# Project Proposal: Guitar Buddy
**Project Number 15**
**TA: Channing Philbrick**

Austin Born and Chris Horn
September 20, 2018

## 1 INTRODUCTION

### 1.1 OBJECTIVE

Learning a new instrument can be challenging, especially when you don't have a proper instructor to teach you. Instructors can help students with reading music, hand and finger arrangements, speed and quality, and they often have the benefit of prior experiences from when they were still learning to play an instrument. Without a teacher, new learners have to take more time to check themselves and confirm that they are using proper techniques. Otherwise, they may learn to play incorrectly and not realize their mistake until much later in time. On the other hand, paying for private music lessons every week can be prohibitively expensive. Based on a 2014 national study by takelessons.com, music lessons typically range from $30 to $50 per hour[1]. If a student attends just one hour-long lesson each week, this adds up to between $1,500 and $2,500 per year. Students who cannot afford to take frequent private lessons are then at a severe disadvantage against students with private tutors who can check for proper technique as the student plays.

The Guitar Buddy system is intended to give learners on a budget the ability to gain real-time feedback about hand position and technique so they learn the proper way to play an instrument without needing to attend private weekly lessons. The Guitar Buddy apparatus consists of an LED array, Bluetooth-connected microcontroller, and other integral hardware that fits over the neck of a guitar. The system helps guitar players who struggle with proper technique by mapping LEDs to each finger position to visually instruct players where to place their fingers. We intend to develop a basic sheet music reader to convert musical notes to finger positions, and the apparatus will then wirelessly communicate with a Bluetooth-connected computer to display finger positions on the neck of the guitar. With this device, new players will not have to go through the daunting task of finding a tutor and taking expensive weekly lessons. Instead, students can learn from a digital instructor that is ready to teach any level, any time.

## 1.2 Background

According to TheGuitarLesson.com, some of the biggest problems new guitar learners face are knowing how to play chords, knowing which songs to practice, and finding time to practice[2]. The Guitar Buddy system helps students learn chords by giving visual cues as to where to put their fingers, making chord progressions an intuitive affair. To that end, the Guitar Buddy will include a wide range of examples for learners at every level of difficulty. Guitar Buddy also addresses two other major problems when learning the guitar: finding the time and money to attend private practice sessions, and receiving quality feedback while they are playing. Students with weekly lessons often only have that single hour per week to receive instruction from their music teacher, and they have no form of feedback for majority of their practice time. With Guitar Buddy, you always have a personal instructor ready to teach you.

## 1.3 High-Level Requirements

- LED array must be able to display chords in real-time with the music.

- Sensing module must be able to correctly sense a player's finger position with 95% accuracy.

- Guitar apparatus must have enough power to last a full instruction session of at least 2 hours.

## 2 Design

The high level block diagram is broken down into six main modules. The power supply module is responsible for the power management of the guitar, including charging and discharging circuits as well as additional safety hardware. The sensing module is responsible for detecting when the user depresses any strings along the neck of the guitar, and reporting that information to the primary microcontroller. The control module houses the primary microcontroller, an ESP32, and is responsible for communication with the software module, storing song information, and sending the control signals for driving the LEDs. The LEDs are driven by the LED output module, which is responsible for managing the 60 LEDS from only a few I/O pins on the primary microcontroller. The final module is the software module, which lives outside of the guitar apparatus and is responsible for processing music and sending the binary song data to the ESP32 over Bluetooth.
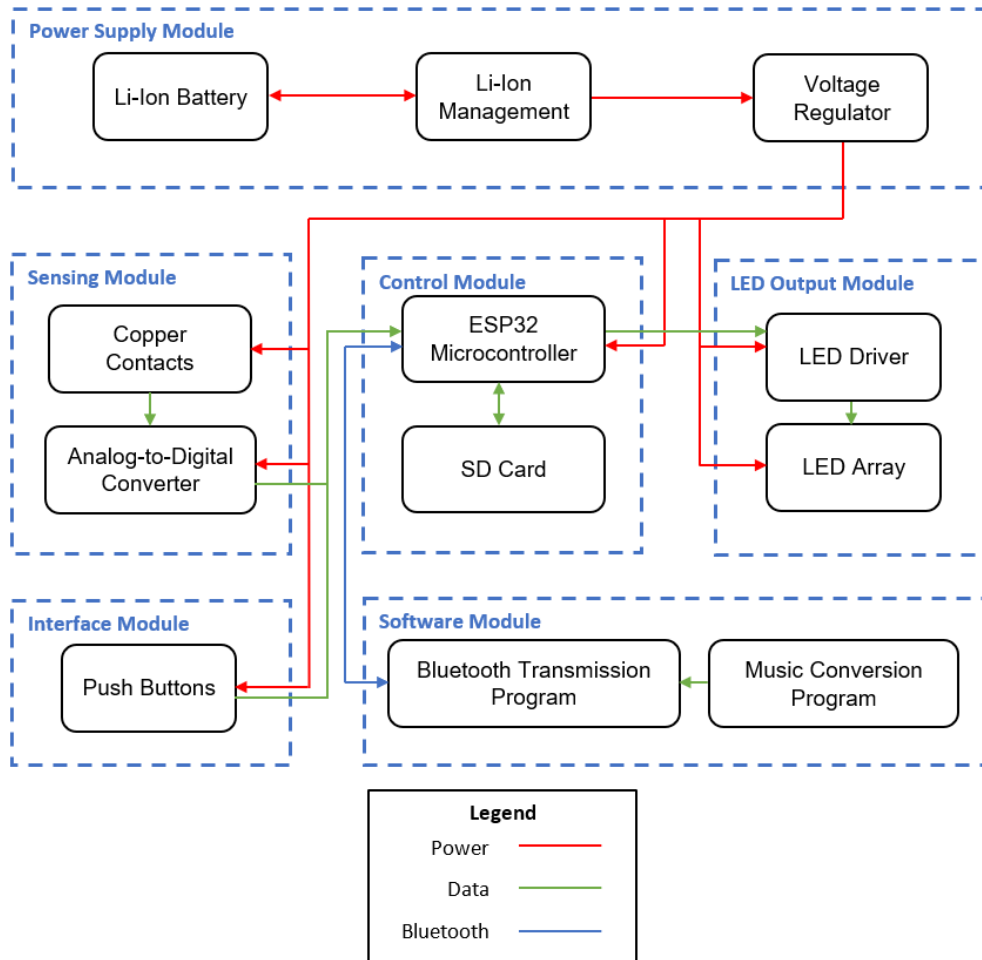
Figure 2.1: High-Level Block Diagram

## 2.1 CONTROL MODULE

The purpose of the control module is to manage communication with the software module via Bluetooth and to send control signals to the other modules. A microcontroller will be responsible for communication and generating control signals, while an additional flash storage is used for storage of the song data.

### 2.1.1 MICROCONTROLLER

An ESP32 microcontroller was chosen to be the primary microcontroller for the onboard electronics due to its low cost, built-in wireless communication (Wi-Fi and Bluetooth) capabilities, and its generous quantity of GPIO pins.

Requirements:

- Operate at 3.3V ± 5% with an $I_{max}$ of less than 500 mA during radio transmission and an $I_{typical}$ of less than 200 mA during regular operation

- Native compatibility with external SD storage

- 10+ GPIO pins for communication with other modules

### 2.1.2 FLASH STORAGE

A general purpose SD card has been selected for the flash storage option. The ESP32 has native compatibility with reading and writing to SD cards without the need for any additional hardware.

Requirements:

- 256 MB or larger storage capacity

- R/W speeds of at least 1 mbps

## 2.2 INTERFACE MODULE

The interface module is responsible for interacting with the user through the means of buttons mounted on the guitar. While away from the software module, the interface module will be the user's only way to control the settings of the guitar. *Reach goal: the interface module will also control the LEDs on the neck of the guitar as a 6-by-10 display to display information during song selection.*

### 2.2.1 PUSH BUTTONS

Push buttons will provide a tactile way for the user to navigate through the guitar settings while they are away from a laptop running the software module.

Requirements:

- Durable and reliable operation; >1000 click life span

- Low cost (<5¢ at bulk)

- Easy to use while holding the guitar in the playing position

## 2.3 LED OUTPUT MODULE

The LED output module consists of the LED driver and the LED array itself. The driver is responsible for turning a few control bits from the ESP32 into 60 individually addressable high current outputs. The LED array is distributed across the 60 LEDs located on the PCBs mounted along the neck of the guitar.

### 2.3.1 LED DRIVER

The LED driver is composed of four 16-bit constant current shift registers. The outputs of these shift registers are connected to the LEDs along the neck of the guitar. This allows the 60 LEDs to be controllable through only three GPIO pins: a shift input, a clock, and a reset. Constant current registers will also reduce the need for external hardware, and reduce cost by simplifying the design. The shift registers will be hooked up in series, meaning that they must have a high maximum clock frequency to support refreshing all 60 bits within 10 ms. *Reach goal: Extend hardware capabilities to support multicolor LEDs by increasing the number of registers from 60 to 180 (3 channels per LED).*

Requirements:

- At least 60 bits of storage (*180 bits for reach goal*)

- Minimum $f_{clock_{max}}$ of 100 kHz

- Minimum $I_{max}$ of 100 mA per channel

- Minimum $I_{max}$ of 1200 mA per chip

### 2.3.2 LED ARRAY

The LED array is spread out along the neck of the guitar. Each fret will house a PCB that contains an LED and some components for the sensing module. Each PCB will be connected to the LED drivers through wires running from the primary PCB on the body of the guitar to the header pins located on each fret's PCB. The LEDs must be small enough to avoid interfering with the user's ability to manipulate the strings. *Reach goal: single color LEDs will be replaced by RGB LEDs, giving the user the ability to vary the colors along the guitar.*

Requirements:

- $V_{on}$ of LEDs is less than 2.0 V

- LEDs are easily visible from at least 1 m away without being uncomfortablely bright

- LEDs are less than 2 mm tall (surface mount)

## 2.4 POWER SUPPLY MODULE

The power supply module is responsible for safely charging and discharging the battery. A Lithium-ion (Li-ion) battery will provide high power density, and the Lithium-ion battery management sub-module will ensure that the battery stays within safe operating conditions at all times. The voltage regulator will step down the variable Lithium-ion battery voltage to a constant 3.3 V for the rest of the modules.

### 2.4.1 LITHIUM-ION BATTERY

The Lithium-ion battery will serve as the sole energy source for the guitar. Due to the number of LEDs that may be driven at the same time, high power density is also an important factor for battery selection. The battery should be capable of powering the guitar for at least 2 hours of regular use.

Requirements:

- Peak $I_{out}$ must be at least 2 A (*Reach goal: using the LED array as a display during song selection would increase minimum $I_{out}$ to 4 A)*

- Minimum 1500 mAh capacity (*Reach goal: additional features would increase capacity requirements to 3000 mAh)*

- Weight must be under 250 g

- Battery must be under 10 cm along its largest axis, and less than 2 cm thick

### 2.4.2 LITHIUM-ION MANAGEMENT

Lithium-ion batteries provide large energy and power densities, but require specialized charging and discharging. A Lithium-ion battery management Integrated Circuit (IC) is used to monitor the voltage of the battery to protect against overvolting and undervolting, as well as monitoring the charging and discharging of current. It will also provide thermal management to avoid damaging the battery or presenting a fire risk.

Requirements:

- Capable of charging a single-cell Lithium-ion battery to 4.2 V (tolerance dependent on particular battery model)

- Charging current of at least 1 A

- Full safety suite including, but not limited to:
    - Undervolt and overvolt protection
    - Support for external thermistor to enable thermal monitoring
    - Short circuit protection

- Must not exceed 80 °C under maximum load

## 2.5 SOFTWARE MODULE

The software module is the only module that will not be located on the guitar. The software module's responsibilities include generating the bytecode for the LEDs on the neck of the guitar, as well as transmitting that data to the ESP32 via Bluetooth. *Reach goal: the software module will also automatically generate the bytecode from MIDI guitar files or sheet music.*

### 2.5.1  MUSIC CONVERSION PROGRAM

The music conversion program's role is to convert traditional music formats into a bytecode that is understandable to the ESP32. In order to reduce the computational requirements of the onboard microcontrollers, the bytecode will be broken down into 8-byte sections, with 60 bits representing the 60 LED states along with four parity and auxiliary bits. *Reach goal: the program will be capable of taking a MIDI file as an input and automatically transcribing it to bytecode without any additional user interaction.*

Requirements:

- Generate the proper bytecode and parity bits

- Capable of converting 5 minutes worth of song notes within 30 seconds

- User friendly and stable; difficult for user to accidentally crash while in use

### 2.5.2  BLUETOOTH TRANSMISSION PROGRAM

The Bluetooth transmission program is responsible for accepting the bytecode from the conversion program, verifying the integrity of the data (via parity bits), and then transmitting that data to the ESP32 on the guitar. The transmission program will also be the primary method for the user to change settings on the guitar. *Reach goal: include a GUI interface for adjusting the color of the LEDs (if applicable) as well as viewing a full list of songs, battery level, etc.*

Requirements:

- Verify and transmit bytecode song data to ESP32 accurately

- Transmission rate of at least 500 kbps at 5 m distance

- Provide command line interface for adjusting settings on guitar

- Provide command line interface for managing song data on the guitar

## 2.6  SENSING MODULE

The sensing module is responsible for detecting when the user depresses a string. This information is then used to track the user's progress in the song and highlight errors in the users inputs.

### 2.6.1  COPPER CONTACTS

Each fret contains a small board that houses the LEDs and 6 bare copper contacts (one under each string). These contacts are connected to 3.3 V through a large pull up resistor. When the user depresses a string, the conductive guitar strings will make contact with the pad. The guitar strings are connected to ground through a smaller (but still relatively large) pull down resistor. When the string makes contact with the pad, it will pull the voltage of the pad down

to near 0 V. By measuring the voltage of the pads through the use of an Analog-to-Digital converter (ADC), we are able to determine which locations the string is being depressed. Since the PCB is lower than the fret bars, they should not interfere with the functionality of the guitar.

Requirements:

- Thickness must be 1.5 mm ± 0.2 mm

- The copper contacts must be durable enough to not be damaged through regular playing of the guitar

- Closed circuit current must be below 1 mA

- Firm contact with a string must result in the voltage of the pad equalizing to <0.2 V within 5 ms.

### 2.6.2 ANALOG-TO-DIGITAL CONVERTER

The Analog-to-Digital (ADC) sub-module is responsible from probing the 60 copper contacts and determining which pads are in contact with a string. This will be done through an MSP430 microcontroller and additional support hardware. The MSP430 has an 8-channel ADC, capable of measuring 8 different sources at the same time. The MSP430 will use additional GPIO pins to control the input of a mux, which will allow the MSP430 to quickly switch between each fret, measuring all 6 contact pads at the same time. Information about which pads are in contact with the strings is then sent to the primary microcontroller, where the data is processed.

Requirements:

- Powered by 3.3 V ± 0.05 V

- Mux capable of switching at least 1 kHz

- ADC max settle time of 1 ms

- ADC precision of at least 0.1 V

- Minimum 16 GPIO pins (including 6 tied to ADC)

### 2.7 RISK ANALYSIS

The biggest concern with Guitar Buddy is the communication between the different hardware components. There are several data links between components that could set the project behind schedule if we do not address them early. The computer sends data over Bluetooth to the ESP32 to store in flash storage, the ADC converter in the sensing module sends wired signals to the ESP32, the microcontroller sends data to the LED driver, and the microcontroller needs to read from the flash storage when sending data to the LEDs. If any of these interactions

becomes problematic, it could set back the project significantly. It is especially difficult to predict the time spent debugging communications because there are a number of reasons why communications might stall or drop during operation. It may be that the hardware was not properly coded, but it could also be due to faulty connections (in the case of wired data lines) or interference (in the case of Bluetooth communication).

To mitigate this risk, when working on a specific module, we will test the communications between it and the other relevant modules before the final integration of each component. This will limit the potential for difficulties during final integration, which already seems to be the most precarious part of the project. In addition, we will take steps to ensure there is a standard verification procedure if communications fail somewhere in the data pipeline. By preparing a set of ordered checks before the final integration, we will maximize efficiency during the debugging phase when difficulties inevitably arise.

## 3 ETHICS AND SAFETY

As a consumer-oriented device, it is especially important that the Guitar Buddy system does not harm any users, other persons, other devices or other objects. In compliance with code 1 of the IEEE Code of Ethics, we will maintain safe engineering practices and will disclose and mitigate any potential safety concerns to the user[3]. One such safety concern is the Lithium-ion battery source for the apparatus. These batteries can be damaged if the temperature is outside the range of 0 - 130°C, and damage to the battery can result in potentially catastrophic failure and harm to the user[4]. Although it is intended primarily as an indoor device, we will incorporate a basic thermistor to verify that the battery is within normal operating range. In addition, we will check the Lithium-ion management system to ensure the batteries never exceed the rated voltage.

In addition, there is a potential safety hazard in connecting the guitar strings to the sensing circuit. To prevent any harm to the user, we will ensure that the strings do not discharge any electricity under normal operation, and we will limit the potential current from the sensors through the strings with high resistances in the sensing part of the circuit.

With the mapping of music to the LEDs, one difficulty is ensuring that the artists for songs are properly attributed for their musical works. In support of the writers of musical works used in this project, and following section 1.5 of the ACM Code of Ethics, we will give due credit to the writers of any songs we use for the Guitar Buddy system and will only use music available for public and/or educational use[5].

The software used in music conversion and Bluetooth transmission must also comply with proper industry standards. As per section 1.2 of the ACM Code of Ethics, we will, to the greatest extent possible, prevent harm to the computer's data and operation[5]. Although the scope of the programs we intend to build for this project are limited, we will take steps to ensure that any software used in the Guitar Buddy system does not harm any data nor functionality on the machine used to run it.

REFERENCES

[1] TakeLessons.com. What people pay for music lessons, annual report. [Online]. Available: https://support.takelessons.com/hc/en-us/article_attachments/200377329/What-People-Pay-for-Music-Lessons.pdf

[2] TheGuitarLesson.com. 5 biggest challenges learning guitar. [Online]. Available: https://www.theguitarlesson.com/guitar-lesson-blog/beginner-guitar-lessons/5-biggest-challenges-learning-guitar/

[3] IEEE.org. Ieee code of ethics. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html?WT.mc_id=lp_ab_ico

[4] BatteryUniversity.com. Lithium ion safety concerns. [Online]. Available: https://batteryuniversity.com/learn/archive/lithium_ion_safety_concerns

[5] ACM.org. Acm code of ethics and professional conduct. [Online]. Available: https://www.acm.org/code-of-ethics