

# **Wirelessly Controlled Guitar Pedal**

## **ECE 445 Design Document**

Marco Palella, Chris Scheurer, Will Newman

Group 22

TA: David Null

4/17/20

# 1 Introduction

## 1.1 Objective

Performing live at a concert often requires musicians to move around constantly on stage. Whether it be to crowd surf, mosh, or just to move to the rhythm of the music they're playing, the actions of a musician on stage can make or break a performance. Traditionally, guitars are connected to sound equipment, using pedal boards to control effects like distortion, echo, reverb, filtering, etc. [1]. Pedals are typically daisy chained together, and contain control knobs to adjust different parameters on each pedal [2]. This means that if the musician wants to make real-time adjustments, they need to be standing near the pedal and are forced to adjust parameters with their feet. There are existing devices on the market that allow for the modulation of certain effect parameters wirelessly, however, there are no devices on the market that allow a guitar player to remotely control several effect parameters on a pedal [3].

This pedal would be desired by musicians because it will increase the musical freedom of the musician on stage. Being able to adjust signals live from a device that is mounted directly onto a guitar is something that a lot of musicians are moving towards. For example, Matt Bellamy, the lead guitarist of Muse has incorporated a MIDI X-Y control pad into his guitar so that he can modulate his guitar during performances [4]. Music is moving in a direction that combines acoustic and digital elements, which would make a guitar-mounted controller very appealing to musicians.

Our solution to this problem is to create a transmitter unit that can be placed on the guitar. This device will be able to turn effect(s) on/off, and will have several potentiometers to give the user control over effect elements. The transmitter will communicate with a receiver unit that will take an analog guitar input, apply the effect/process the audio data digitally based on parameters coming from the transmitter, and then convert it back into an analog output. This receiver can be daisy chained to a guitar pedal board and is set up exactly like any other guitar pedal.

## 1.2 Background

Guitar pedals are widely used to manipulate the sound coming from a guitar during live performances. Typically, pedals are switched on with a button or a large potentiometer that can be moved with your foot, and they have knobs that adjust different parameters of the effect. Guitar pedals typically adjust the dynamics of the guitar, filter the guitar, or add reverb/echo to the guitar.

While performing, guitarists are often limited in their ability to move around because they must remain close enough to their guitar pedal board to operate the effects. We aim to alleviate this problem by exporting the operation of a digital guitar pedal to a wireless controller mounted to the guitar. Another ECE 445 team previously attempted to solve this problem in the Spring semester of 2003. However, their design (while functional) introduced several other issues [5]. Briefly put, their system operated on Radio Frequency (RF) and transmitted the guitar signal wirelessly. The reason RF transmission is a problem is because RF is open to any RF receiver such as a "walkie talkie" (commonly used by venue personnel), which means that instead of the guitarist's "sick" guitar solo, the crowd may be hearing the security guards talking about their after-show plans. Although transmitting the guitar's signal/sound over RF frees up the guitarist's movement, the guitar will be entirely removed from the audio mix if the RF signal is

interrupted. If the guitar output is fed to the receiver and the wireless signal is interrupted, it will not impact the audio feed at all; the user will just be unable to make live adjustments.

The solution to these problems, which we will implement, will be to convert RF to Bluetooth and to have the audio from the guitar still be transmitted via cable instead of being transmitted wirelessly. This reliance on the cord still restricts the guitarist's movement; however, it is significantly easier to obtain and use a longer cord, and the guitarist's freedom of movement is still greatly expanded.

### *1.3 High Level Requirements*

- If a 1V peak to peak analog sine wave is inputted to the receiver, the receiver's output must be an identical signal that has a latency less than 13 ms with a Signal-to-Noise Ratio (SNR) greater than 20dB
- Bluetooth transmitter must be able to send 12-bit digital potentiometer values to the bluetooth receiver at a distance of 10 feet
- Receiver microcontroller must be able to process audio data (data from the guitar) at a sampling rate of at least 48 kHz and a bit resolution of 16-bits

## **2 Design**

Our design is going to consist of two major systems: a transmitter module and a receiver module. These modules will communicate with one another using a bluetooth transmitter and receiver, and will be powered independently from one another. This will make it possible for the user to travel freely.

The transmitter module will be mounted on the guitar and powered by an on-board battery. It will have a potentiometer module that a user can interact with, along with a power light and switch. The first submodule is the power module, which will supply a voltage drop across the potentiometers, along with regulating power to other modules on the board. The voltage drop across each potentiometer will be converted into a digital signal by an Analog-to-Digital Converter (ADC) and sent into a microprocessor, which will communicate data and control commands to an onboard bluetooth module. There will also be an on-board oscillator to ensure that the transmitter is synchronized with the receiver.

The receiver module will be a separate device that will be connected to the guitar data line like a guitar pedal (i.e. the line from the guitar is the input, and the output is either connected to an amp or other pedals). This module will be powered by a 9V supply, and will have an LED to verify that power is being supplied to the device. The analog input will be converted into digital data and sent to a microcontroller via an I2S data line. This data will be processed by the microcontroller based on parameters being transmitted to the bluetooth receiver by the transmitter. The processed digital data is then converted back into an analog signal with a Digital-to-Analog Converter (DAC).

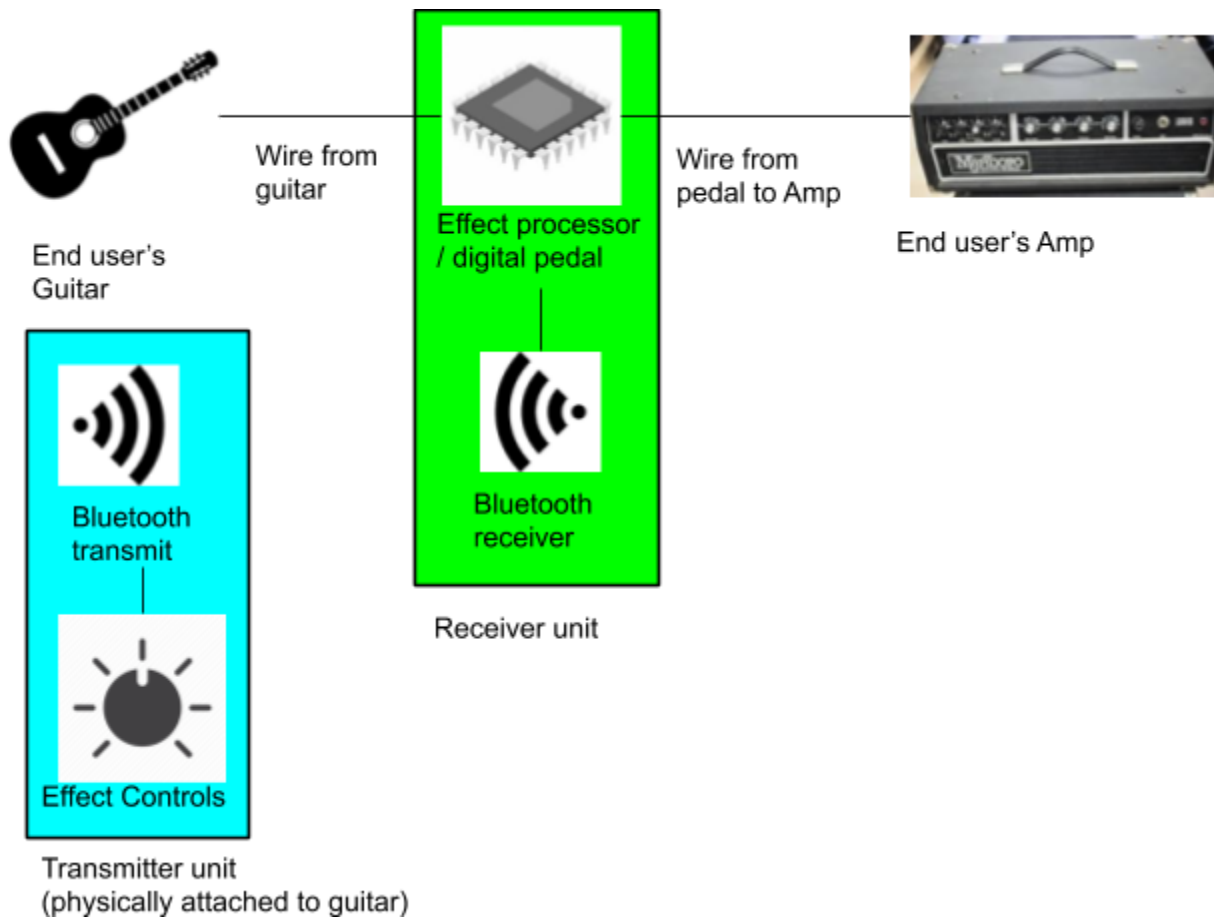


Figure 1. Physical layout

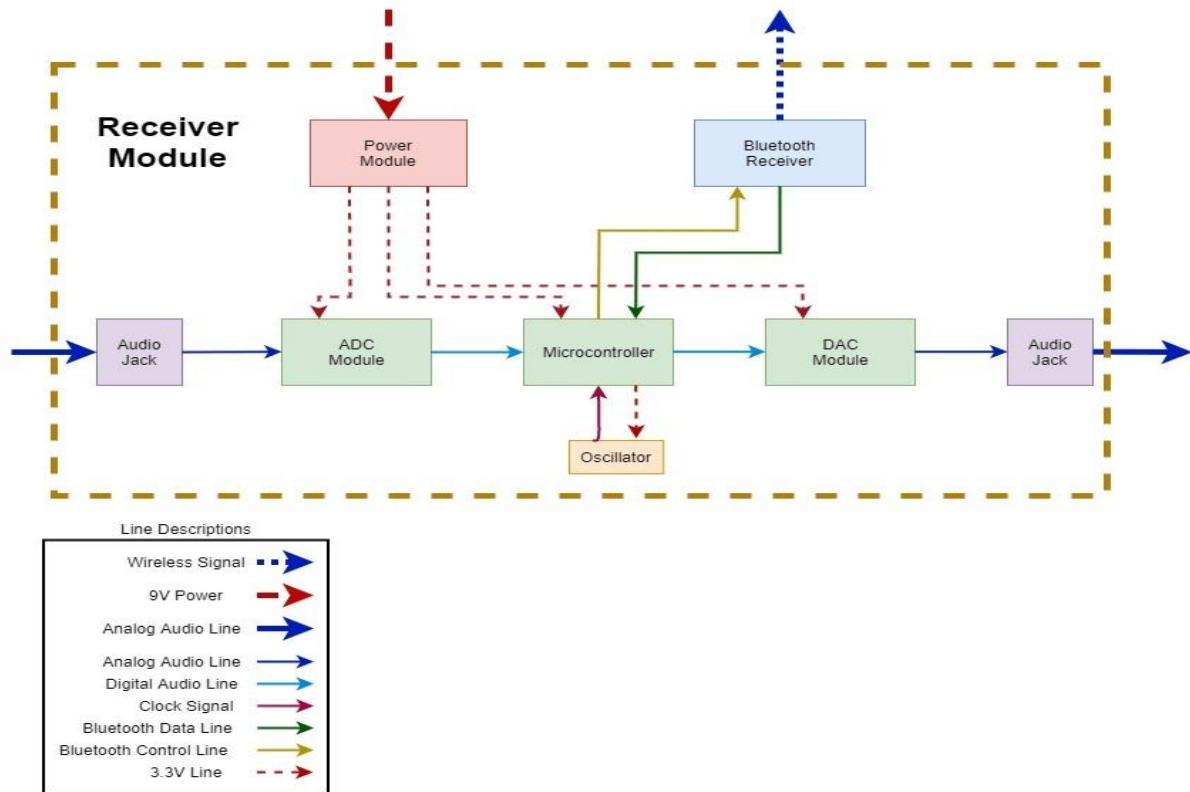


Figure 2. Block Diagram of Receiver Module

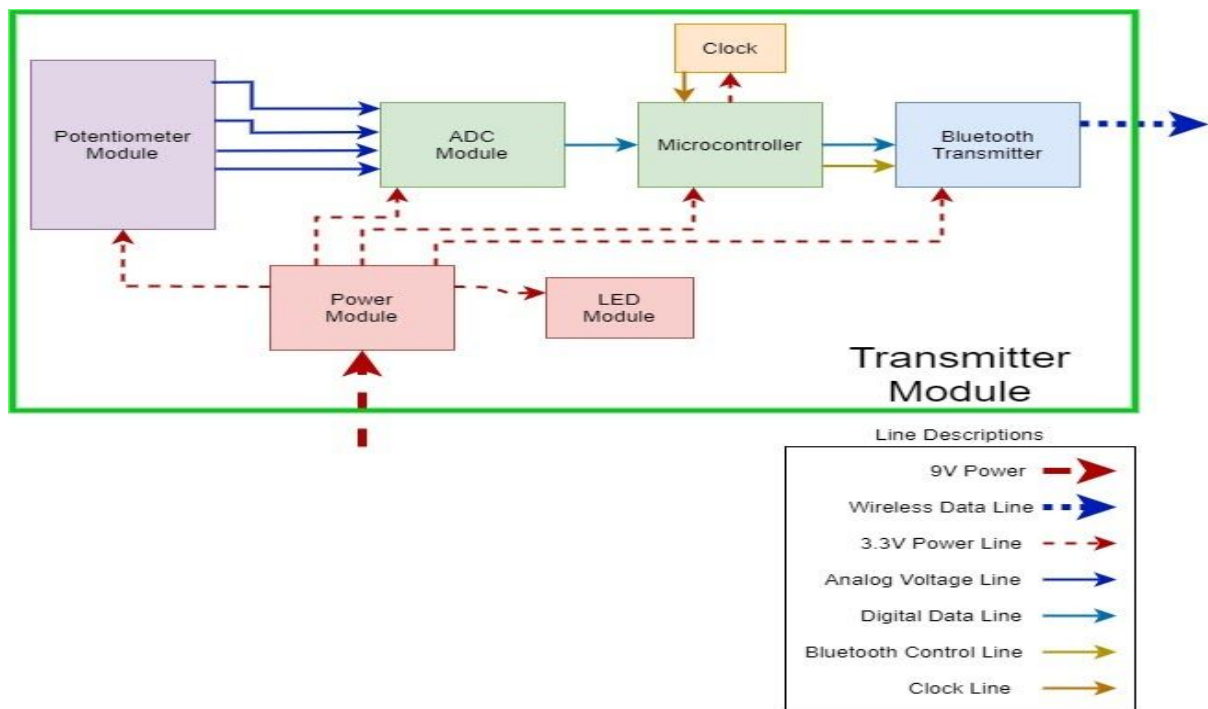


Figure 3. Block Diagram of Transmitter Module

## 2.1 Transmitter Subsystem

The transmitter subsystem is responsible for wirelessly sending the user input data to the receiver. The user will operate several potentiometers, which will create analog voltage values. These values will then be digitized and relayed to a bluetooth transmitter. A microcontroller will be in charge of managing how the data is transmitted.

### 2.1.1 Power Module

The power module is responsible for supplying a consistent 3.3 volt signal to power the microcontroller and all other modules on the transmitter. To ensure the power supply is consistent, a voltage regulator circuit is placed immediately after the power source.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must consistently output a 3.3V DC signal when given a 4.5V input</li><li>2. Must be able to power the transmitter system for at least 8 hours with 3 fully charged AA batteries.</li></ol>	<ol style="list-style-type: none"><li>1. connect 3 AA batteries in series to power connection, measure output with multimeter to ensure the voltage is 3.3V</li><li>2. Connect batteries to the system for 8 hours. Watch the power LED and measure the output of the power module with a voltmeter and</li></ol>

### 2.1.2 ADC Module

The ADC module is the sub system that is responsible for digitizing the analog signals from the potentiometers. The analog signal will be read from the Potentiometer module via a voltage divider rule on the potentiometer.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must convert the Voltage divider voltage into an 8-bit I2S digital word.</li><li>2. Must have an SNR greater than 20 dB</li><li>3. Must be powered by a 3.3V line</li></ol>	<ol style="list-style-type: none"><li>1. Using a predetermined analog signal (potentiometer at <math>\frac{1}{2}</math> max value) convert this into a digital signal. Double this control signal and the digital value should double as well</li><li>2. Provide a predetermined signal ( 1V sine wave) and subtract the digital representation of the same signal. Divide the signal by the noise.</li><li>3. Read the power input line with a voltmeter and ensure it is 3.3V</li></ol>

### 2.1.3 Bluetooth Transmitter

The Bluetooth Transmitter is responsible for transmitting the digitized data from the ADCs to the Bluetooth Receiver in the receiver system. The receiver and transmitter must additionally pair for secure data transfer. The process of getting the data to the transmitter is not governed by the transmitter (see 2.1.4 Microcontroller for data control). Rather, the transmitter will broadcast whatever data it has when it is told to transmit data from the receiver.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must be able to pair with receiver system's Bluetooth module</li><li>2. Must be able to transmit 8-bit data at a rate of 1 kHz to the receiver</li><li>3. Must respond to the serial data line and clocking signals sent to it by the microcontroller</li></ol>	<ol style="list-style-type: none"><li>1. System will not be detectable by the bluetooth on your phone if pairing is not possible or broken.</li><li>2. If digital data is sent by the transmitter the serial line on the receiver will read the same (this also tests the receiver)</li><li>3. If system is not properly controlled by the microcontroller the 2 previous tests will fail</li></ol>

### 2.1.4 Microcontroller

The microcontroller will be in charge of data flow. This includes controlling the Oscillator module for proper clocking, managing the data transfer protocol to move data from the ADCs to the bluetooth transmitter and controlling pairing and transmission timing for the bluetooth system.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must control data transfer from the ADC module to the Bluetooth transmitter</li><li>2. Must send ADC data only when it is prompted to by the receiver</li><li>3. Must control the Bluetooth transmitter and read data at least every 1 ms</li></ol>	<ol style="list-style-type: none"><li>1. Sweep any one potentiometer and read the I2S data leading into the bluetooth transmitter. If this data doesn't change as the potentiometer is swept this test is failed</li><li>2. Store arbitrary data in memory. Measure data line going to bluetooth. Make sure no data is being transmitted, then send command from receiver to get data and ensure that arbitrary data is getting sent</li><li>3. Use a debugger to ensure that the bluetooth data is being stored inside of memory on the microcontroller</li></ol>

## 2.1.5 Oscillator Module

The Oscillator module will be the clocking source for the transmitter system. The clocking speed is governed by the microcontroller and used to dictate the speed of data transfer and how often bluetooth data is sent.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must oscillate at 24 MHz</li><li>2. Must be powered by a 3.3V line</li></ol>	<ol style="list-style-type: none"><li>1. Read the clocking line on an oscilloscope. Measure the period of the clock oscillation. Find the inverse of the period to ensure the oscillator is providing a 24 MHz input</li><li>2. Read the power input line with a voltmeter</li></ol>

## 2.1.6 LED Module

The LED module is used to show proper pairing. When the LED is on, the transmitter is paired to the receiver and able to transmit data. The LED will also start blinking when the transmitter system is running low on battery life.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must light when bluetooth is fully paired</li><li>2. Must blink when power module drops below 3.3V</li></ol>	<ol style="list-style-type: none"><li>1. Observe when not paired, and then pair, if the LED goes from off to on and vice versa the system works</li><li>2. Send a 3.3V DC signal into the module, if LED begins to blink the system works</li></ol>

### 2.1.7 Potentiometer Module

The potentiometer is the user interface for the transmitter. By turning the potentiometers, the user controls the value of parameters in the guitar effect. In the transmitter, these are derived via a voltage divider rule. The ADC module converts this to a digital signal, and the actual interpretation of the signal is the job of the microcontroller in the receiver sub system

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must have 4 Potentiometers wired parallel, so that the adjustment of any one potentiometer will not affect the value of the others.</li><li>2. Each potentiometer branch must not draw more than 1mA</li><li>3. Must be able to range the voltage over each potentiometer from ~0v to 1.65V</li></ol>	<ol style="list-style-type: none"><li>1. Using the Oscilloscope to track the 4 separate analog signals, change one potentiometer at a time. If no one signal affects the rest, repeat moving any 2 potentiometers at one time. If both tests pass this requirement is verified</li><li>2. Measure current flow with a DMM while varying the potentiometer resistances</li><li>3. Set potentiometer to MAX resistance and measure voltage across the potentiometer, repeat for MIN resistance position,</li></ol>

## 2.2 Receiver Subsystem

The receiver sub system is responsible for receiving data sent out by the transmitter sub system and interfaces with the guitar's sound signal. The receiver not only gets the data sent out by the receiver but is the mechanism actually interpreting the data so that the potentiometers on the receiver can affect the sound of the guitar.

### 2.2.1 Power Module

The power module is responsible for supplying a consistent 3.3 volt signal to power the microcontroller. To ensure consistent power supply a voltage regulator circuit is placed immediately after the power source.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must step down the voltage from the stomp pad down to a 3.3 Volt DC signal</li><li>2. Must protect from voltage surges</li></ol>	<ol style="list-style-type: none"><li>1. Send in a 4.5V DC signal from a signal generator and sweep up to a 10V DC signal, output should maintain 3.3V</li><li>2. If the input signal is above 10V this module should break to protect all other systems.</li></ol>

### 2.2.2 ADC Module

The ADC module in the receiver is used to convert the analog guitar signal into a digital signal. This allows a digital effect to be applied to the guitar. Since the ADC module is affecting the guitar's sound, a critical quality of the module is to sample fast enough and have a large enough bit depth to prevent aliasing and noise from being injected into the data stream.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must convert the guitar's sound signal to a 24-bit I2S digital signal</li><li>2. Must be able to sample 48 kHz or faster</li><li>3. Must have an SNR greater than 20 dB</li></ol>	<ol style="list-style-type: none"><li>1. When guitar is strummed, I2S data should be output from the ADC, this can be viewed on the Oscilloscope. The accuracy of this data is handled via the SNR test.</li><li>2. Connect clocking line to Oscilloscope and read clocking frequency</li><li>3. Provide a predetermined signal ( 1V sine wave) and subtract the digital representation of the same signal. Divide the signal by the noise.</li></ol>

### 2.2.3 DAC Module

The DAC module is naturally performing the inverse of the ADC module. After the digital effect or lack thereof is applied to the guitar's digital signal, the DAC module will convert the signal back to analog data. This is essential, as all other pedals and the amp for a guitar need analog data. Should this module malfunction, the data will become gibberish for any additional in-series systems.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must convert a 24-bit I2S digital signal from the microcontroller into an analog signal</li><li>2. Must operate at 48 kHz or faster</li><li>3. Must have an SNR greater than 20 dB</li></ol>	<ol style="list-style-type: none"><li>1. Store a sine wave of 1kHz and 1V amplitude in memory and output it to the DAC I2S data lines. Measure the outputs of the DAC with an oscilloscope.</li><li>2. Measure bit clock channel from DAC. find the frequency of the signal and divide it by the bit depth(24 bits). This should be 48kHz.</li><li>3. Measure the output signal for 10 seconds on the oscilloscope using a sine wave with the same parameters from test one. Measure the same sine wave directly from the function generator. Import the data from the oscilloscope and divide the ideal situation by the output. This will give you the total noise in your system.</li></ol>

## 2.2.4 Microcontroller

The microcontroller will be in charge of data flow. This includes controlling the Oscillator module for proper clocking, managing the data transfer protocol to move data from the ADCs to the DAC and data from the Bluetooth receiver to itself. Additionally the microcontroller will control pairing and receiver timing for the bluetooth system.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must control the I2S data transfer protocol</li><li>2. Must be able to initialize bluetooth receiver and obtain data from it</li><li>3. Must be able to transmit commands to bluetooth receiver</li><li>4. Must apply the pedal effect. If there is no effect or the effect is turned off, the microcontroller must pass the signal through without modifying the signal</li></ol>	<ol style="list-style-type: none"><li>1. Use another microcontroller to transmit I2S data to the microcontroller ADC input. Read the I2S DAC output data line of the microcontroller and ensure that the same data is being pushed through this line.</li><li>2. Try to pair phone with bluetooth receiver to ensure it has been configured correctly, and ensure that data is being stored from bluetooth receiver port using a debugger</li><li>3. Send a command to receive data from transmitter and measure it on data line using an oscilloscope</li><li>4. Input a 1V sine wave into the guitar in jack, and have no effect programmed, if the signal is not the same signal / noisy version of the signal this test fails. Turn the effect on and ensure that output has changed accordingly.</li></ol>

### 2.2.5 Audio Jack Modules

The Audio Jack modules act as interfaces to external guitar systems. This includes but is not limited to the guitar itself, other pedals, the amp, and so forth. The input and output will ¼” inch jacks, an industry standard for guitar audio equipment.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. The Audio jacks must not make the SNR less than 60 dB</li><li>2. The Audio jack must be durable enough for rouged conditions</li></ol>	<ol style="list-style-type: none"><li>1. Provide a predetermined signal (1V sine wave) and subtract the digital representation of the same signal. Divide the signal by the noise.</li><li>2. Plug in and pull out a guitar cord 20-30 times to simulate normal use. After this max durability can be tested by inserting a cord and aggressively pushing on the coupling perpendicular to the jack. If this doesn't break the jack or the rest of the system this test has exceeded expectations/verification</li></ol>

### 2.2.6 Bluetooth Receiver

The Bluetooth receiver is in charge of receiving the data sent out by the Transmitter system's bluetooth transmitter. The Bluetooth receiver will be paired with the Bluetooth transmitter in the transmitter system to ensure secure data transfer. When the receiver tries to get data will be dictated by the clocking of the microcontroller.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must be able to pair with transmitter system's Bluetooth transmitter</li><li>2. Must be able to receive data to the transmitter</li><li>3. Must respond to the serial data line and clocking signals sent to it by the microcontroller</li></ol>	<ol style="list-style-type: none"><li>1. System will not flash a built in LED if pairing is not possible or broken.</li><li>2. If digital data is sent by the transmitter the serial line on the receiver will read the same (this also tests the transmitter)</li><li>3. If system is not properly controlled by the microcontroller, the 2 previous tests will fail</li></ol>

### 2.2.7 Oscillator Module

The Oscillator module will be the clocking source for the receiver system. The clocking speed is governed by the microcontroller and used to dictate the speed of data transfer and how often bluetooth data is received and how fast the guitar's sound signal is sampled.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must provide at least 24 MHz of clocking.</li><li>2. Clock must be powered by 3.3V input</li></ol>	<ol style="list-style-type: none"><li>1. Read clocking line on an Oscilloscope</li><li>2. Apply 3.3V input, and measure the output of oscillator with oscilloscope to ensure the clock is operating correctly</li></ol>

### 2.2.8 LED Module

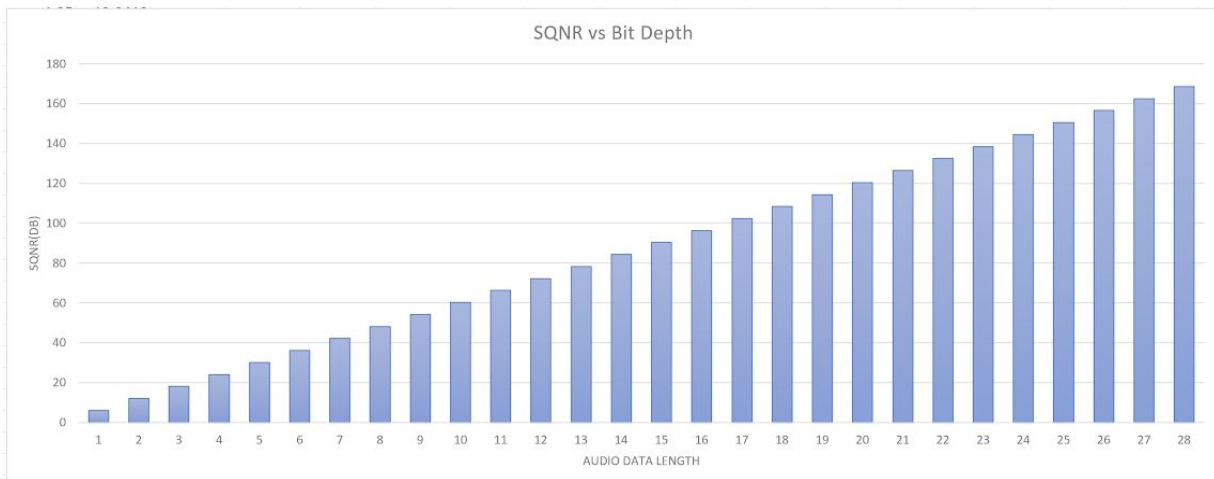
The LED module is used to show proper pairing. When the LED is on, the receiver is paired to the transmitter and able to receive data. The LED will also blink similar to the transmitter if the power module drops below 3.3V, indicating damage to the power module.

Requirements	Verifications
<ol style="list-style-type: none"><li>1. Must light when bluetooth is fully paired</li><li>2. Must blink when power module drops below 3.3V</li></ol>	<ol style="list-style-type: none"><li>1. Observe when not paired, and then pair, if the LED goes from off to on and vice versa the system works</li><li>2. Send a 3.1V DC signal into the module, if LED begins to blink the system works</li></ol>

## 2.3 Tolerance Analysis

The most important section of our design will be the system in charge of ensuring audio data playback. This includes all the modules that will be inputting analog audio, converting it to digital data, processing it, then converting it back into an analog signal, and outputting it. This section of the design is the audio jacks, ADC/DAC module, and the microcontroller. This section is vital because, without this system working, it cannot be effectively determined whether the rest of the pedal is working properly. If a clean signal cannot be outputted in the first place by the pedal, and everything else is working perfectly, the device is completely useless to a consumer. The most important specification of this system that needs to be analyzed to ensure a clean audio signal is the overall SNR of the system, and all the elements of the system that can potentially produce error. Our high level requirements outline that we need an SNR of 20 dB or greater, so this analysis will ensure that this requirement will be satisfied using an ADC and DAC with a bit resolution of 24 bits.

The largest amount of noise that will enter our audio signal will occur during the transition from analog to digital data and vice versa. So long as no data loss occurs during reception, processing, and transmission inside of the processor, there will be no noise added by the microcontroller. This can easily be done using a 32-bit MCU that can process more than 40 instructions per second [6]. ADCs and DACs will naturally add noise to your signal due to the effect known as quantization error. This error occurs because any ADC or DAC must round the value of an analog signal by some degree[7]. This error is reduced as you increase the bit resolution of your audio signal. Below is a bar chart that shows the relationship between the SQNR, which is the SNR created due to quantization error, and bit length in an ADC with a 5V voltage span [7].

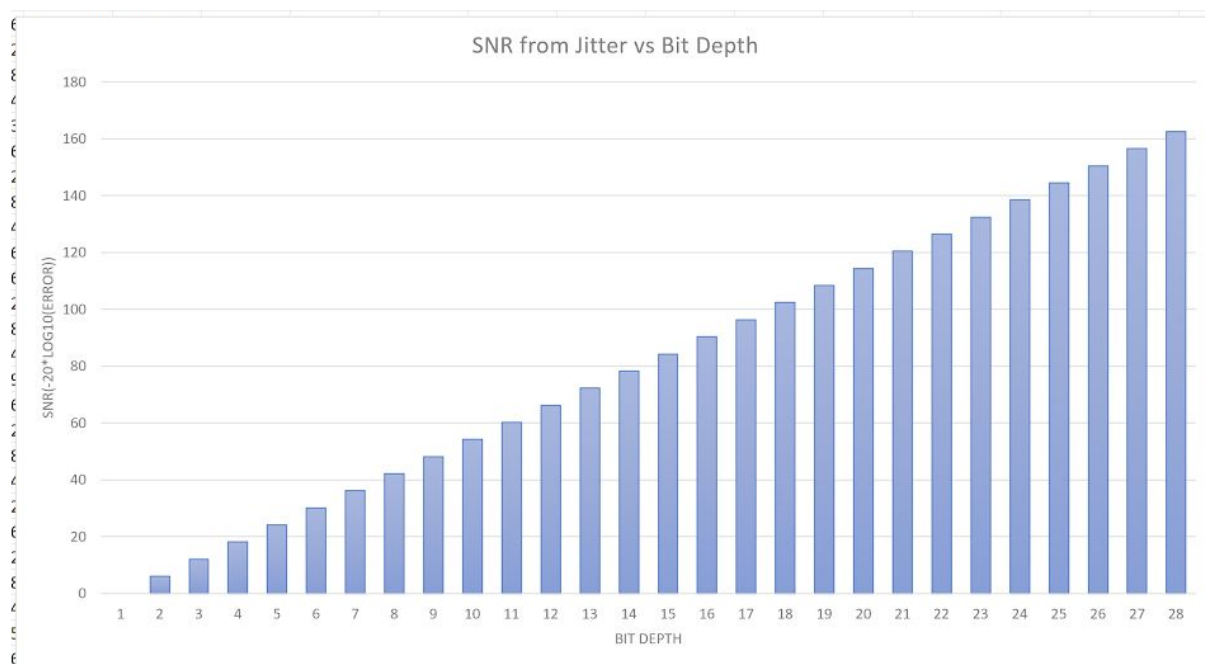


**Figure 5. SQNR vs Bit Depth**

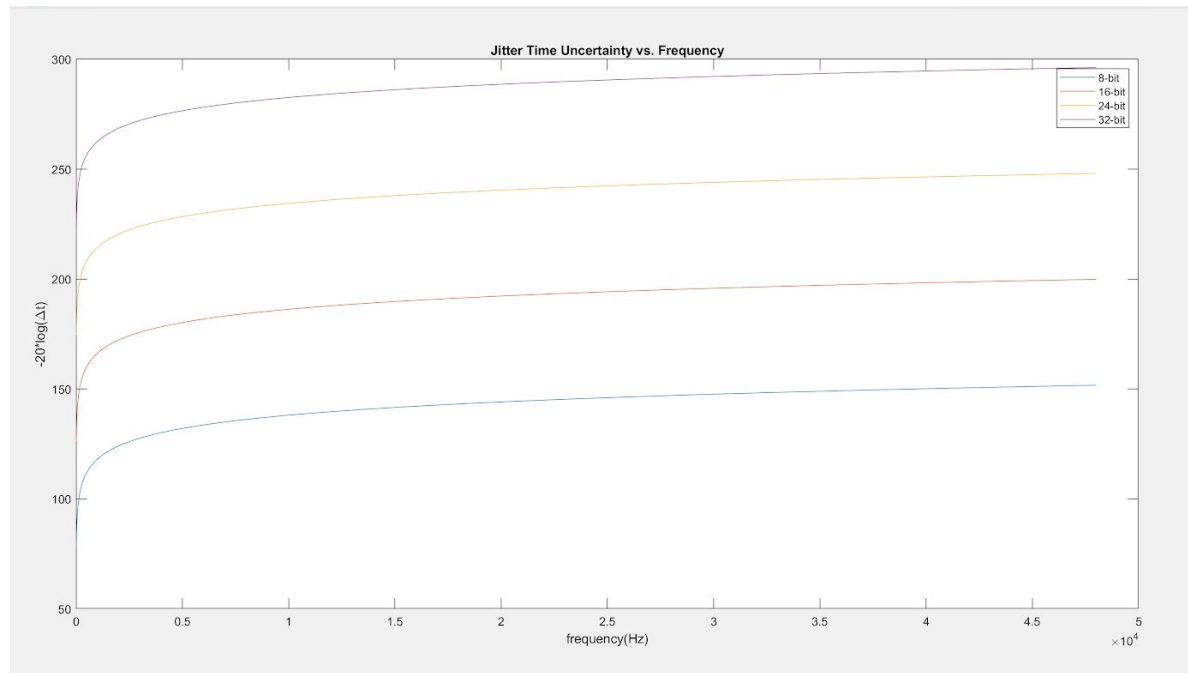
This graph shows that the SQNR (in dB) is linearly related to the bit depth, and that a 4-bit data length will give us adequate SQNR; however, since quantization error is not the only form of noise that can be introduced into the system, we have selected a bit resolution that will give a much higher SQNR than needed. Even though a bit depth of 16 is plenty for our ADC, we have chosen 24 bits for marketability purposes; people typically affiliate a higher bit-depth with higher audio quality.

Another thing to consider when making design decisions for the ADC and DAC is their jitter. Jitter is a phenomenon that is caused by sampling clocks being unideal. These clocks can sometimes

produce error due to there being an uncertainty about when samples are recorded [7]. Jitter is dependent on the frequency of the audio and on the bit depth. The time uncertainty  $\Delta t < \frac{1}{2^q \pi f_o}$ , where  $f_o$  is the frequency of your signal and  $q$  is the bit depth of your audio. Since we will be using a sine wave for testing purposes, we can model our audio signal as being  $s(t) = A \sin(2\pi f_o t)$ . According to wikipedia, the error that corresponds to the jitter of a audio signal corresponds to the following inequality:  $E_j \leq |s'(t)\Delta t|$ . Thus, with the sine wave we are using for testing,  $E_j \leq 2A f_o \Delta t < \frac{A}{2^{q-1}}$ . With this equation, we can calculate the maximum noise created by jitter, assuming we are passing in a sine wave with an amplitude of 1 Volt. A plot of  $\Delta t$  is also attached to ensure that jitter is relatively small at frequencies below 48 kHz. This has been plotted for several standard audio bit depths [8]. These charts show that a 24-bit data value will give minimal jitter in the range of 0 Hz to 48 kHz, and this makes sense because jitter takes major effect at extremely high frequencies [7].



**Figure 6. Jitter SNR vs. Bit Depth**



**Figure 7.  $\Delta t$  vs. Frequency**

## 3 Project Differences

### 3.1 Overview

When we went through the process of picking an old project to redesign, this one stood out to us because there were so many things that could be done in a different (and better) way.

Group 42's project from Spring 2003 did a good job at addressing the problem they introduced and finding a realistic and feasible solution. They decided that the best way to implement this wireless guitar effect system was by way of standard Ultra-High Frequency (UHF) radio transmission. In addition, the IC that they decided to base their project around (the Motorola MC13190) was a really good option that was capable of doing exactly what they wanted to do.

Their plan was to run the guitar's "dry" output directly into a transmitter unit. Within this transmitter unit, their guitar signal went through a low pass filter (to ensure that aliasing of the signal would not happen), into an ADC, into a microcontroller, and transmitted to the receiver via the Motorola RF IC. Their transmitter's microcontroller was responsible for receiving the data from the guitar's output and receiving data from the end user's adjustments of the effect parameter potentiometers.

The transmitter's output was then received by the receiver, where it went into another Motorola RF IC, into a second microcontroller, into a DAC, into a "Routing Unit", and finally out to an amplifier or PA system. This second microcontroller in the receiver unit was responsible for receiving the digital signal from the transmitter containing the guitar's dry output and the user's effect parameter settings. The microcontroller sent the dry guitar signal to the DAC, but it sent the user's effect parameter settings to the "Effects Loop," which Group 42 said would "mix and route the analog signal out of the DAC to different effects loops, and eventually out to the speaker output. The routing is determined by the signals that the MCU reads in from the transmitter [5]."

There are several key differences between Group 42's proposed solution and ours:

First, Group 42's solution relied on standard UHF radio transmission. We believed that this method of transmission was inherently problematic due to the possibility of interference, as well as the possibility of transmission interception or hacking. In a big concert venue, there can be catastrophic impacts because of RF signal interference, bad reception, and loss of connection between transmitter and receiver. Because of these things, we decided that transmitting and receiving data via Bluetooth was a far better option. Bluetooth establishes a secure transmission channel between master and slave. For our purposes, the receiver would be the master, and the transmitter would be the slave. Once two devices are paired via bluetooth, they only listen to each other. This makes interference problems with Bluetooth significantly less than interference problems with RF.

The next difference between the 2 projects is that Group 42's design required an uninterrupted connection between the transmitter and the receiver. Again, this is a problematic approach due to the fact that RF is susceptible to interference. However, because we cannot completely rule out interference between

Bluetooth transmission either, we opted to split up the guitar's dry output connection and the effect parameter transmitter. This way, even if our bluetooth receiver and transmitter lose connection, the end user will still be able to output their guitar's signal to the effect board/receiver unit. The only thing they will lose is the ability to change effect parameters wirelessly, but this is a significant improvement from Group 42's design where their device would fail all together.

The next difference is the transfer protocols between the 2 projects. Group 42 chose to use a Motorola RF IC to transmit data. This IC also required data to be encoded using a "Manchester" encoding. Our group decided to transfer data via Bluetooth and the I<sup>2</sup>S transfer interface.

Another difference between the 2 projects is that Group 42's design required a separate "Routing Circuit" to mix effects with guitar signal. Our group thought that this decision was a bit too complicated and created an inflexible hardware design and decided that the guitar signal could be mixed with the desired effects by the microcontroller alone, rather than requiring a separate unit. This will allow for our hardware to be able to support any kind of guitar effect instead of just a single effect

Finally, Group 42's design did not include a method to monitor system connectivity status. We decided that an LED module to monitor whether or not the transmitter and receiver were connected was a really important feature to include.

Group 42's Design	Our Design
<ul style="list-style-type: none"> <li>• UHF radio transmission (susceptible to attack)</li> <li>• Depended on uninterrupted connection between transmitter and receiver</li> <li>• Used Motorola MC13190 that required data to be sent with "Manchester" encoding</li> <li>• No real-time display for end user</li> <li>• "Routing Circuit" handles mixing dry guitar output with pedal effects</li> </ul>	<ul style="list-style-type: none"> <li>• Bluetooth <ul style="list-style-type: none"> <li>◦ Still UHF radio transmission, but inherently more secure because it establishes a Master/Slave Piconet for secure data transmission</li> </ul> </li> <li>• Still has physical connection from guitar to receiver unit (via 1/4" TRS cable) in order to negate impact from lost connection between transmitter and receiver</li> <li>• Uses I<sup>2</sup>S audio data transfer interface</li> <li>• Includes LEDs on transmitter unit for real-time display for end user</li> <li>• Microcontroller mixes pedal effect with dry guitar output in microcontroller</li> </ul>

**Figure 8** - highlight of the key differences between the 2 designs.

## 3.2 Analysis

For this analysis, we wanted to point out specifically why Bluetooth is a better wireless transmission method than standard UHF radio frequency.

Bluetooth, established as IEEE standard 802.15.1, is an inherently more secure means of wireless transmission than RF. It works by establishing a “piconet” which is an ad-hoc network specifically for bluetooth connections [9]. It is a master/slave architecture. This means that for our purposes the guitar transmitter device will be the master and the receiver will be the slave. This will allow our “master” (the transmitter) tell the “slave” (receiver) what to do, in the form of changing guitar effect parameters. Once a piconet personal area network (PAN) is created, slave devices will only listen to their master and vice-versa. This is the main reason why Bluetooth is superior to standard RF. Bluetooth is not nearly as susceptible to interference and attack as an RF transmission is.

Comparison between Wi-Fi, Bluetooth, ZigBee, RF and GSM/CDMA

Feature	Wi-Fi (IEEE 802.11B)	RF (ISO/IEC 14443-2)	Bluetooth (IEEE 802.15.1)	Zigbee (IEEE 802.15.4)	GSM/GPRS CDMA/1X RTT
Radio	DSSS#	FHSS	FHSS*	DSSS	-
Data Rate	11 M bps	150 kbps	1 M bps	250 kbps	64-128+
Nodes Per Master	32	65	7	64000	1
Slave Enumeration Latency	Up to 3s	Up to 3s	Up to 10s	30 ms	-
Data Type	Video, audio, graphics pictures, file	Compatibility with print, digital and magnetic media	Audio, graphics, pics, files	Small data packet	Wide Area Voice And Data
Range(M)	100	35	10	70	10+
Extendibility	Roaming possible	Yes	No	Yes	Yes
Batter Type	AAA battery power	Lithium-ion polymer	Li-ion battery pack	alkaline batteries	Nickel cadmium
Battery Life	Hours	6 to 12 months	1 Week	>1 year	1.7
Complexity	Complex	Simple	Very complex	Simple	Simple
Frequency	2.4GHz	3kHz-300 GHz	2.4GHz	850-930MHz	1,710–1,785 MHz
Security	Authentication service set ID(SSID)	-	64 bit,128 bit	128 bit AES and application layer user defined	-
Operating Environment	+32°F to +104°F	32° F to 122° F	32° F to 122° F	0 to 55°C (32 to 131°F)	- 30 °C ~ + 85 °C
Success metrics	Speed ,flexibility	Compatible	Compatible, easy to handle	Reliability , power ,cost	Reach, quality

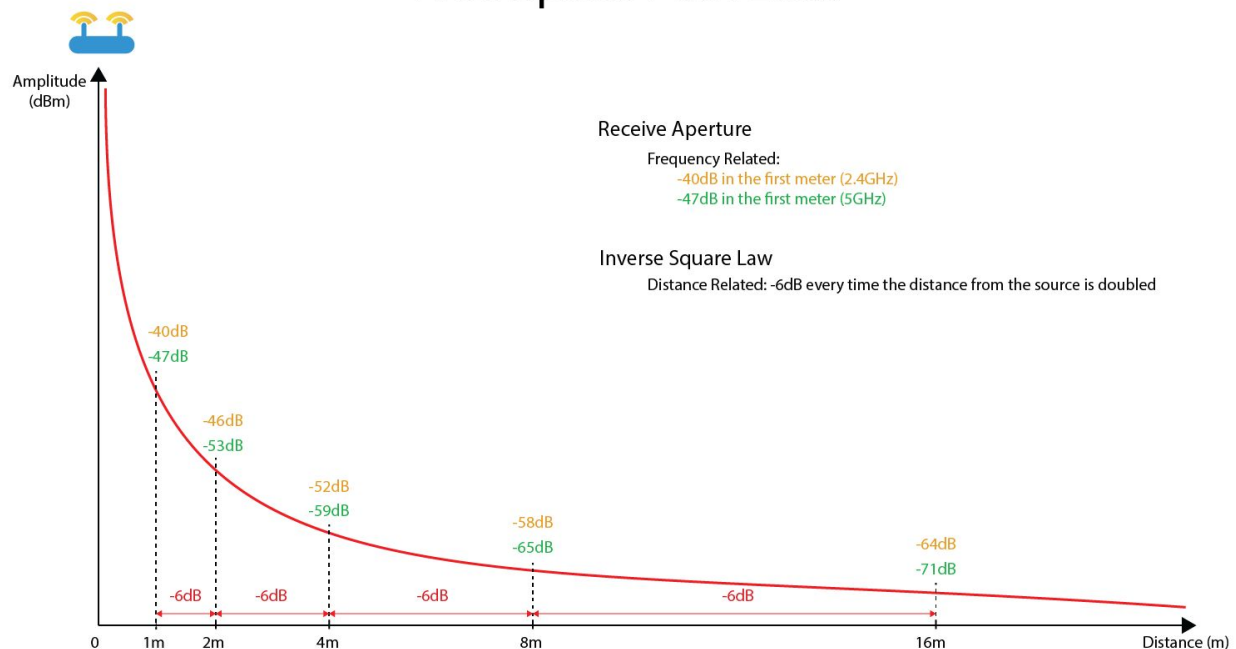
#DSSS,direct sequenced spread spectrum, \*FHSS,frequency hopping spread spectrum.

**Figure 9** - Comparison between several different wireless transmission methods (*see specifically differences between RF and Bluetooth*) [10]

Above in Figure 9, we can see a comparison of several different wireless transmission methods. Obviously, for our purposes we are only concerned with the “RF” and “Bluetooth” columns. It is shown that both Bluetooth and RF use what is called a Frequency Hopping Spread Spectrum (FHSS). This frequency hopping allows for connections to “hop” around to many different frequencies every second to

avoid complications from competing signals [9]. However, due to the piconet established with a bluetooth master/slave connection, Bluetooth is still a far less susceptible wireless transmission method. For Bluetooth, FHSS is an added level of connection integrity due to the establishment of a piconet, unlike standard RF transmission, where it is the only method to maintain connection integrity. Next, we see that transmission rates are faster via Bluetooth than RF according to Figure 9. This will allow for fewer problems with latency and will be a much better solution than the 150 Kbps. Both of these data rates are rather slow compared to a standard wireless internet connection (at least 60 Mbps in most places around the U.S.), but even a 1Mbps data rate works for master/slave short range communication. Finally, the most important advantage that Bluetooth has over standard RF is the fact that Bluetooth has prebuilt security. In the “Security” row of Figure 9, we see that RF has no security whatsoever, whereas Bluetooth can have either 64 bit or 128 bit security. This encryption is a proprietary encryption handled via the respective Bluetooth transceiver IC chosen.

## Free Space Path Loss



**Figure 10** - Free Space Path Loss graph for 2.4GHz and 5GHz wireless signal. [11]

The biggest improvement that our design has over the original design is the fact that Bluetooth is an inherently more secure data transmission method than standard RF. However, Bluetooth is still just as susceptible to signal attenuation as RF because of the fact that they are both 2.4GHz RF signals. Figure 10 above shows a Free Space Path Loss (FSPL) graph of a 2.4GHz and 5GHz WiFi signal. Because WiFi is also a 2.4GHz RF transmission, this same data can be used to show Bluetooth attenuation. This graph shows how a 2.4 GHz signal’s amplitude changes for a given distance. We see that amplitude drops from “unity” (0 dB) to -40dB at only ~1 m, and continues to follow the inverse square trend as distance increases [11].

Another major improvement between the old design and our design is the implementation of digital signal processing for handling effects. Group 42 originally proposed an analog effect circuit that handled the signal processing aspect of modifying their guitar sound (i.e. applying effects). The decision to go with an analog based signal processing system is problematic in and of itself. It limits the ability to program effects because it depends on physical passive components. By implementing a DSP system with our microcontroller, we will be able to program all kinds of different effects (digital delay, phasor, chorus) with relative ease. Changing a resistor value is as simple as modifying a few variables in program code instead of swapping out physical components in a circuit, making our device extremely flexible for many different applications.

## 4 Costs and Schedule

### 4.1 Costs

Our estimated working cost will be \$36/hr, with an average workload of 8 hours per teammate. We plan on completing 80% of the final design in one semester, and we are excluding the costs of building a casing for the guitar pedal:

$$3 * \frac{\$36}{hr} * \frac{8 hr}{week} * \frac{16 weeks}{semester} * \frac{1 semester}{0.8} = \$17,280$$

The cost for building a Wireless Guitar Pedal is as follows:

Part	Cost(prototype)	Cost(Bulk)
STM32F407 Microcontrollers	\$2.50	\$1.75
Passive Components (capacitors, resisters, diodes, voltage regulators, potentiometers)	\$20.66	\$15.88
AK5720 ADC	\$1.66	\$0.66
AK4430 DAC	\$0.65	\$0.33
Oscillator Chip	\$1.92	\$1.05
Bluetooth Transmitter/Receiver Module	\$20	\$18.95
PCBs (from PCBway)	\$10.15	\$5.80
Total	\$57.54	\$44.42

Thus, the total cost of our project will come out to be \$17,337.50

## 4.2 Schedule

Date	Marco	Will	Chris
<b>Week 1</b>	Order Components for receiver	Order components for transmitter and bluetooth receiver	Research Microcontroller and build power module and infrastructure to power the microcontroller
<b>Week 2</b>	Start developing PCB for receiver	Research Bluetooth chip and understand circuit schematic	Develop schematic for potentiometer system and begin development of PCB for transmitter
<b>Week 3</b>	Complete the development for PCB of receiver	Order PCBs and begin developing a test for SNR	Finish PCB
<b>Week 4</b>	Research Method to set up audio streaming inside of microcontroller	Continue developing a procedure to test SNR in audio	Research bluetooth communication and how to program microcontroller
<b>Week 5</b>	Solder Receiver PCB and begin testing the board for operability	Solder transmitter pcb together and ensure that the board is powering everything properly	Begin trying to test the bluetooth module, and get data transmitting across bluetooth on arduino
<b>Week 6</b>	Beginning trying to program microcontroller on receiver	Begin programming the microcontroller on the transmitter to receive data from the ADC	Move bluetooth code over to embedded C that can be programed into STM Microcontroller

<b>Week 7</b>	Continue debugging software and coding for receiver	Continue programming the microcontroller and begin incorporating bluetooth code into the controller as well	Help Will debug bluetooth code for transmitter microcontroller
<b>Week 8</b>	Begin working on bluetooth code for receiver	Perform validations testing for the transmitter	Help Marco debug the bluetooth code for the receiver
<b>Week 9</b>	Begin code for basic effect for demo	Perform SNR testing for receiver	Begin working on Final Report
<b>Week 10</b>	Week to Debug	Week to debug	Continue Final Report
<b>Week 11</b>	Work on Final Report	Work on Final Report	Work on Final Presentation

## 5 Ethics and Safety

After careful consideration, there are only a few ethical and safety concerns regarding the pursuit of this project. One potential safety/security risk that could be considered is the ability for the bluetooth communications between the transmitter and the receiver to be maliciously intercepted. A malicious attacker attempting to disturb a band/musician's performance could intercept and/or alter the data being sent to the receiver resulting in unwanted/bad sounding output from the amplifier. However, because this device only deals with guitar effect parameter data, a compromise in this data transmission would not be as detrimental as if other wireless bluetooth data was intercepted such as a classified document. Nevertheless, in this case, we will work to ensure that statement 9 of the IEEE Code of Ethics is satisfied, in the fact that we will work to ensure that there is no injury in reputation to any person or their property in the event that important data is not compromised from the use of our device [12]. In the event that our device's bluetooth transmissions are being tampered with, we can always switch the transmitter/receiver system off and continue to allow the guitarist to change effects manually.

Another concern that comes to mind with a system like this is equipment damage. We would not want our device to cause an amplifier to be overdriven to the point of failure. However, after reading through several guitar forums [13], we have come to the conclusion that this will not be a problem. At its worst, an incredibly distorted signal would be simply that: a distorted signal. Guitar amplifiers are designed to handle a ton of feedback and distortion, unlike a standard public address speaker system. Because of this research, we have concluded that our device will be in total accordance with statement 3 from the IEEE Code of Ethics in the fact that the available data tells us honestly and realistically that an end user's guitar amp will not be damaged from the possibility of an immensely distorted signal. Our device will also be in accordance with statement 9 from the Code of Ethics in the fact that no injury of property will occur from the usage of our device.

Finally, in regards to environmental ethics, we will fully comply with statement 1 from the IEEE Code of Ethics by ethically designing our system and developing it with sustainability in mind [9]. For this reason, it is important to discuss and be aware of the resources required to manufacture the Integrated Circuits found in our design. Stanford University reported that, "it takes roughly 10 gallons of water to make a single computer chip" [14]. When developing our system, we will try to be conscientious of waste and try to minimize it throughout. Although, because of the fact that we are not actually building this system physically, environmental factors are not really an issue.

## 6 References

- [1] Unknown Poster, "How are guitars hooked up to speakers at concerts?" Accessed: 1APR2020. [Online]. Available: [https://www.reddit.com/r/Guitar/comments/1ta0nn/how\\_are\\_guitars\\_hooked\\_up\\_to\\_speakers\\_at\\_concerts/](https://www.reddit.com/r/Guitar/comments/1ta0nn/how_are_guitars_hooked_up_to_speakers_at_concerts/)
- [2] Hugo, Strymon, "Setting Up Your Effect Signal Chain" 29FEB2016. Accessed: 1APR2020. [Online]. Available: <https://www.strymon.net/setting-up-your-effect-signal-chain/>
- [3] "Source Audio Hot Hand 3 Wireless Effects Controller Pedal," Accessed: 1APR2020. [Online]. Available: [https://www.sweetwater.com/store/detail/HotHand3--source-audio-hot-hand-3-wireless-effects-controller-pedal?mrkgadid=3331289787&mrkgcl=28&mrkggen=gpla&mrkgbflag=0&mrkgcat=guitars&&acctid=21700000001645388&dskeywordid=92700046934877524&lid=92700046934877524&ds\\_s\\_kwgid=58700005283398557&ds\\_s\\_inventory\\_feed\\_id=97700000007215323&dsproductgroupid=373037010169&product\\_id=HotHand3&prodctry=US&prodlang=en&channel=online&storeid=%7bproduct\\_store\\_id%7d&device=c&network=g&matchtype=&locationid=%7bloc\\_physical\\_ms%7d&creative=332063179833&targetid=pla-373037010169&campaignid=1709882817&gclid=EAIaIQobChMI3dPf4LPG6AIVWPfjBx12jAtEEAQYAyABEgLjcvD\\_BwE&gclsrc=aw.ds](https://www.sweetwater.com/store/detail/HotHand3--source-audio-hot-hand-3-wireless-effects-controller-pedal?mrkgadid=3331289787&mrkgcl=28&mrkggen=gpla&mrkgbflag=0&mrkgcat=guitars&&acctid=21700000001645388&dskeywordid=92700046934877524&lid=92700046934877524&ds_s_kwgid=58700005283398557&ds_s_inventory_feed_id=97700000007215323&dsproductgroupid=373037010169&product_id=HotHand3&prodctry=US&prodlang=en&channel=online&storeid=%7bproduct_store_id%7d&device=c&network=g&matchtype=&locationid=%7bloc_physical_ms%7d&creative=332063179833&targetid=pla-373037010169&campaignid=1709882817&gclid=EAIaIQobChMI3dPf4LPG6AIVWPfjBx12jAtEEAQYAyABEgLjcvD_BwE&gclsrc=aw.ds)
- [4] Dave Burrluck, "Manson Guitars MB-1 Standard Matt Bellamy Signature Guitar Review," 12APR2010. Accessed: 2APR2020. [Online]. Available: <https://www.musicradar.com/reviews/guitars/manson-guitars-mb-1-standard-matt-bellamy-signature-guitar-241595>
- [5] J Miller, J Smith, E Bellas, "Wireless Guitar Unit," 12FEB2003. Accessed: 1APR2020. [Online]. Available: <https://courses.grainger.illinois.edu/ece445/projects.asp>
- [6] Lee H. Goldberg, "Choosing The Right MCU For Your Audio Applications," 05MAY2012. Accessed: 14APR2020. [Online]. Available: <https://www.digikey.com/en/articles/choosing-the-right-mcu-for-your-audio-application>
- [7] "Analog-to-Digital-Converter," Accessed 14APR2020. [Online]. Available: [https://en.wikipedia.org/wiki/Analog-to-digital\\_converter#Resolution](https://en.wikipedia.org/wiki/Analog-to-digital_converter#Resolution)
- [8] Bonnie Baker, "Jitter and the Ins and Outs of SNR," 21JAN2010. Accessed: 14APR2020. [Online]. Available: <https://www.edn.com/jitter-and-the-ins-and-outs-of-snr/>
- [9] Concerning Reality. "How Does Bluetooth Work?," YouTube, September 16, 2019. [Video file]. Available: [https://youtu.be/cxP0Mdoz\\_Bo](https://youtu.be/cxP0Mdoz_Bo). [Accessed: April 11, 2020].

[10] R. Singh, P. Singh, and K. Yadav, "Wireless Communications Enlargement: A Review of Advancement in Technologies," *International Journal of Current Engineering and Technology*, vol. 4, no. 4, Aug. 2014.

[11] F. Vergès, "Free Space Path Loss Diagrams," SemFio Networks. [Online]. Available: <https://www.semfonetworks.com/blog/free-space-path-loss-diagrams>. [Accessed: 14-Apr-2020].

[12] "IEEE Code of Ethics," *IEEE*. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed: 01-Apr-2020].

[13] *Can you overload/damage an amp with too much distortion?* [Online]. Available: <https://www.soundonsound.com/forum/viewtopic.php?f=22&t=56495&p=507054&hilit=>. [Accessed: 01-Apr-2020].

[14] *Can computer chip makers reduce environmental impact?* [Online]. Available: <https://news.stanford.edu/pr/96/960605chipsenvir.html>. [Accessed: 01-Apr-2020].