microcontroller
3/13/2017	Test LED Display lighting for each note	Test the foot pedal sound modifier	Develop the PCB for circuit
3/27/2017	Fix any bugs in	Create prototype	Test and perform

	algorithm for LED Display.	version 2	first revision on PCB
4/3/2017	Perform safety test on prototype	Run test on prototype version 2	Perform final revision and test on PCB
4/10/2017	Modify prototype to obey safety test	Create final prototype	Test that a note is being played correct
4/17/2017	Prepare for Mock demo	Run test on final prototype	Test that two notes can be played at the same time.
4/24/2017	Prepare final presentation	Design prototype case	Record a sample made by prototype
5/1/2017	Begin Final Report	Combine prototype and case to finalize product	Edit Final Report

Figure 13: Schedule of each member table

4 Discussion of Ethics and Safety

Safety is a very important factor in our implementation of the project. Throughout the development of our project, one safety risk that we will pay close attention to is the monitoring of our power source(s). It is important that as we develop our project we keep in check what voltages are going through each component to make sure that it does not overload the capacity that could result in possibly frying it or starting a fire. Beyond the development phase, we will apply a safety measure to the power module with a fuse to ensure that the current flowing through it is the correct amount to prevent the component from being overloaded and circuit being shorted.

One possible ethical issue that could arise is that we might not be aware of our product not being able to produce all the notes/pitches that we claim. According to IEEE Code of Ethics code 3, it is important that we are honest and realistic in stating claims based on the data that is available to us [1]. Therefore, to ensure that our design address that issue, we will be using a music tuner to test each pitch. We will check that all the range of sound that we claim can be achieved by our instrument.

We as a group envision to follow the IEEE Code of Ethics. As stated in IEEE Code of Ethics code 1, we will accept all responsibilities in decisions that we make to ensure the well-being and safety of the public [1]. We want our product to be as safe as possible to any user by applying safety measure listed above. Furthermore, we will be seeking feedback from many different people such as friends, families, instructors, and teaching instructors. As an engineer, under the guidance of IEEE Code of Ethics code 7, we will accept all the criticism of the product, acknowledge the errors, and correct them [1]. Our goal is to meet the expectation that others have for our product and to make it as flawless as possible. Although our product will initially be far from perfect, but with every mistake corrected we will be one step closer to reaching that goal.

5 Citations

[1] "IEEE IEEE code of ethics," in *IEEE - IEEE Code of Ethics*. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: Feb. 17, 2017.

[2] S. S, "Why learning piano theory is important for all musicians," in *takelessons*, 2014. [Online]. Available: http://takelessons.com/blog/piano-theory-for-all-musicians. Accessed: Feb. 17, 2017.

[3] "Sharp GP2Y0A21YK0F analog distance sensor 10-80cm," in *pololu*. [Online]. Available: https://www.pololu.com/product/136. Accessed: Feb. 17, 2017.

[4] "12-BIT ANALOG-TO-DIGITAL CONVERTERS WITH SERIAL CONTROL AND 11 ANALOG INPUTS," in *texas instruments*, 2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/tlc2543m.pdf. Accessed: Feb. 17, 2017.