Digital Theremin with LED

By

Chawakorn Wichulada
Juan Chapa
Phong Che

Final Report for ECE 445, Senior Design, Spring 2017

TA: Jose Sanchez Vicarte

2 May 2017
Project No. 71

Abstract

The main idea behind our project was to re-imagine the Theremin. The original Theremin is an analog instrument invented in the 1920s that is played by positioning hands in midair at specific distances from two antennae to control volume and pitch respectively. The goal of our senior design project was to change how the Theremin sounds as well as how it is played. We accomplished the task by including an LED display that shows the user what frequencies they are playing, a killswitch that creates a stuttering effect, a volume foot pedal, and waveform selection. These features are designed to ease handling of the instrument for the player, eliminate challenges the original Theremin presented, and improve upon the functionality of the instrument.

Table of Contents

1. Introduction

1.1 Purpose and Goals

The Theremin is a musical instrument that a player can play without ever having to make physical contact with the device. Changes to pitch and volume occur as a result of modifying the charges in the antennas which are manipulated by hand movement. The antennas and the user's hand act like a capacitor that produces the voltage. This voltage produces a sound that can be adjusted by increasing or decreasing the distance between the user's hand and the antennas. The problem with the original Theremin is that it is an instrument that requires a very skillful player to produce a precise pitch. Furthermore, the instrument's sound and volume control reading can be inconsistent since it can easily be influenced by external factors such as changes in humidity and temperature.

Our goal was to improve on the design of the original Theremin to address the above issues. In our design of a Digital Theremin, we removed these external factors' influence by changing the antennas to infrared sensors in order to produce more reliable distance reading. To help assist the player in playing the instrument easier, we incorporated an LED display and a more familiar volume control using a foot pedal. The LED display indicates which note is being played which ultimately gives the player feedback on the relationship between distance and the corresponding note. To expand on the functionality of the instrument, we used two sensors which allowed us to play two notes at the same time, a waveform modifier which generated unique sounds, and a killswitch which created a stutter effect.

1.2 Block Diagram

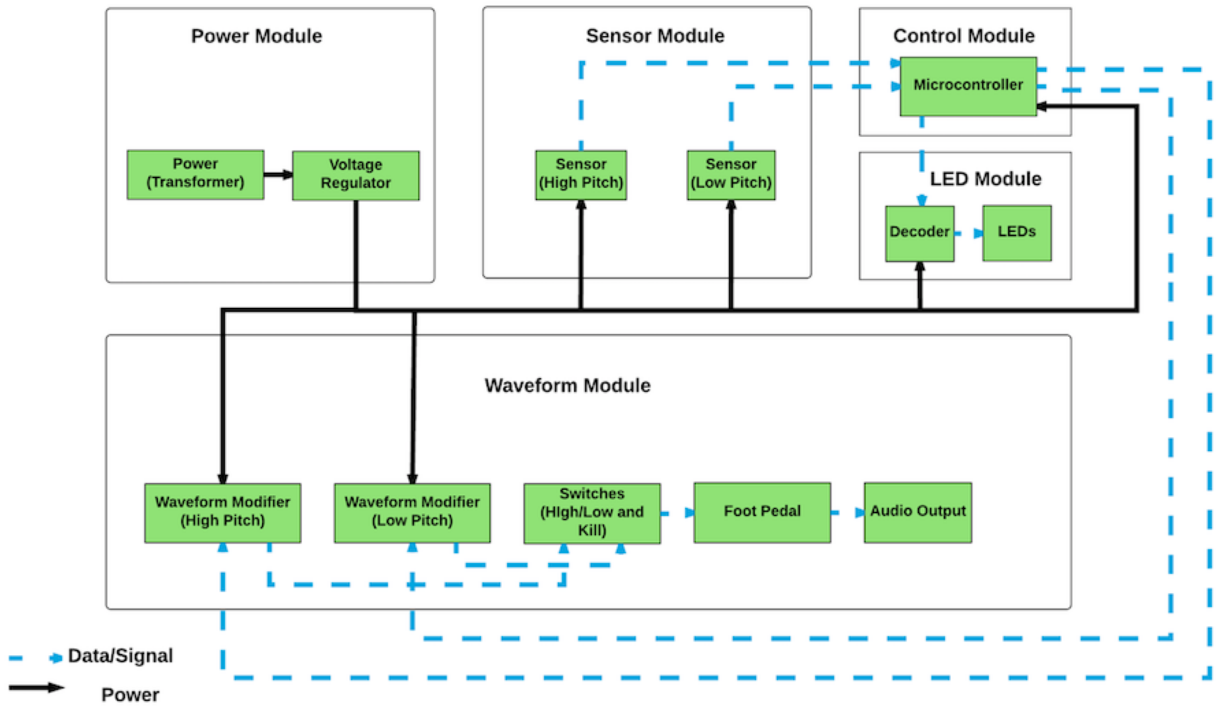


Figure 1: Block diagram of Digital Theremin

This project is divided into five blocks consisting of power module, sensor module, control module, LED module, and waveform module. The purpose of the power module is to ensure the correct and stable voltage is supply to all the other modules. The sensor module will detect up to a set distance from the player's hand to the sensor and generate a voltage reading. The voltage reading is then process by the control module using an algorithm to create a clean square waveform with a period corresponding to the voltage. Once a waveform is produced, it is pass to the waveform module which can modify the square wave into another clean waveform (sine and sawtooth) using switches and produce an audio output. The foot pedal in the waveform module is to adjust the volume of the sound by applying varying resistances using a potentiometer to the audio output signal. The LED module function as a feedback for the player by lighting up a LED corresponding to a specific note being played.

2. Design

2.1 Design Procedure

2.1.1 General design alternatives

The power module we chose to use a transformer and voltage regulator to provide stable and accurate voltages that we need for other module. In the sensor module we chose analog sensors to allow flexibility in sound resolution instead of fix resolution set by digital sensor. In our original design we included an external ADC (analog digital converter) chip to convert the analog output of the sensor to a digital input for the micro controller. However, the ADC does not meet our requirement set in Appendix A table in speed conversion therefore we utilize an internal ADC in the microcontroller instead. In the microcontroller module we chose this specific microcontroller instead of less complex ones for its computing power to avoid latency and the specific pins that can be program for our algorithm. In the waveform module we originally plan to use a waveform generator chip to produce other types of waveform for different sounds. However, the waveform generated was not clean which led to poor sound quality. In order to produce a much cleaner waveform we implement our own circuit for each waveform modifier.

2.2 Design details

2.2.1 Power module

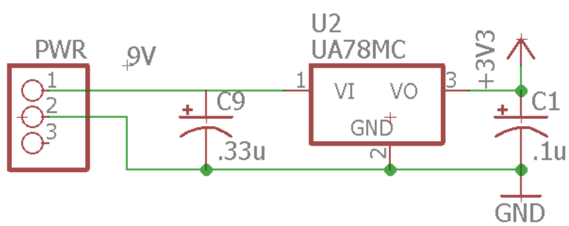


Figure 2: Schematic of the power module

The power supply for our project comes from a 9V DC power adapter that can be plugged into a standard wall plug. Within the circuit, there were electronic components that needed voltage supplies of 9V, 7V, 5V, and 3.3V. The 3.3 volts was generated by implementing the linear regulator shown in figure 2 to supply a steady supply of 3.3 volts. This is the voltage we used to supply the digital electronics in the circuit such as the microcontroller and decoder. 9 volts was used to power the op amps in the circuit. 7 volts was used to power the hex inverters. The 7 volts used for the hex inverters was achieved by using the voltage divider rule. 5 volts was needed to supply power to the distance sensors. When we tried to power the sensors using voltage divider, we ran into issues since the voltage kept fluctuating. In the process of trying to properly power the sensors, one of them burned out. For the demo, we supplied voltage to the sensor using an external voltage source. We later learned it was possible to achieve what we wanted using another linear regulator.

2.2.2 Infrared Sensor Module

We use two analog infrared sensors as our inputs. They are SHARP GP2Y0A21YK0F. They detect the distance between the sensors and the hands of the player. The distances are converted to digital values for the microcontroller to determine the period of square waves.

2.2.3 Control Module (Microcontroller)

The microcontroller was responsible for so many tasks in our design. The first task it is responsible for is reading the data output from the distance sensors. Since both sensors output analog voltages varying in amplitude based on the distance readings, the analog to digital conversion feature is used on our microcontroller. The analog voltage is converted into 12-bit digital data that can be used in the algorithm. The second task is generating the waveforms. The outputs of both pins 28 and 26 are square wave audio signals. The period of the square waves depends on the data received from the sensors. The third task carried out by the microcontroller is controlling the LEDs. This is done by the data output of pins 15 to 18 which sends 4 bits of information to the decoder, which lights up specific LEDs based on the notes from the high pitch output being produced.

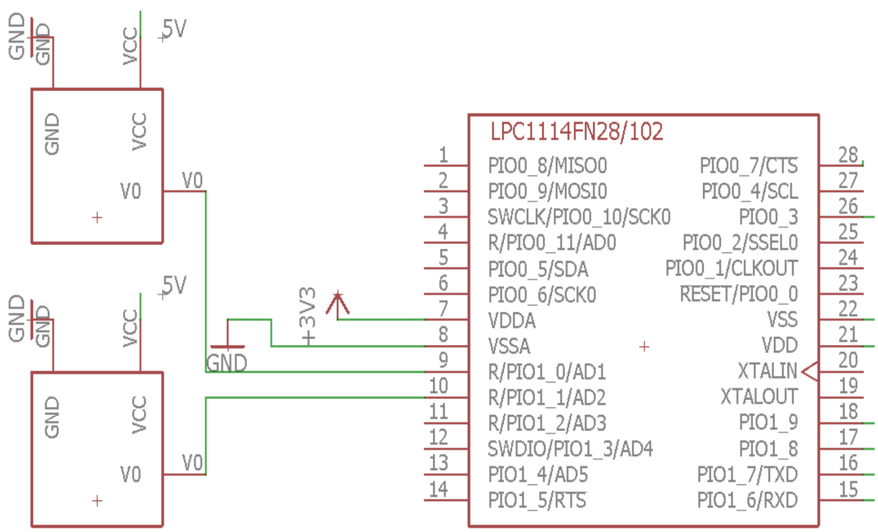


Figure 3: Schematic showing connections between sensors and the microcontroller

The following image shows the final connection of the microcontroller to the main circuit of the instrument.

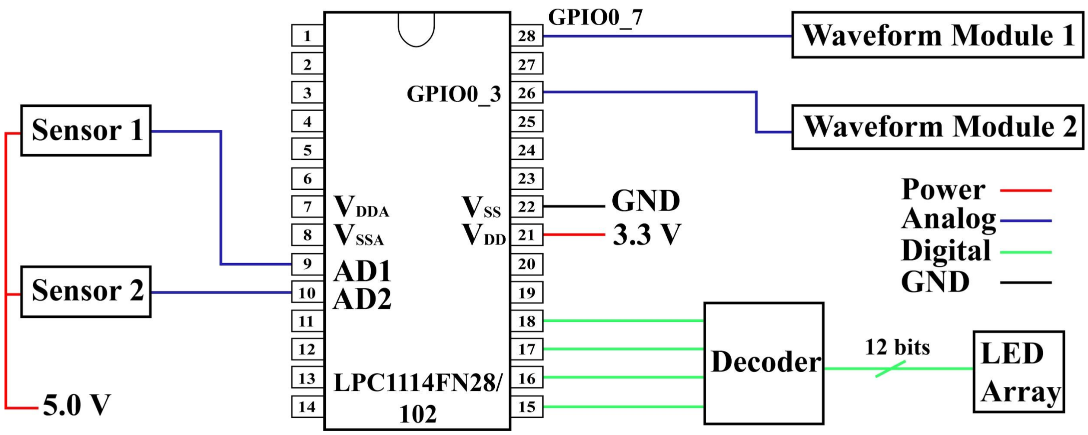


Figure 4: The final connection of the microcontroller

As you can see, we only use 6 digital general input/output pins to output the square waves (pin 26 and pin 28) and the 4-bit radix-2 value (pin 15, pin 16, pin 17, and pin 18) to the decoder. Pin 9 and pin 10 are used as analog input channels to the internal Analog/Digital converter.

2.2.4 Software design details

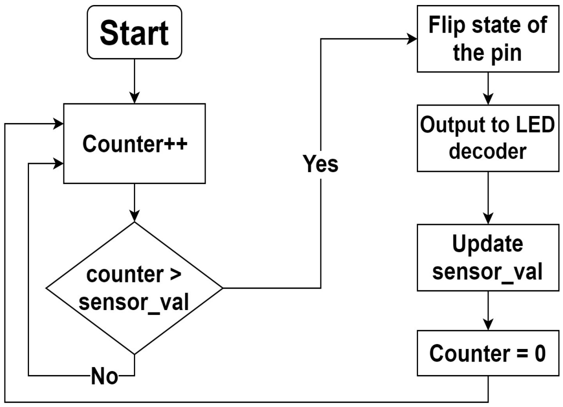
2.2.4.1 Analog/Digital Converter driver details

The first step of programming the microcontroller is to initialize the chip and relevant internal hardware, such as the internal Analog/Digital converter and pin configurations. The steps of setting up the microcontroller can be described as follows:

1. Enable power to relevant regions, which are the internal A/D converter and pin configuration hardware.
2. Enable clock to the internal A/D converter and pin configuration hardware.
3. Set up each pin that we use to the correct mode (A/D input or general input/output)
4. Enable **Burst mode** of the A/D converter **(important)**
    - This makes the A/D converter hardware to operate in the background. It converts each channel and saves the corresponding value to the channel registers. [2]
    - Software just has to retrieve from those registers.

2.2.4.2 Square wave algorithm design details

A software is used for generating square waves based on sensor values from the two sensors. The generating square waves part is simple because it just flipping the state of the output pin (from LOW to HIGH or from HIGH to LOW). The following flowchart illustrates the simplified algorithm for generating square waves for each output.




```
while true do
    counter = counter + 1
    if counter > Read_value then
        Flip state of the pin  (HIGH / LOW)
        Output to LED decoder based on Read_value
        counter = 0
    end if
end while
```

Figure 5: Flowchart and pseudocode for generating square waves

2.2.5 LED Module

The LED display circuitry takes the 4-bit output of the microcontroller and translates it into a specific LED lighting up. This is done by using 12 out of 16 of the outputs to light up the LEDs at the correct time. Since the decoder we used is active low, meaning that the bit that selects the desired LED is a low output (or 0), two hex inverters were implemented to switch the output from active low to active high. Figure 6 shows how the LED display circuit takes 4 bit inputs 0000 to 1011 and converts them into specific LEDs lighting up. Each LED represents a specific note being played.

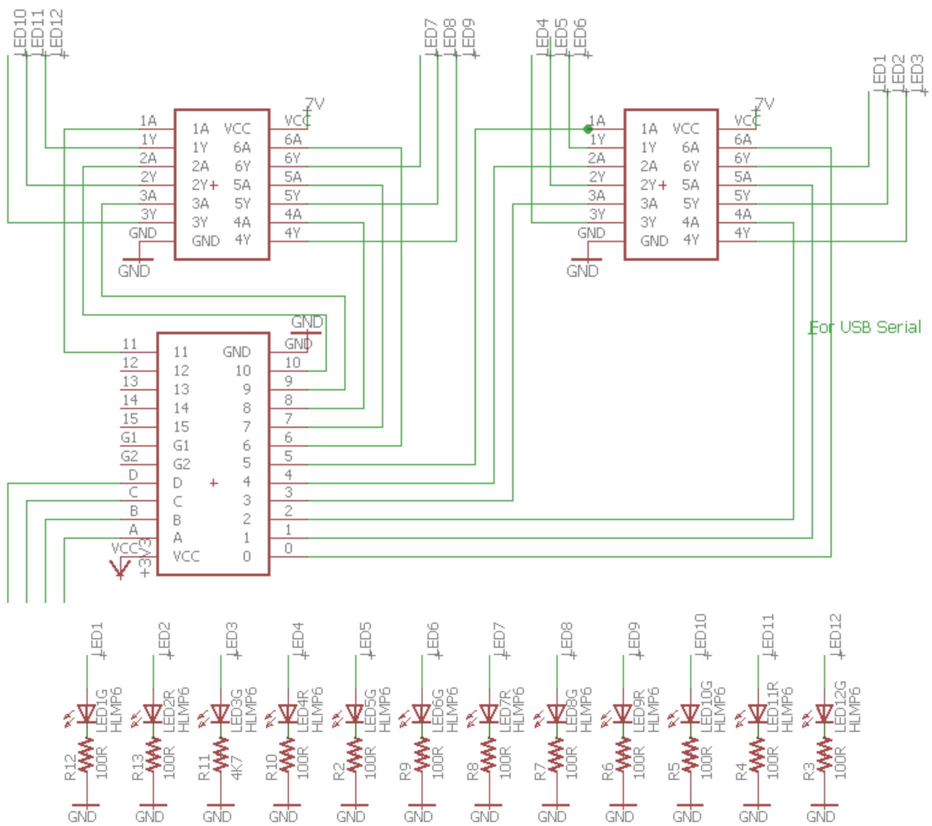


Figure 6: Schematic showing connection of LED display circuit

2.2.6 Waveform Module

2.2.6.1 Square to Sawtooth Converters

Figure 7 shows the twin square to sawtooth converters that generate the sawtooth waveform. The inputs come from the microcontroller and the outputs lead to the audio jacks. The left and right halves of the circuit create two different waveforms that when combined, form a sawtooth.

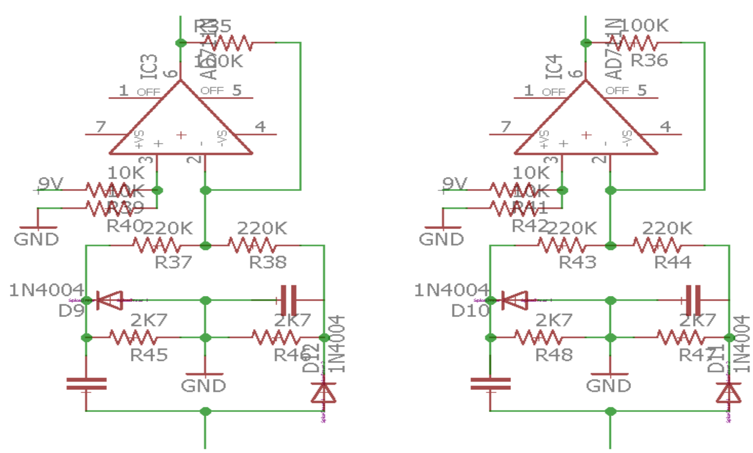


Figure 7: Schematic of square to sawtooth converter

2.2.6.2 Square to Sine Converters

Figure 8 shows the twin square to sine wave converters. The inputs come from the microcontroller output and the output goes to the quarter inch audio outputs. There are three processes happening between the input and the output of the square to sine converter. The first process is converting the square wave into a sine wave. A problem we encountered with our circuit was that the higher frequencies were dramatically quieter than the lower frequencies. This is because the square to sine converter we implemented also acts as a high pass filter, thus making the amplitude of higher frequencies much lower. The second process fixes this problem by implementing compressors for the waveforms. The compressors take the highest volume amplitude and the lowest and make them much closer together. This makes the difference between the high and low frequencies less dramatic in terms of volume amplitude. The third process takes the output from the compressor and amplifies it using an op amp.

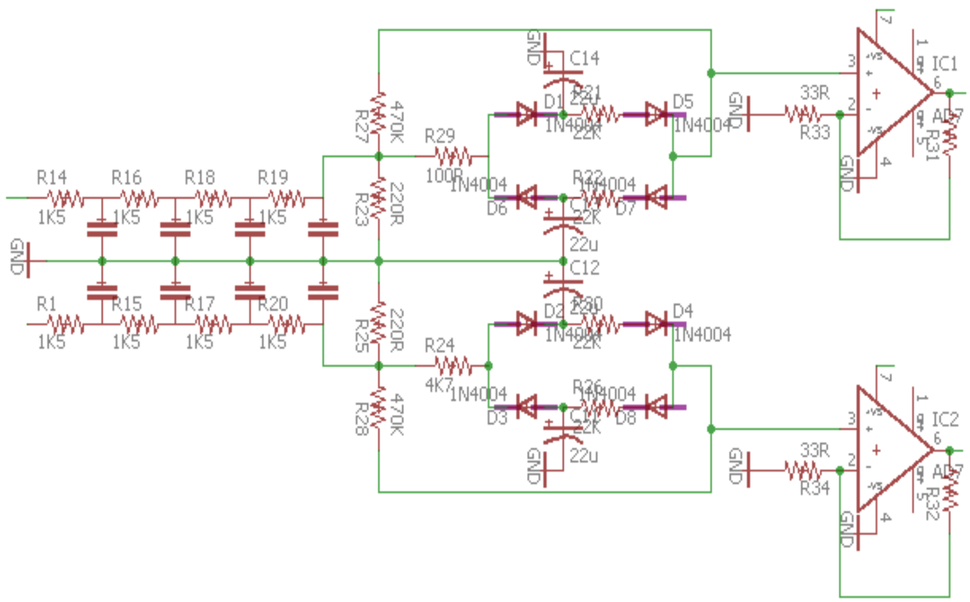


Figure 8: Schematic of the square to sine converter

3. Requirements and Verification

3.1 Infrared Sensor Verification (SHARP GP2Y0A21YK0F)

The sensor was connected to the supply voltage (Vcc) at 5.0V for testing. In the datasheet, the sensor should have the Vcc at least 4.5 volts but not exceed 5.5 volts, and the sensor can work between 10 cm and 80 cm [1]. We wanted to make sure that it follows the specification by testing it using a tape measure, a voltage generator, and a multimeter.

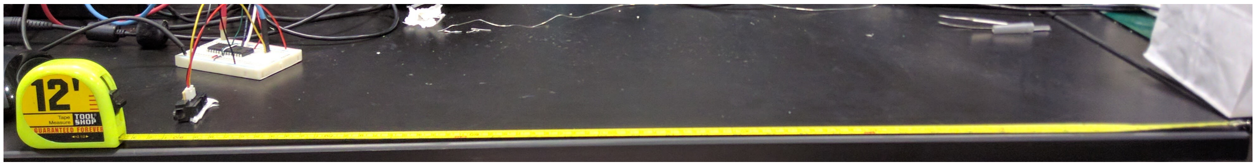


Figure 9: Output voltage measured base on the distance from the hand

The graph below, which shows the collected data of the output voltage by using a multimeter, shows relationship between the output voltage of the sensor and the distance away from the hand.

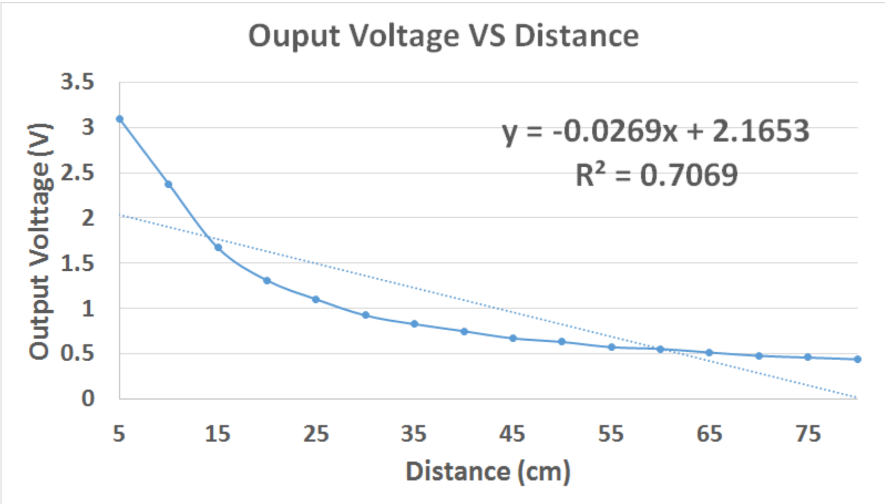


Figure 10: Relationship between output voltage and distance from hand

The data that we collected confirms that the sensor actually follows the specification from the data sheet, which is from 10 cm to 80 cm. However, the rate of change in output voltage when the sensor is about 80 cm away from the hand is very small comparing to when the sensor is closer to the hand. The trend line tells us that the relationship between the output voltage and the distance away from the hand is not best described by a linear function. R^2 is only 0.7069. This could be fixed by using logarithmic scale.

3.2 Microcontroller verification (Control Unit)

We decided to use LPC1114FN28/102 as the main microcontroller for our project. The following are the specifications that we were looking for:

- The chip must have at least 6 general digital input/output pins.
- The chip must have an internal Analog/Digital converter with at least 2 input channels
- The Analog/Digital converter must not have its delay more than 3 ms
- The chip must be able to output square waves up to 600 Hz

The followings are the actual specifications of the microcontroller:

- The chip has 22 general input/output pins [2]
- It has an internal Analog/Digital converter (up to 6 input channels) [2]
- The Analog/Digital converter takes about 2.44 **µ**s per conversion [2]
- The system clock is 50 MHz [2] (which is more than enough for processing 600 Hz square waves)

Therefore, the LPC1114FN28/102 meets all the requirements that we need for our project.

3.3 Software Verification

3.3.1 Analog/Digital converter driver verification

We measured and tested the performance of the internal Analog/digital converter. According to the datasheet, it can measure analog voltage between 0 volt to 3.3 volts [2]. Therefore, we testing it by inputting voltage from a voltage generator to the input of the internal A/D converter and reading the value from the register corresponding to the channel. We collected data and plot a graph for illustrating the relationship between the input voltage and read value. The plot is shown below

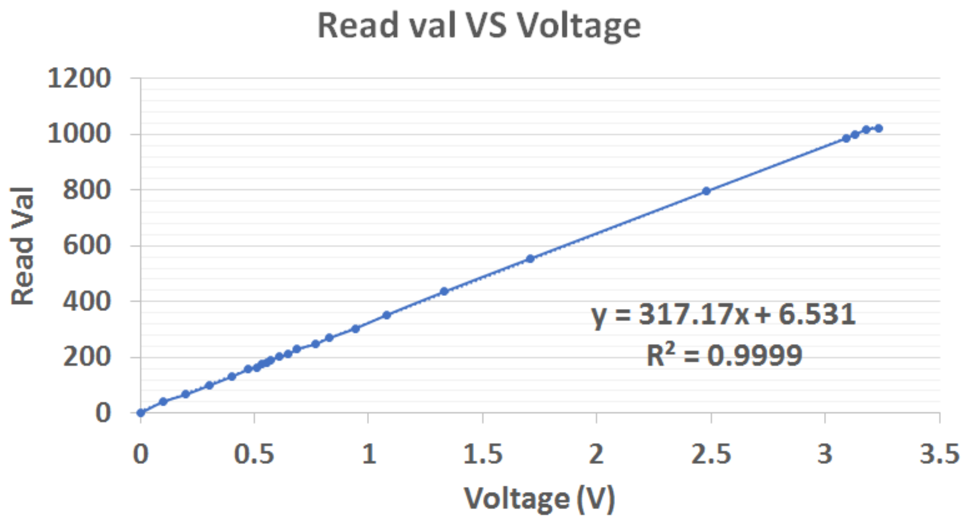


Figure 11: Relationship between read value and input voltage of the A/D converter

From the result above, we got a very great linear relationship. The $R^2 = 0.9999$ means that this linear function can explain the relationship 99.99%. The relationship function is as follow:

$$\boldsymbol{y = 317.17x + 6.531}$$

Equation 1: Relationship between read value and the input voltage

We tried reading the output voltage from the sensor by using the microcontroller. The result is shown below.

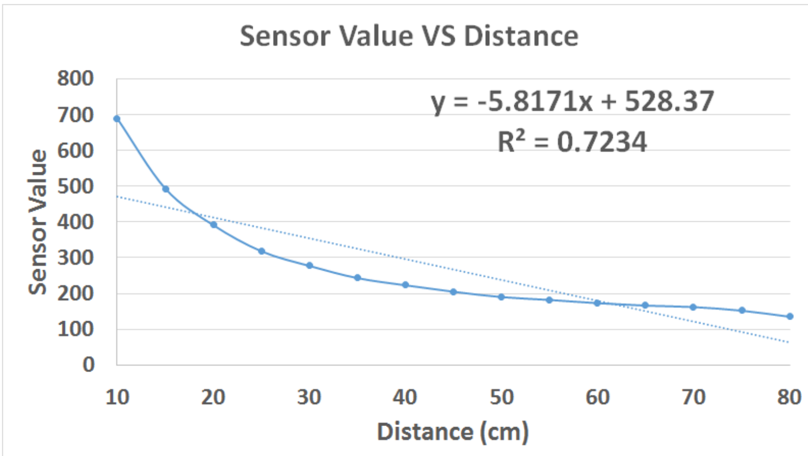


Figure 12: Relationship between read value and the distance from the sensor

The result is not linear due to the characteristic of the sensor that we measured earlier. The R^2value is only 0.7234, which means that the function can only explain the relationship 72.34%. This can be improved by using logarithmic scale to make the function more linear. The following is the result of using logarithmic scale.

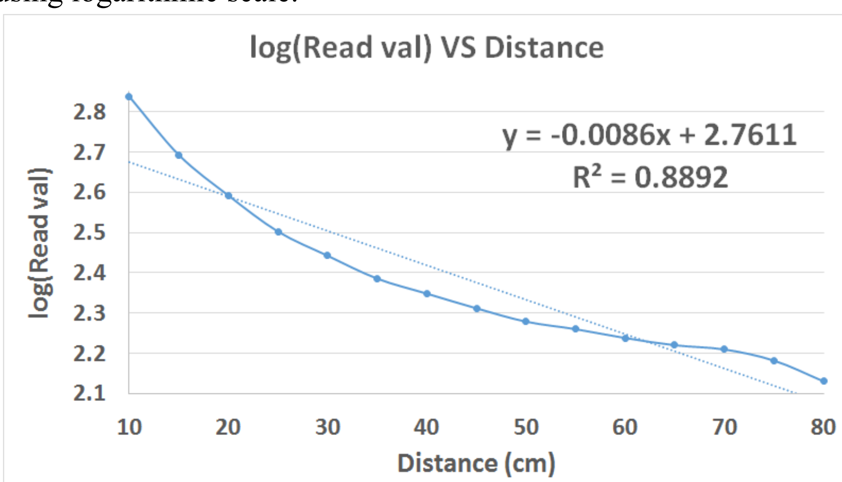


Figure 13: Logarithmic scale of the input voltage vs distance from the sensor

The logarithmic scale improves the relationship between the read value and the distance from the sensor. The R^2value is now 0.8892, which means that it can describe the relationship 88.92%.

3.3.2 Square wave generation algorithm verification

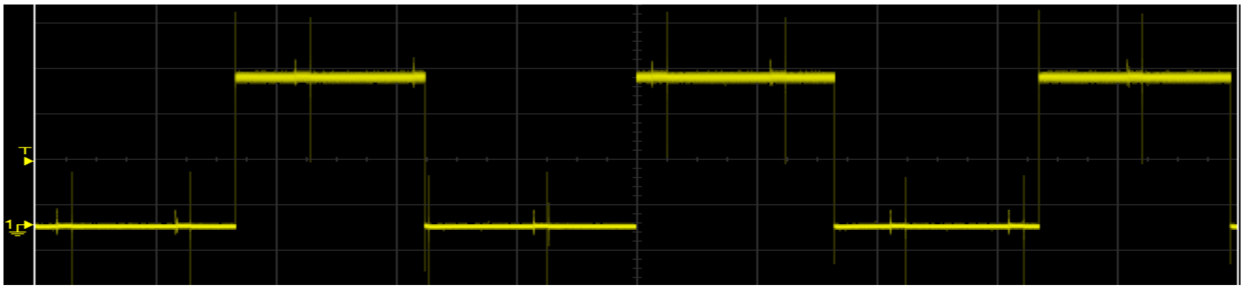


Figure 14: Snapshot of a square wave generating from the microcontroller

The output square waves can be tested by using an oscilloscope. It can generate square waves with the maximum frequency of 600 Hz and the minimum frequency of 60 Hz. A snapshot of a clean square wave is shown below.

3.4 Waveform Module Verification

3.4.1 Square to Sawtooth Converters Verification

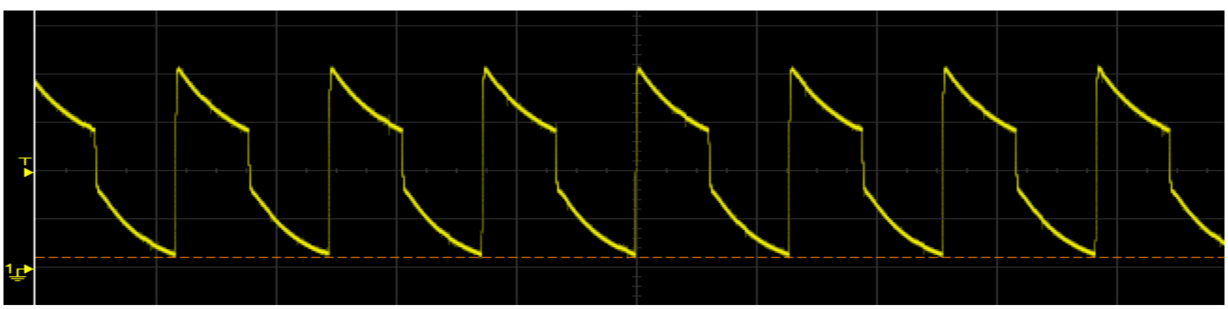


Figure 15: Snapshot of a sawtooth wave after passing the waveform module

The sawtooth waveform in figure 15 is not a perfect sawtooth. This is simply how the waveform ended up after correctly implementing the circuit. However, this imperfect waveform suits us for our purposes. This is because the purpose of these waveforms is to create subjectively pleasing sounds. After several subjective audio quality tests (i.e. simply listening to the waveform), the sawtooth waveform in figure 15 accomplishes this goal. It can be seen that the sawtooth waveform is clean, clear, and has consistent form and amplitude.

3.4.2 Square to Sine Converters Verification

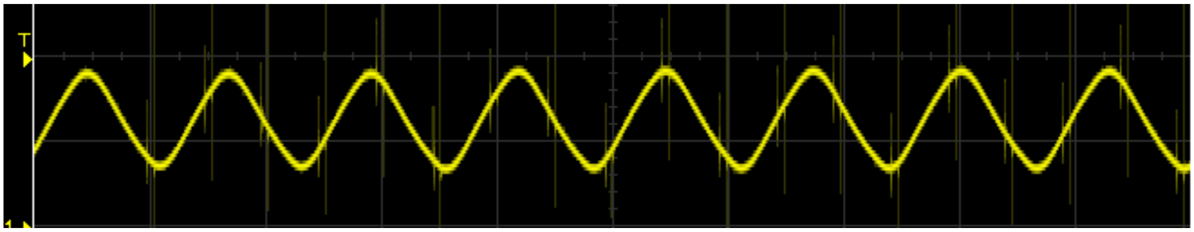


Figure 16: Snapshot of a sine wave after passing the waveform module

The square wave in figure 14 is much smaller in amplitude at the audio output that in the provided image. This is because the waveform in figure 14 was taken before it passes through the 46 kΩ resistor that attenuates the amplitude.

4. Costs

The fixed cost for development would be ($35/hour) with (15 hours/week) actual spent for each member on the team. There are three people on the team working over the course of the semester (13 weeks).

Labor = 3 people * $35/hour * 15 hours/week * 13 weeks * 2.5 = $51,187.50

Equation 2: Labor Cost

Table 1: Table of Parts costs

Quantity	Part	Cost per unit ($)	Cost per unit in bulk ($)
2	Infrared Sensors (Sharp GP2Y0A21YK0F)	9.95	6.95
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
6	Toggle switches (generic)	1.55	1.15
2	Inverter (generic)	0.75	0.28
1	Foot Pedal (generic)	62.00	47.00
3	Audio output (generic)	2.27	1.59
1	Voltage Regulator (UA78M33)	0.55	0.30
1	Power transformer (Generic)	6.00	5.50
1	Perfboard	1.00	0.50
	Total	140.00	93.62

We estimated that the cost of the parts and manufacturing the prototype would be $200.00 each. During the development phase we created a prototype.

Cost of Parts = 1 prototype *200/prototype = $200.00

Equation 3: Cost of Parts

Grand Total = Labor + Cost of Parts

Equation 4: Grand Total

Therefore, the total development cost will be $51,387.50.

5. Conclusion

5.1 Accomplishments

We faced several difficulties throughout the course of this project. At times, it felt as though everything was working against us. From proposing the project idea at the beginning of the semester until the night before the demo, our group faced several setbacks every step of the way. Despite fighting against the grain for an entire semester, our group ended up with a final product that we are all very proud of. Our Theremin produced clean and clear tones that spanned nearly three octaves. The LED display accurately represented the pitches being played. The Theremin was able to produce full and distinct square, sine, and sawtooth waveforms. The volume foot pedal flawlessly created the volume swells that are characteristic of a Theremin. The killswitch articulated the notes to create a unique staccato effect. And to top it all off, the box we designed to house the electronics ended up looking stylish.

5.2 Uncertainties and Future Work

We implemented the waveform modifier for both the high pitch and low pitch sensors. However, the low pitch sensor was accidently destroyed by overloaded voltage during assembly. Hence, we were not able to test the combination of both notes being played at the same time. We aim to improve this project in the future by incorporating and testing low pitch sensor, and by adding 5V linear voltage regulator for the two sensors. We would also like to improve the quality of the sound by further tuning and testing algorithm for tuning an instrument, and examining waveform circuit to improve wave modifier.

5.4 Ethical Consideration

We as a group envisioned 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 [3]. We want our product to be as safe as possible for any user by applying safety measures listed above. Furthermore, we will be seeking feedback from many different people such as friends, families, instructors, and teaching instructors. As engineers, 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 [3]. Our goal is to meet the expectation that others have for our product and to make it as flawless as possible.

6 References

[1] SHARP Corporation, "Pololu robotics and electronics," 01 Dec 2006. [Online]. Available: https://www.pololu.com/file/download/gp2y0a21yk0f.pdf?file_id=0J85. [Accessed 01 May 2017].

[2] NXP Semiconductors, "Product data sheet," NXP Semiconductors, [Online]. Available: http://www.nxp.com/documents/data_sheet/LPC111X.pdf. [Accessed 1 May 2017].

[3] "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].

Appendix A Requirement and Verification Table (R&V)

Table 2: Table R&V for Infrared Sensors

Requirement	Verification	Verification status (Y or N)
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 38 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 detect at desired maximum and minimum distances. 2. 2. The update period confirm by datasheet	1. Y 2. Y

Table 3: Table R&V for Microcontroller: LPC1114FN28/128

Requirement	Verification	Verification status (Y or N)

1. The chip must have at least 6 general digital input/output pins. 2. The chip must have an internal Analog/Digital converter with at least 2 input channels 3. The Analog/Digital converter must not have its delay more than 3 ms 4. The chip must be able to output square waves up to 600 Hz	1. The chip has 22 general input/output pins [2] 2. It has an internal Analog/Digital converter (up to 6 input channels) 3. The Analog/Digital converter takes about 2.44 µs per conversion 4. The system clock is 50 MHz (which is more than enough for processing 600 Hz square waves)	1. Y 2. Y 3. Y 4. Y

Table 4: Table R&V for Waveform modifier

Requirement	Verification	Verification status (Y or N)
1. The waveform circuit must be able to produce clean sine wave and sawtooth wave.	1. Run a square waveform through the circuits and examine the waveform converted is clean using an oscilloscope.	1. Y

Table 5: Table R&V for Decoder/Demultiplexer

Requirement	Verification	Verification status (Y or N)

1. It has to be able to decode to select at least 12 different output. 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.	1. Y 2. Y

Table 6: Table R&V for LEDs

Requirement	Verification	Verification status (Y or N)
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.	1. Y