**TuneBox**

Jake Reilly • Richard Bloomfield

Austin Hickman • Charles Filipiak

**Table of Contents**

1. Introduction 3
2. Problem Statement and Proposed Solution 3
3. System Requirements 3
4. System Block Diagram 4
5. WiFi Considerations 5
6. High Level Design Decisions 7
7. Open Questions 7
8. Major Component Costs 8
9. Conclusions 8

# Introduction

The electric guitar is one of the most popular instruments in the world and has been an iconic statement of rock music for generations. This instrument is often characterized by its ability to create numerous distinct sounds through the integration of amplifiers and signal distortion. However, for new guitar players, this key element comes at a price. Often what is required to manipulate the sound of an electric guitar is either an expensive amplifier or a complicated series of pedals that allow the user to switch between different effects using their feet. This is where the TuneBox will offer a solution.

The TuneBox will be an integrated microprocessor and amplifier device that digitally records the output from an electric instrument. Then, the signal will be run through a variety of different onboard amplifiers that allow the user to select different sounds that suit them. The TuneBox will also offer internet connectivity. With WiFi functionality and an associated mobile application, the effects played through the TuneBox will be selectable and controllable via smartphone. The app will also work with the microprocessor to analyze and display the pitch of each note, telling the user if the guitar is in tune. Finally, this internet connectivity will allow the user to record and upload their music to SoundCloud and other similar services for future playback.

# Problem Statement and Proposed Solution

Problem:

New and amateur musicians typically find it difficult to acquire electronic effects units. Commercially produced effects units are often expensive and confusing to operate. Furthermore, during practice sessions it is often difficult or otherwise undesirable to use one’s feet (the normal method during live performance) to toggle between different effects.

Solution:

We will design a system that allows for the recording, manipulation, and filtering of an electric instrument’s output signal. Furthermore, the device will assist the user in tuning his or her instrument.

# System Requirements

* Greater than 44 kHz sampling, memory writing, and reproduction
* Minimal unwanted signal distortion from amplifiers
* Flash memory that can store up to 10 minutes of continuous music
* 5V corded power supply
* No dangerously high voltages included in this system
* System plugs in through two ¼’’ female jacks similarly to other guitar effects pedals
* Output voltage needs to be compatible with most guitar amplifiers (±1.5 VAC)
* Needs to hook comfortably on instrument strap
* Only user interface is the app
* The app is downloaded and automatically connects with the TuneBox
* Wireless connectivity with one smartphone at a time
* App is able to communicate/upload to www.soundcloud.com

# System Block Diagram

## Overall System:



## Subsystem and Interface Requirements:

Following the flowchart given above, each block is representative of one subsystem. The adder consists of a simple op-amp circuit designed to step up the instrument-level input voltage to a level usable by the microcontroller and other circuitry. Next, the analog to digital converter we are using is contained in the kit board and is described by both the hardware spec sheet and the ADC software assignment we’ve previously completed. As we prototype our design, we plan to use the microcontroller included in the kit board, again using coding knowledge we’ve developed during the software assignments. The microcontroller will be used for digital signal processing and reception and transmission of control signals and status information. Connected to this microcontroller will be a flash memory device capable of storing a significant amount of audio sampled at a bit-rate of 8 bits per sample. Also connected to the microcontroller will be the ESP8266 WiFi interface. The exact software requirements for this device are yet to be determined, and we could be using an Arduino IDE to program it. We intend to communicate with the WiFi chip via an SPI interface.

Once the audio signal has been processed in software within the microcontroller it will be sent out to an external DAC that we can communicate with through SPI. Once the audio has been converted back to an analog signal, it will be passed through a Butterworth low-pass filter to correct for the higher-than-audible frequencies and smooth-out the delta functions which have been created by the DAC. From the filter, the signal is sent to a software-controllable multiplexer where the signal can be sent to a number of different analog, hardware-based effects (distortion, overdrive, tremolo, etc.) Once the signal has been passed through the user-selected hardware effects, it will be demultiplexed and sent through another amplification circuit to bring it to a level compatible with standard instrument amplifiers.

## Future Enhancement Requirements

In the future, we would be interested in being able to process more of our signals digitally and therefore incorporate additional effects into the TuneBox. As a market product, this would allow us to eventually design downloadable effects that could allow the user to access a large library of sounds without having to purchase individual effects units. We would also be interested in the ability to control multiple units from a single app interface, which would allow sound engineers to control the sound and equalization of a group performance.

# WiFi considerations

The two major features of our project, recording and the application of sound effects, will both incorporate WiFi to enhance user experience. To record audio, the microcontroller will store the input signal in flash memory, which will then be transmitted to the phone application. The transmission process will be performed via WiFi. To select sound effects, the user will chose from a menu in the phone application. This selection will be sent to the microcontroller via WiFi, which will set the multiplexer accordingly.

## Connection Considerations

The device will need to connect to a user-chosen WiFi network and from there it must be able to communicate with the IP address of the phone being used to control it. Connection methods to secure networks are TBD.

## Data Flow

An analog-to-digital converter is used to take the input guitar signal and convert it to digital data. The data is recorded and sent in the form of bits.The amount of data sent will depend on the length of the recording, but will be on the range of 10-20 megabytes. The information will be sent from the board to the phone application when the user requests a transfer (via the app). There will be no server involved.



## ESP8266

The ESP8266 will provide WiFi connectivity but will not be able to process all of our information at the speed that we require. We are planning on using the ESP8266 ESP-05 because we only plan on using the SPI interface to interact with the microcontroller and send signals to a smartphone via WiFi.

## Protocol for Data Exchange

We will be using SPI to communicate with the flash memory, DAC, and ESP8266 chip. We will use WiFi protocols to connect to a smartphone device.

# High Level Design Decisions

*Broken down by subsystem and major interface, this section presents your high level design of each subsystem or interface.* ***Your design decisions should be guided by choosing options which best support you system requirements.***

Adder:

The guitar outputs an approximately 3Vpp voltage signal generated from the pickups. The adder circuit will add the 1.5V needed to read a positive voltage on the ADC using a simple op-amp and resistor combination. The op-amps were purchased due to their high bandwidth and frequent use in audio application in order to better preserve the signal.

ADC:

We will use the onboard ADC in the Pic32 controller and configure it similarly to the setup done in Software Assignment 8. We will configure one of the input pins to read analog values while the rest are digital to be used in SPI functions.

Flash:

Then we will read the data to a flash memory using SPI commands similar to that used in Software Assignment 6. We will need to cycle through memory locations instead of continually erasing bits.

DAC:

The DAC has been purchased and shown to work using an SPI interface as well. It is able to take digital values and transform them into an analog voltage.

Filter and Mux:

The op-amps have been purchased and implemented in a simple filter design to select audible frequencies. The multiplexer has not yet been purchased but will need to be able to be controlled by pins on the microcontroller that will in turn receive information from the Wifi module.

Effects amplifiers

The amplifiers will also use the same audio-specific op-amps to distort the signal and possibly step down the voltage before it is output to a speaker.

# Open Questions

* How well can we recreate an audio signal after digital processing?
* Will we be able to build professional-quality audio amplifiers, either with analog or digital processes?
* How many licks does it take to get to the center of a tootsie pop?
* Will we be able to connect to WiFi over a secure network like the one on campus?
* Will SoundCloud allow for uploading through an external app?
* Can we accurately recreate or approximate discrete impulses in order to translate them into an audible analog signal?

# Major Component Costs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part #** | **Manufacturer** | **Description** | **Price per Unit($)** | **Quantity** | **Total Price** |
| NJM4580D | NJR | IC OPAMP AUDIO 15MHZ 8DIP | 0.72 | 7 | 5.04 |
| 111PCS | REAN / Neutrik | Phone Connectors 1/4" STEREO VERTICAL REAN | 0.75 | 4 | 3 |
| MCP4921-E/P-ND | Microchip | IC DAC 12BIT SNGL W/SPI 8DIP | 1.97 | 2 | 3.94 |
| 377-2091-ND | Bud Industries | Breadboard | 8.4 | 1 | 8.4 |
| 3352T-103LF-ND | Bourns Ind. | POT 10K OHM THUMBWHEEL CERM ST | 1.53 | 10 | 15.3 |

The list above indicates the parts of which we’ve already taken delivery. These are being used in the prototype of the recording system which we will demonstrate at the end of first semester. Additional parts will have to be ordered next semester as we develop our low-level design. Major costs will include the custom circuit board containing the microchip and effects hardware, any materials and fees associated with 3D printing the outer casing, and any additional hardware, including the WiFi transceiver and other circuitry components. All together, we anticipate that costs will not exceed $250. Development of the mobile application will not cost anything and we’ve already got a guitar and amplifier on hand for testing.

# Conclusions

We’ve been asked to demonstrate, by the end of first semester, the ability to record audio to memory and then play it back out. Once this demonstration has been completed, we will begin looking into how to connect WiFi to the microcontroller and we will begin developing the mobile application which will be used to control the entire device. After that, we will likely design a PCB containing the microcontroller, the analog effects, and the other subsystems mentioned above, as appropriate. Further design decisions and flow have yet to be determined.

**Resources**:

DIY Kits Similar to our project

* [Homebrew Digital Effects Pedal](http://www.instructables.com/id/Homebrew-Digital-Effects-Pedal/?ALLSTEPS)
* [Sound recorder/playback using limited RAM](http://blog.vinu.co.in/2011/07/attempt-to-access-memory-card-mmc-using.html)
* [Really helpful description and flowcharts but using a different microcontroller](https://www.adestotech.com/wp-content/uploads/doc1456.pdf)

Parts

* [Op-Amp that runs above 3V and BW of 1MHz](http://www.njr.com/semicon/PDF/NJM4580_E.pdf)
	+ [Link to order](http://www.digikey.com/product-detail/en/NJM4580D/NJM4580D-ND/673772)
* [Audio CODEC](http://www.cs.columbia.edu/~sedwards/classes/2008/4840/Wolfson-WM8731-audio-CODEC.pdf) from [Mouser Electronics](http://www.mouser.com/Search/Refine.aspx?Keyword=WM8731)
	+ [Link for purchase](http://www.mouser.com/ProductDetail/Cirrus-Logic/WM8731SEDS-V/?qs=sGAEpiMZZMtq3QB8qGen7WAQ3nuC248g%252b7lHmTcUSOA%3d)
* [¼’’ 2 Conductor Enclosed Jack](http://www.mouser.com/ProductDetail/REAN-Neutrik/NYS2343/?qs=sGAEpiMZZMv0W4pxf2HiV4x%2fcBMPvwwR0BZXYzbJ6Og%3d)
* [dsPIC33FJ256GP506](http://www.microchip.com/wwwproducts/Devices.aspx?product=dsPIC33FJ256GP506)
* Explanation of op-amp parameters: <http://www.ti.com/lit/ml/sloa083/sloa083.pdf>
* [DAC from Microchip](http://ww1.microchip.com/downloads/en/DeviceDoc/22248a.pdf)
* [ESP8266](https://nurdspace.nl/ESP8266#Building_the_gcc_toolchain)
	+ [Module comparison](http://blog.squix.ch/2015/03/esp8266-module-comparison-esp-01-esp-05.html)

Forums

* [Electro-Tech](http://www.electro-tech-online.com/)
* [Audio Recording and Playback](http://www.electro-tech-online.com/threads/audio-recording-playback-using-pic.41586/) within Electro-Tech
	+ I was reading this same one, they suggested using SD cards for storage, is that a better idea? → Depending on the type they can have faster writing speeds

Guitar Frequencies and Harmonics

* [Notes range from 80Hz-1200Hz](http://recordingology.com/in-the-studio/guitars/) but there are lots of harmonics above that range
* Vpp can get as high as 1V or slightly more, sample from -1.5V to 1.5V
* [Impedance matching](https://robrobinette.com/How_Amps_Work.htm#Input_and_Output_Impedance)

Application Notes

* [High-Quality Audio Using the PIC32](http://ww1.microchip.com/downloads/en/AppNotes/01422A.pdf)
* [ADC Description](http://extremeelectronics.co.in/microchip-pic-tutorials/using-adc-of-pic-microcontroller/)
* <https://chess.eecs.berkeley.edu/eecs124/reading/LeeAndVaraiya11.pdf>
* http://www.ti.com/lsds/ti/analog/webench/overview.page?DCMP=sva\_web\_webdesigncntr\_en&HQS=sva-web-webdesigncntr-vanity-lp-en