UMassAmherst
The Commonwealth's Flagship Campus

LEDshred

Midway Design Review
Team 5
ECE 415
SDP 2021

Nov 18, 2020

The Team

Jordy Mukania

Electrical Engineer

Team Coordinator

Isaiah Provencher

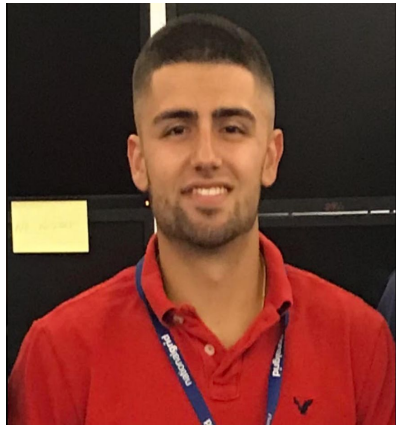
Electrical Engineer

Altium Lead

Jake Colapietro

Computer Engineer

Budget Management



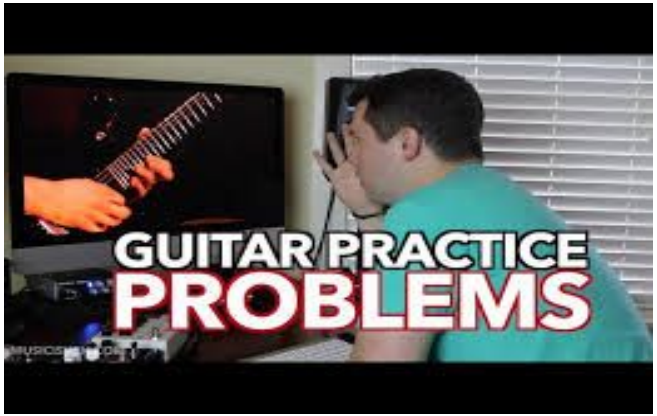
Kivan Daruwalla

Computer Engineer

Technical Responsibilities

Problem Statement

Many people are easily deterred as soon as they run into difficulties when playing a guitar. Designed for beginners, our product will use LED lights on guitar fretboards to light up chords in order to guide the user through a song. It will then offer feedback on their performance by checking if they played the correct notes. We are hoping that our product will help users play proficiently and that the feedback will further improve their play.

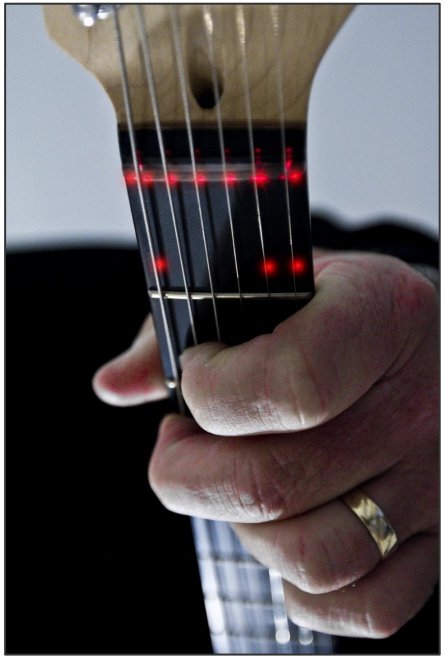


Frustrated guitar player

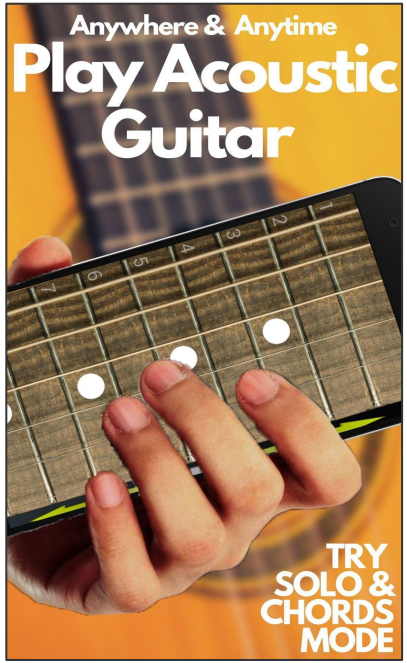
Visual Representation of Our Solution

1

2



3

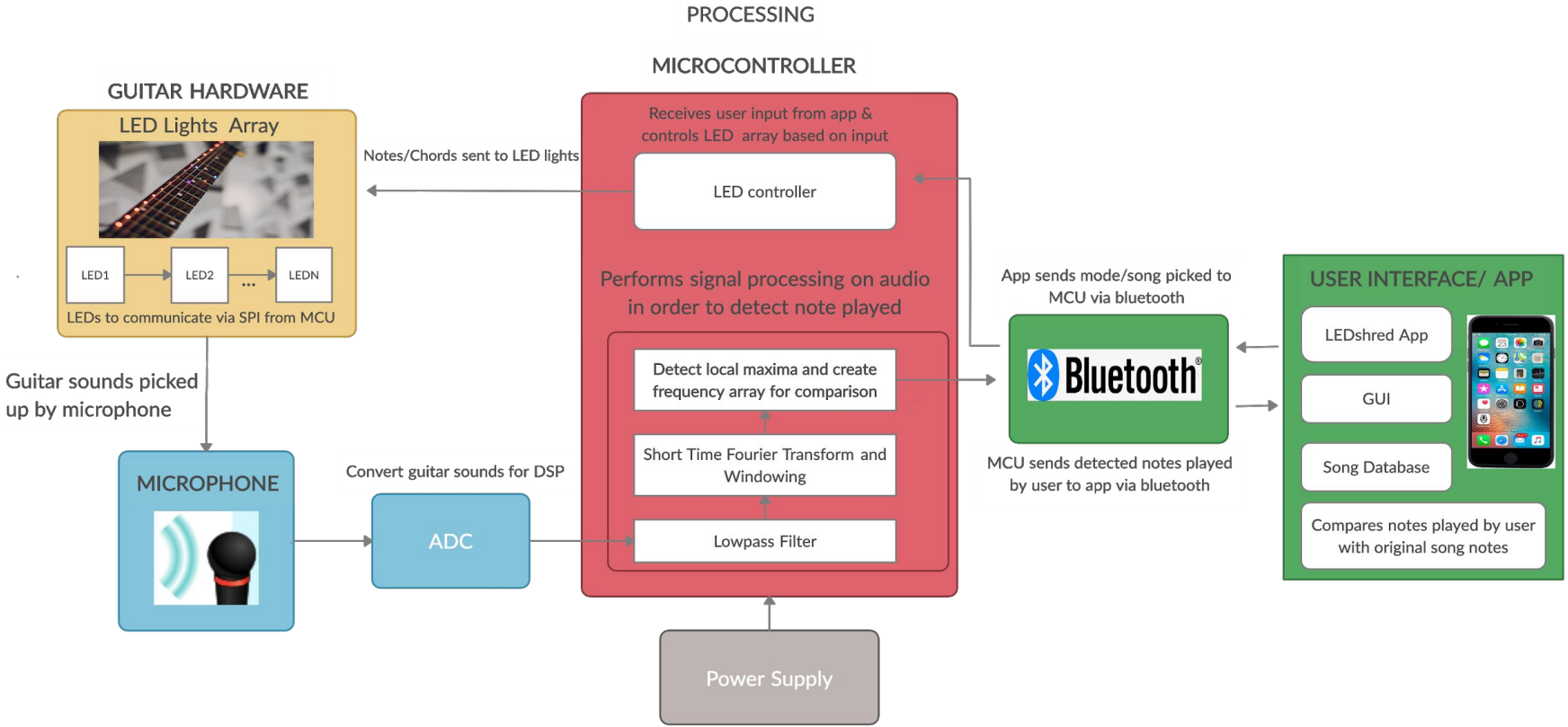


4

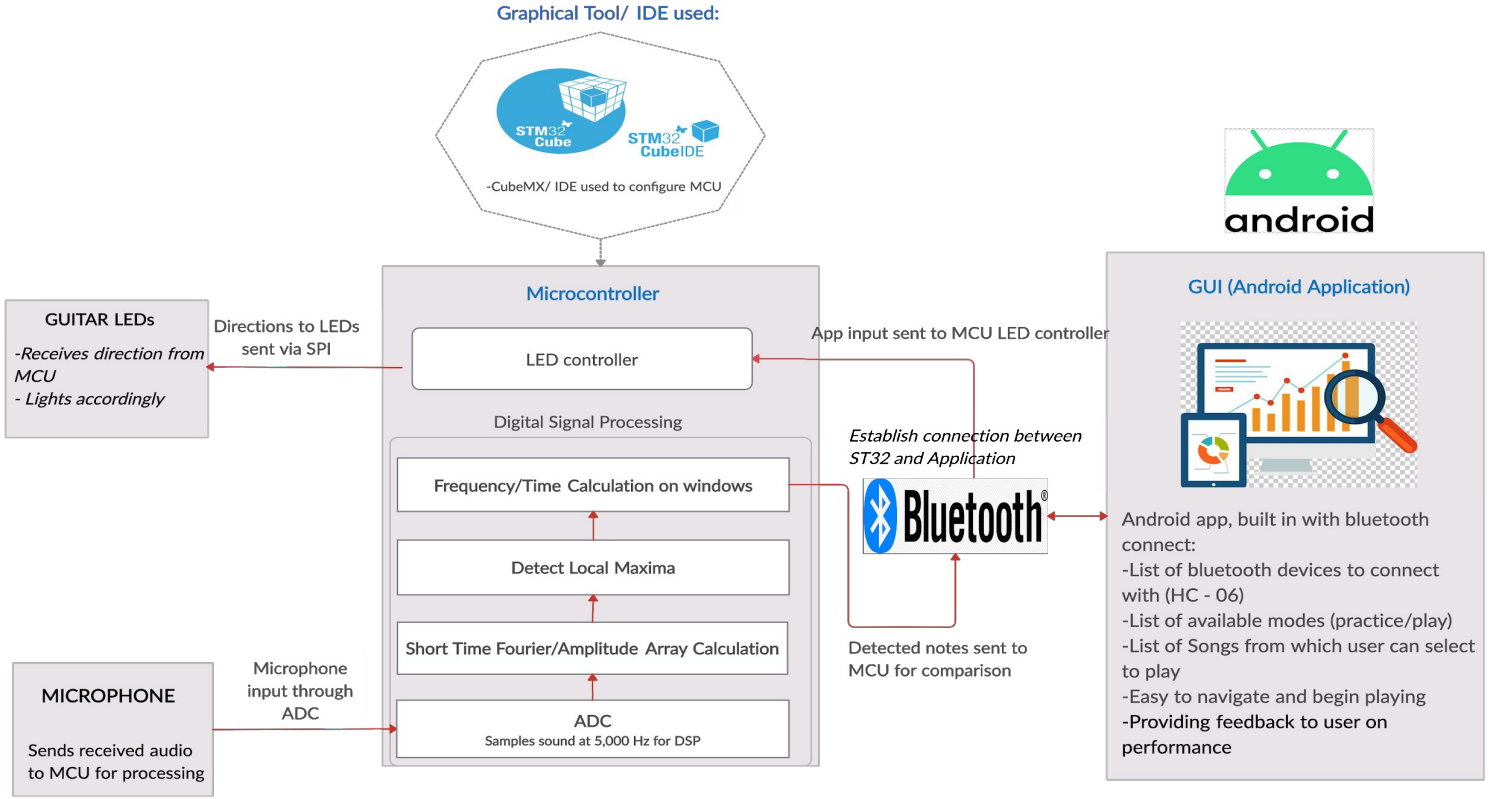
System Specifications

- LED lights attachable to any guitar (Typical 38 inch long guitar)
- 5 X 6 array of LEDs (5 rows with 6 LEDs across each strings)
- Height of LEDs less than 2.5 mm
- Accompanied by app which supports bluetooth connectivity to user device
- Samples guitar notes at 5,000 Hz
- Samples sound for ¼ of a second at metronome of given song
- Correctly distinguishes a note (open and chords) > 95%
- Offers user performance report within 5 seconds after play session is finished

Updated System Block Diagram



Updated Software Diagram

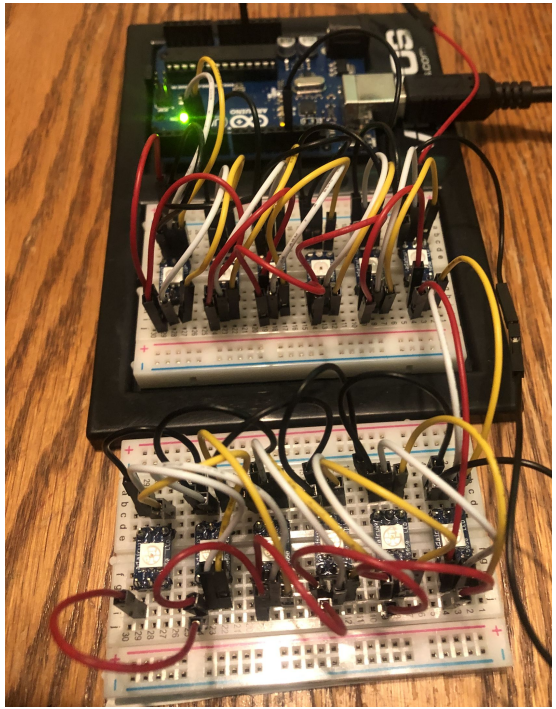


MDR Deliverables

Engineer	Deliverable
Jordy	Able to individually address LEDs over 2X6 matrix of LEDs (12 in total)
Isaiah	Ability to convert sound picked up by microphone into digital signal to be used by microcontroller
Jake	Create algorithm to correctly identify recorded single note frequencies > 80% of the time
Kivan	Fully functioning connection/communication between application and microcontroller / some app functionality

MDR Demo - LED Lights

- Full control over 2X6 array of LEDs
- All Individually addressable
- LEDs fulfill height specification
  - Height with LED and breakout board used less than 2.5 mm
- Able to use lights to “light up” corresponding guitar strings for notes on a guitar



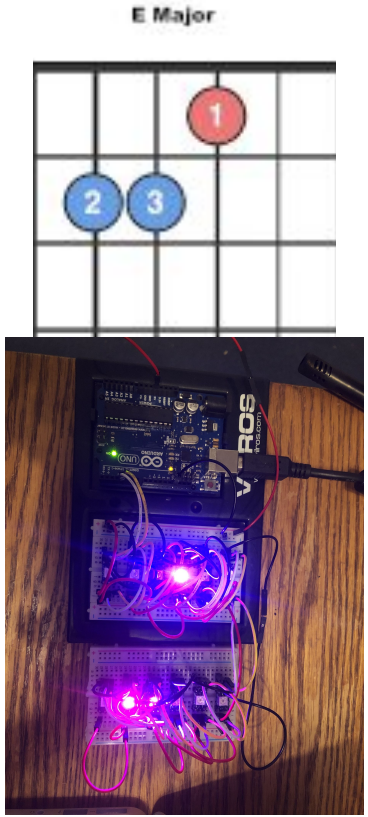
LED array implementation

LED Lights Recorded Demo

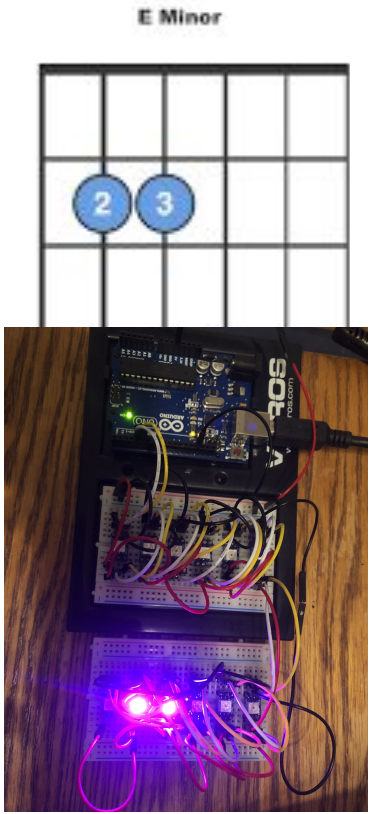
Lights Hardware (2)

Figures showing planned implementation for future

1



2

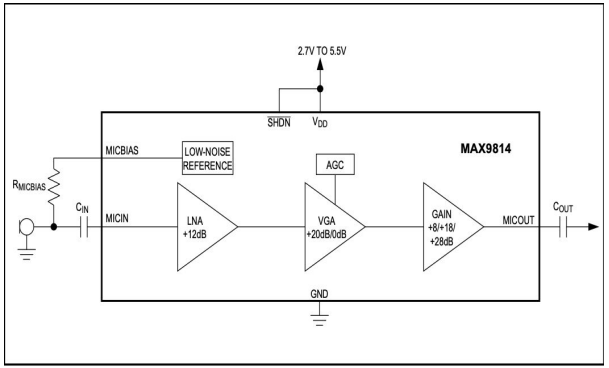


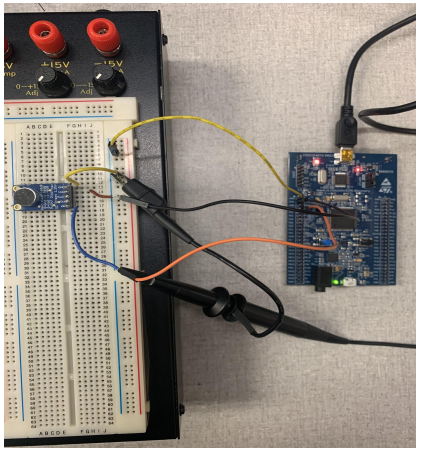
3

Microphone Circuit

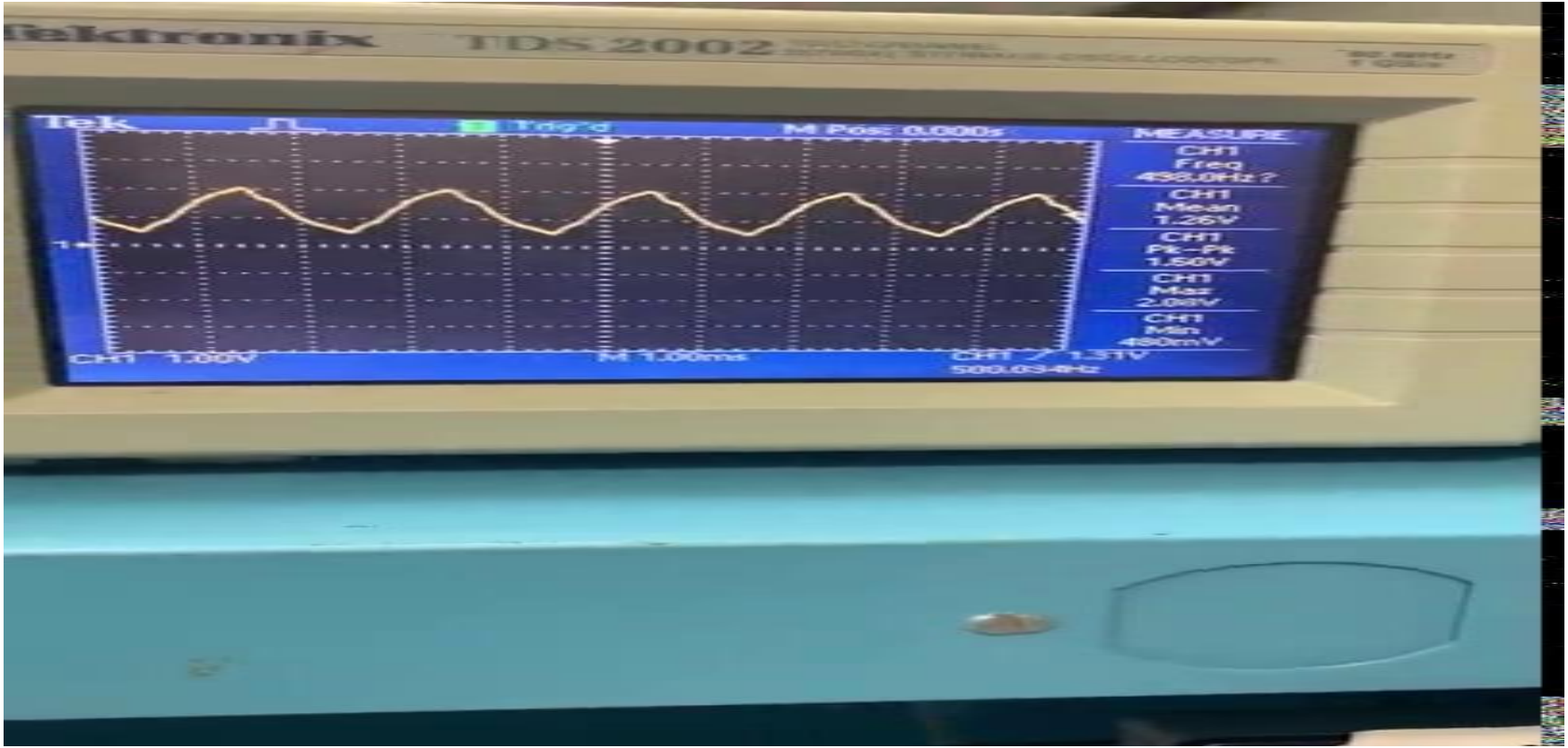
- Ability to convert sound picked up by microphone into digital signal to be used by microcontroller

- Input sinusoidal frequency tones into microphone and measured the output on an oscilloscope



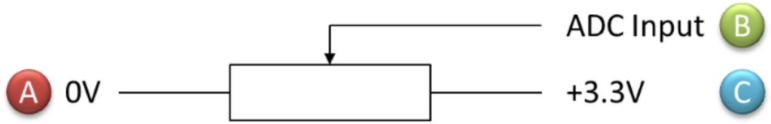


Microphone Demo

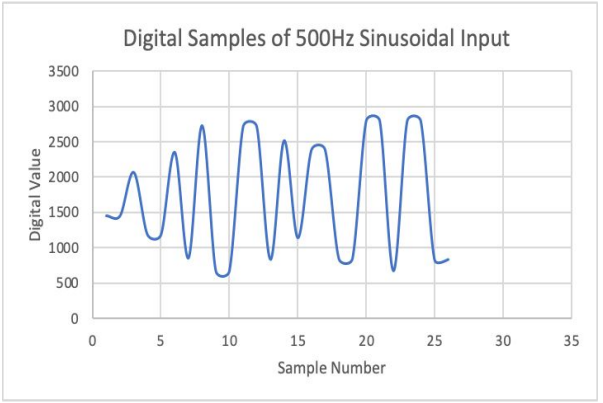


Analog Signal to Digital Conversion

- Used ADC pins on the Microcontroller
- Same process as testing the microphone output, but instead of plotting the output on an oscilloscope we sent the output to the ADC pin of the microcontroller
- Need the output of the microphone to fill as much of the space between 0V and 3.3V as possible for low quantization



ADC Demo


Digital Samples of 500Hz Sinusoidal Input
Digital Value
3500
3000
2500
2000
1500
1000
500
0
0
5
10
15
20
25
30
35
Sample Number

Audio Processing

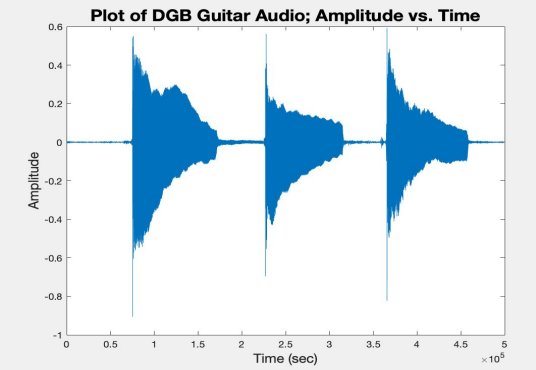
- As of right now we have a working Matlab Script that can take in an audio signal and produce arrays of time and frequency values
- The script is in Matlab but will be put on the micro in the future
- The accuracy was about 85%

1.6640	1.8773	2.1333	2.3467	2.6027
147.1722	147.0395	147.2200	147.1777	147.3548

Example of time and frequency values

Subsystem Diagram

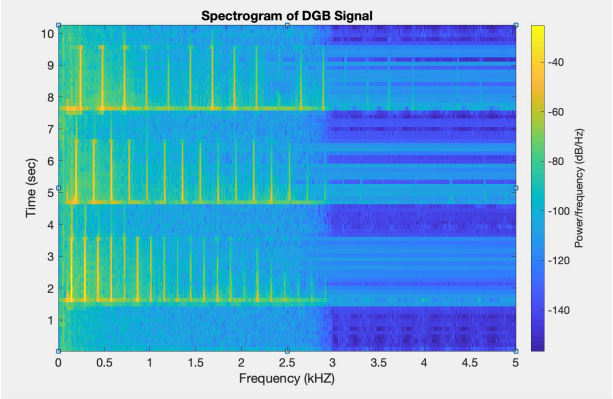
ADC



LPF

Windowed FFT

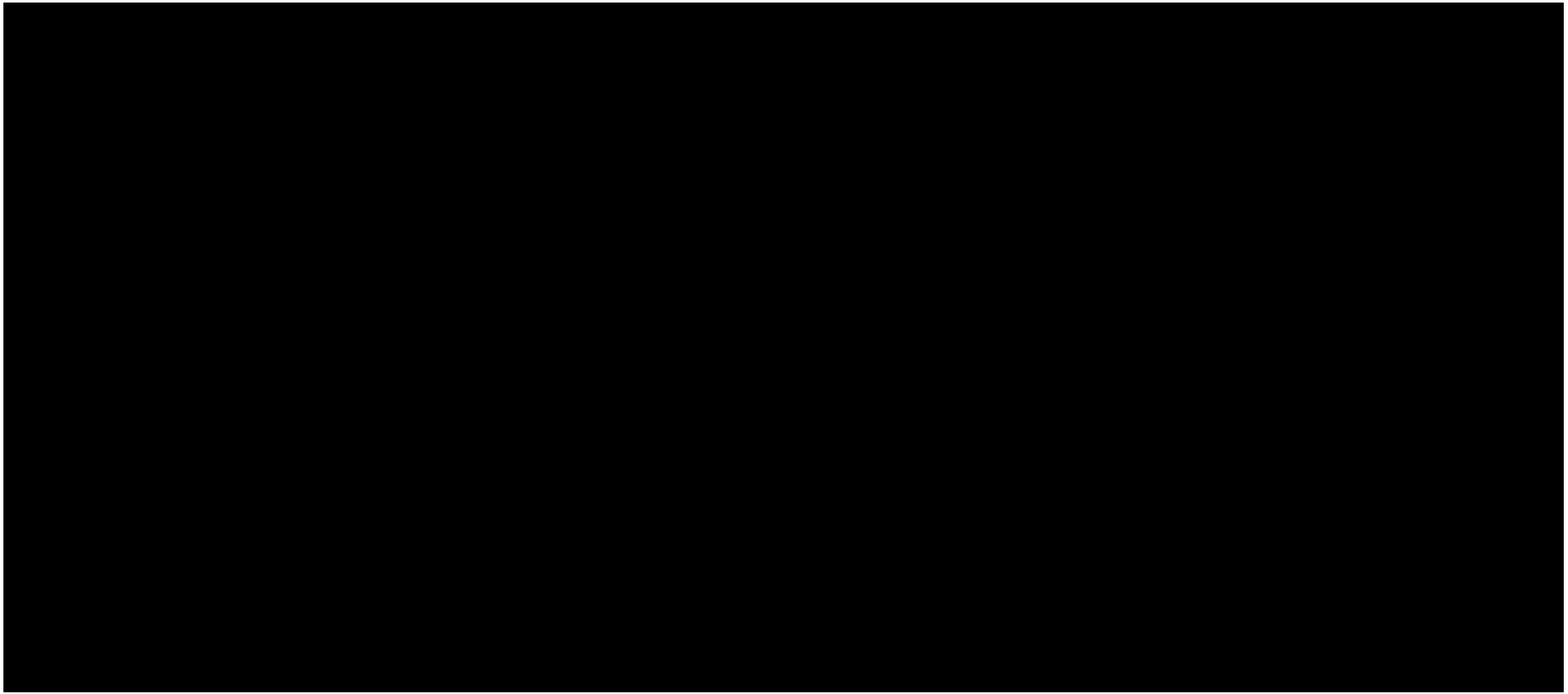
(Produces time localized frequency information)



Thresholding/Local Max Calculation

Frequency & Time Calculation

Demo



Results

- To test, 15 different multi-note recordings were put through the algorithm
- Each had different content and varying frets contained
- Overall, 93 frequency readings were pulled from the recordings in total
- Out of the 93 only 14 were outside an acceptable range (depending on note but at the lowest < 3 Hz) which yields 84.9% accuracy
- The average difference between the estimates and the expected was about 2.5 Hz

- *Kivan Daruwalla*

Android Application - Deliverables

- **Develop working connection between GUI application and C++ code for the MCU.** ✓
  - Show that we will be able to use the app to talk to the MCU

- **Create Functionality on the app itself** ✓
  - Illustrate some of the general properties the app will have

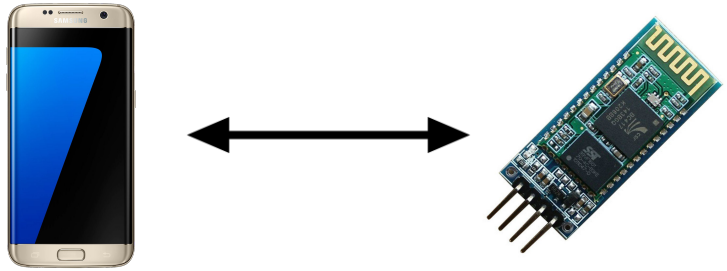
(No longer using Cloud Service in Project)

Android Application - App Connectivity

Developed a fully functioning connection between Android app / HC - 06 bluetooth module.

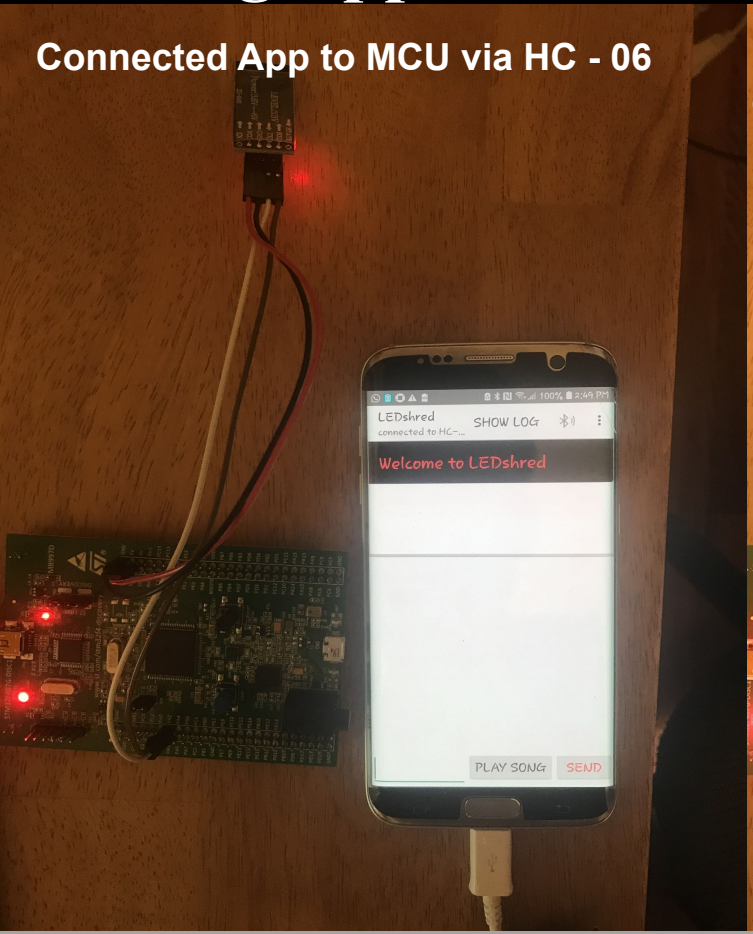
- Developed in android studio, app connects with the MCU through the module
- Commands can be sent to and from the STM32 and Android App
-

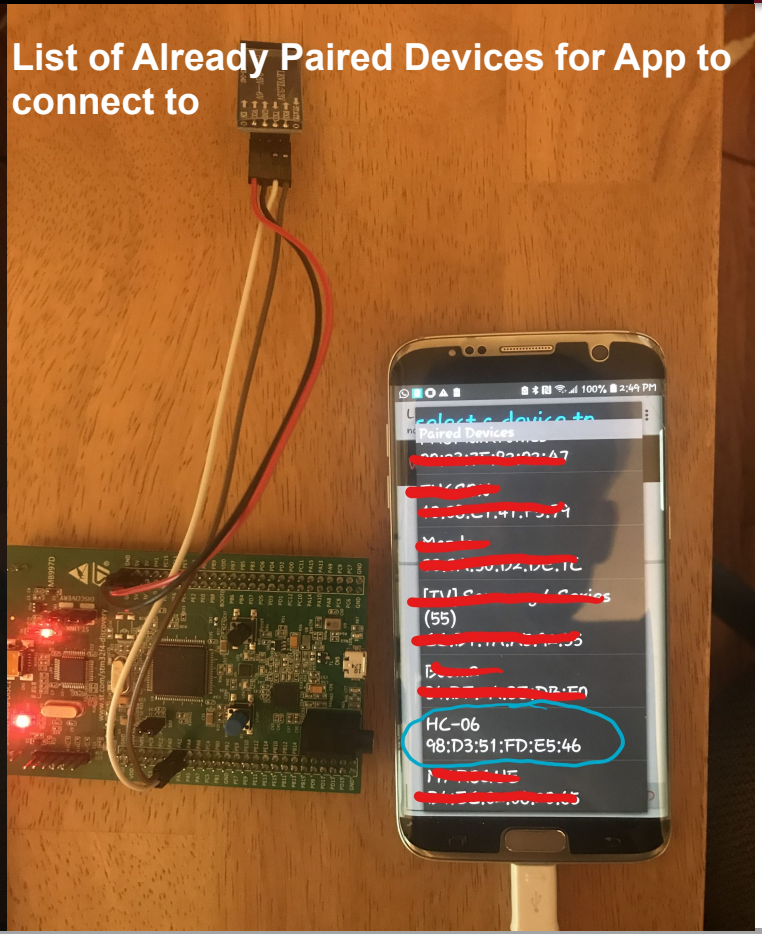
- *Demoed on a real Android devices (Galaxy S7); illustrates the working connection between device/ MCU*



Showing App Connection - Demo 1

UMassAmherst

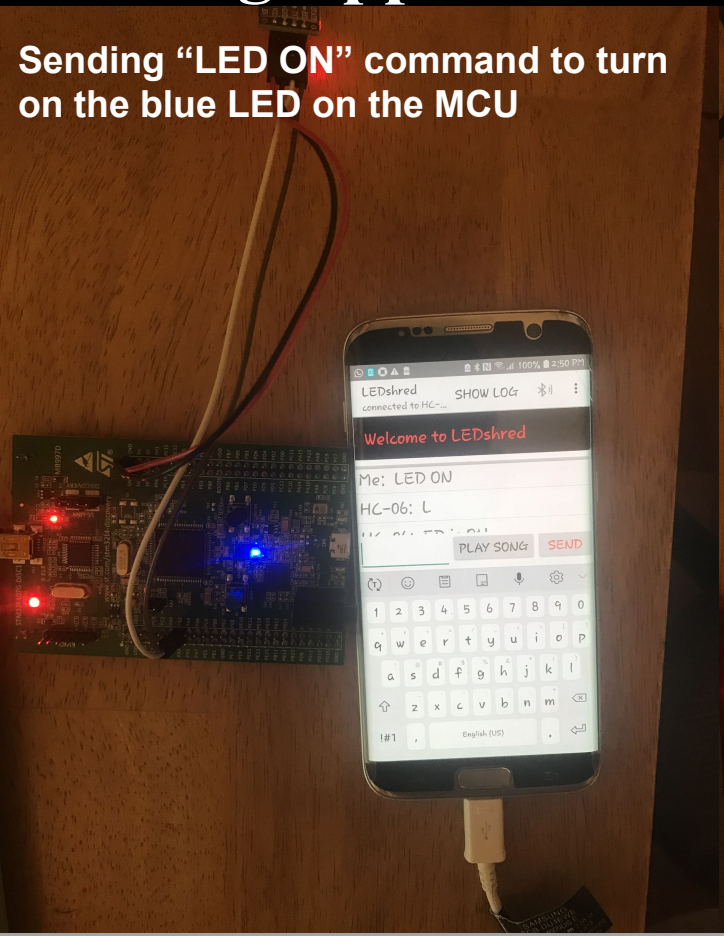


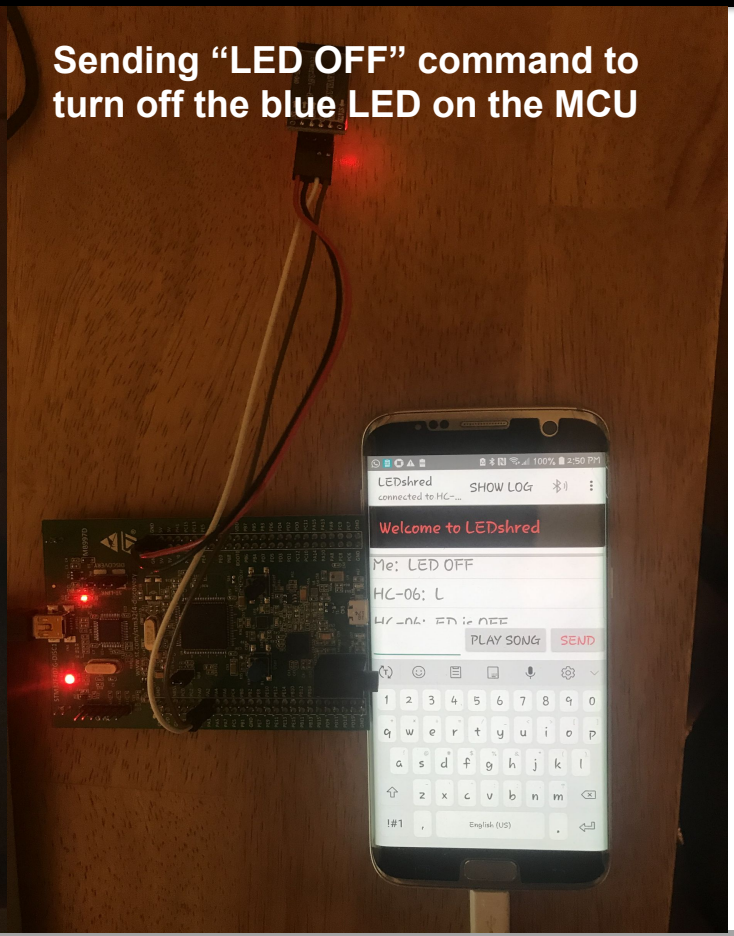


- *Kivan Daruwalla*

- A drop down on App showing bluetooth devices to connect to

Showing App Connection - Demo 1





- *Kivan Daruwalla*

- A button that toggles blue LED on MCU On/Off (Send)

Demo 1 - App Connectivity

- *Kivan Daruwalla*

Android Application - App Functionality

Created Functionality on the app itself:

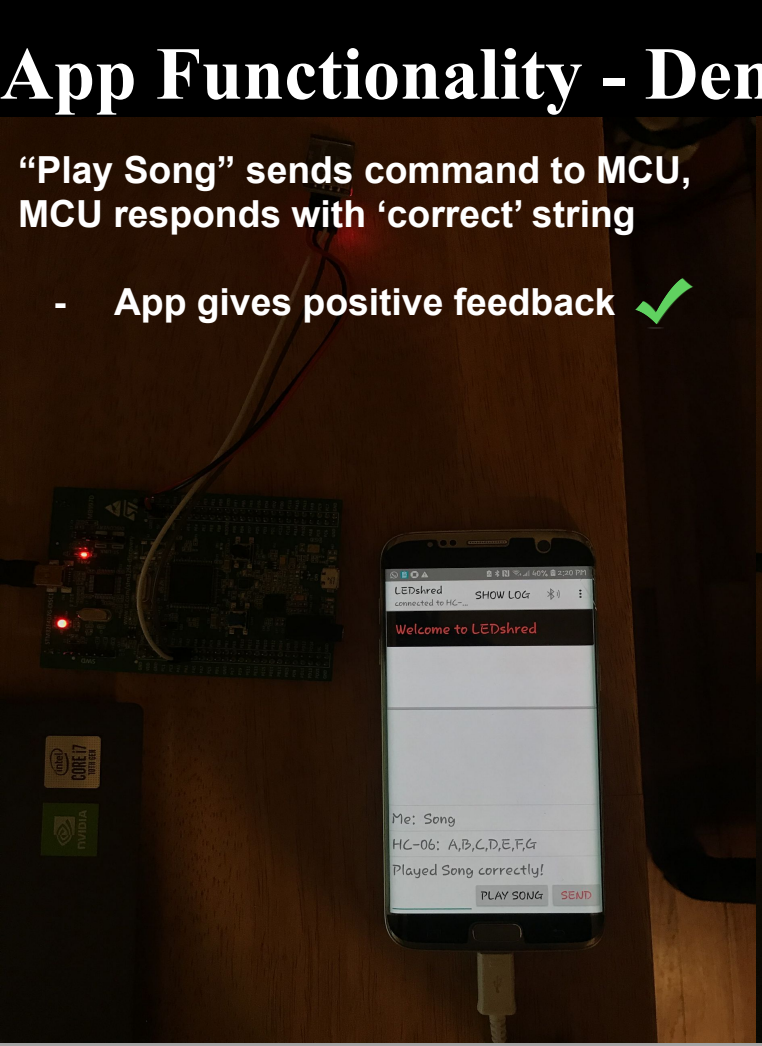
- A button that sends a string of arbitrary letters to the MCU (Play Song)
  - Followed by response from a buffer on the MCU with a sting of arbitrary letters, sometimes identical to what the MCU received, and sometimes different

- On App: Compare received string from MCU with that on app and create spoof "results"

App Functionality - Demo 2

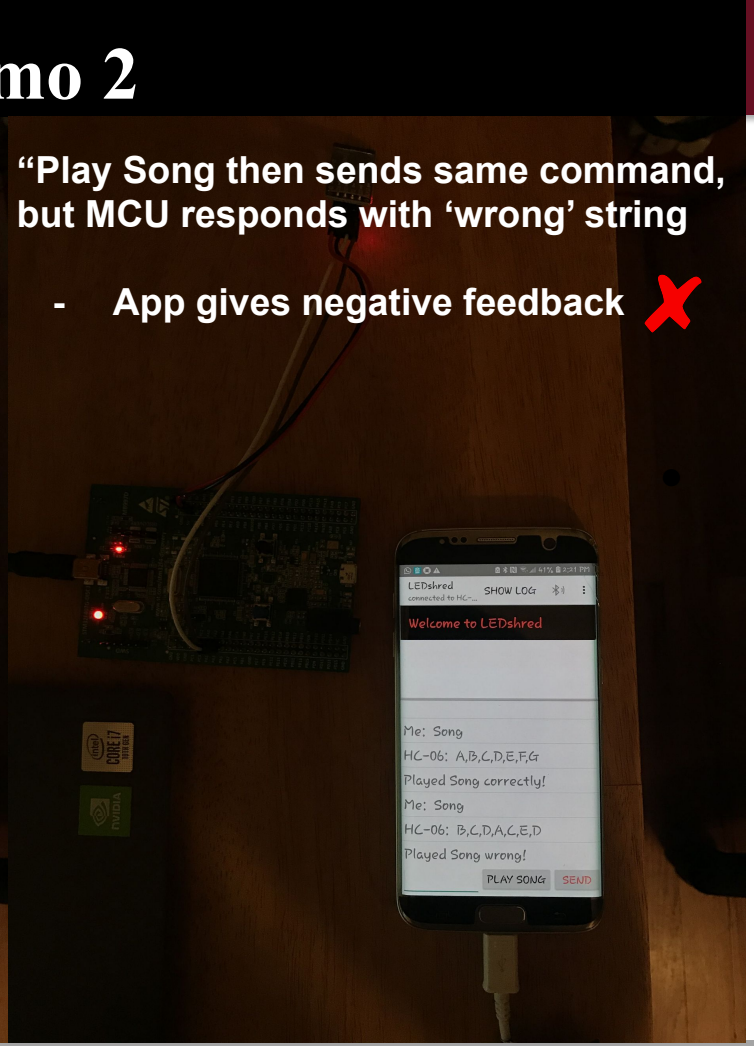
“Play Song” sends command to MCU, MCU responds with ‘correct’ string

- **App gives positive feedback** ✓



“Play Song then sends same command, but MCU responds with ‘wrong’ string

- **App gives negative feedback** ✗

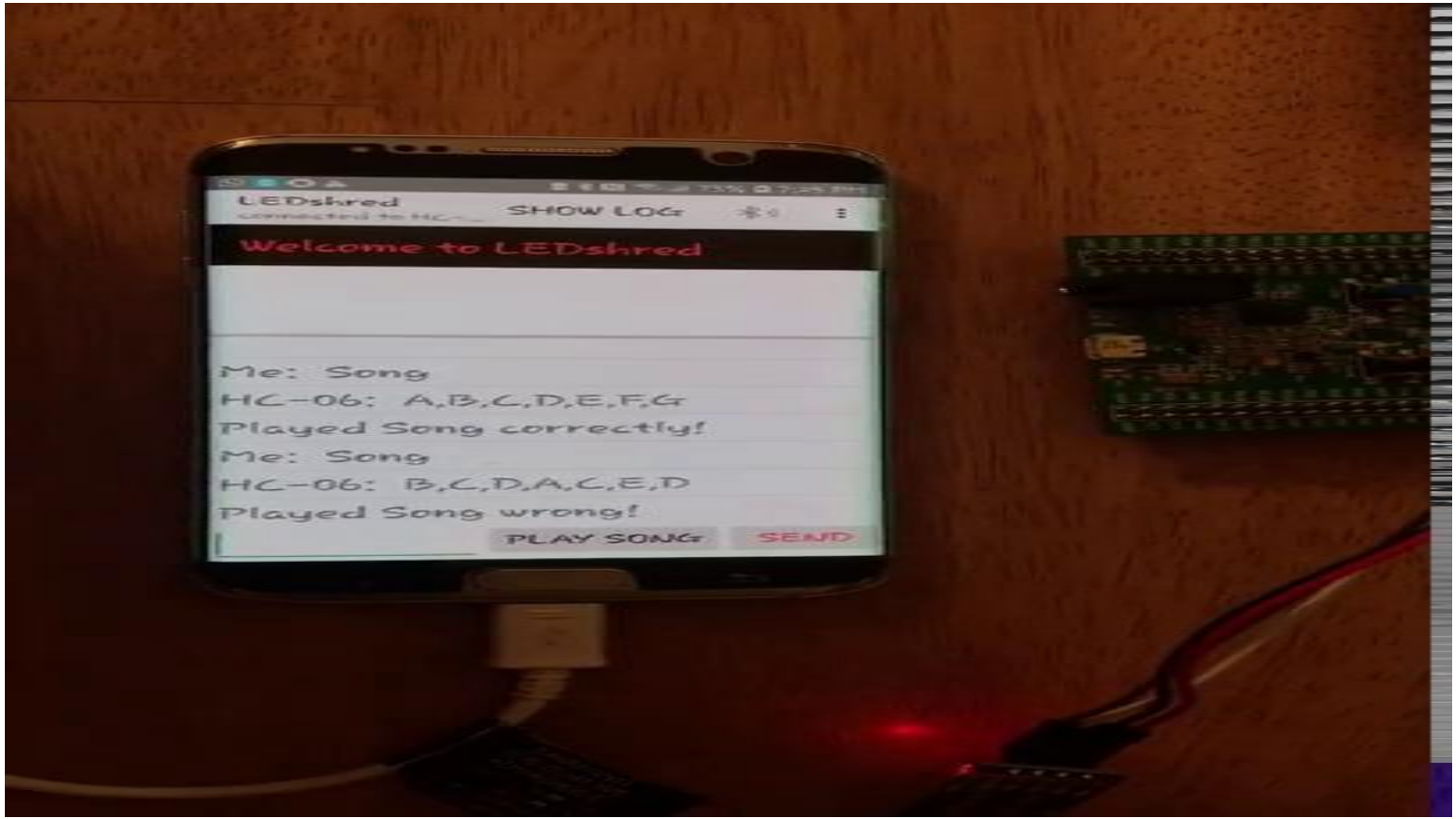


- *Kivan Daruwalla*

- “Play song” button to send sting if letters (notes) to the MCU
- MCU replies

Demo 2 - App Functionality

- *Kivan Daruwalla*



FPR Hardware Plan

- PCB with Microcontroller mounted on it
- Attachable to guitar in small casing in order to not interfere with user and play
- LEDs will be on thin, flexible PCB for each fret
- Will be using hexaphonic pickup in the future to precisely pick up the sound from the guitar

Planned implementation of LED lights on thin PCB

List of Hardware and Software

MDR List

Hardware

- STM32 MCU
- Arduino Uno
- Dotstar LEDs (5x5 mm)
- Microphone breakout board
- Bluetooth module

Software

- STM32Cube IDE
- STM32CubeMX
- Matlab
- Arduino IDE
- Android Studio

Future List

Hardware

- STM32 MCU
- Dotstar LEDs (2x2 mm)
- Hexaphonic pickups
- Bluetooth module
- Power supply

Software

- STM32Cube IDE
- STM32CubeMX
- Android Studio

Project Expenditures

Expenses so far

Item	Qty	Cost ($)
STM32F4Discovery	2	62.57
Microphone	1	1.97
Dotstar LED 5050	2	13.53
Dotstar breakout board	2	13.53
WS2812B Strip	1	20.88
Dotstar APA102-2020	1	17.45
		Total: $129.93

Estimated future Expenses

Item	Qty	Cost ($)
PCB	1	30
LED APA102 2020	1	5.95
PCB Power supply	1	12.56
Ubertar Hexaphonic Pickups	1	(Subsidized by dept)
Acoustic guitar	2	80
		Total projected cost: $128.51

Project Management Gantt Chart for Spring 2021

Task	Engineers	Feb 1	Feb 8	Feb 15	Feb 22	Mar 1	Mar 8	Mar 15	Mar 22	Mar 29	Apr 5	Apr 12	Apr 19	Apr 26
Transfer code from Arduino to STM32 and order LED lights	Jordy													
Order Hexaphonic Microphone and guitar	Isaiah													
Build Matlab Script for hexaphonic pickups	Jacob													
Generate layout of application with available functionality	Kivan Daruwala													
Design and implement Hexaphonic pickups	Isaiah													
Start PCB design in Altium for MCU and hexaphonic pickups	Isaiah													
PCB design in Altium for LED lights	Jordy													
Transfer MATLAB script into C code and test on STM32	Jacob													
Order PCB and test	Isaiah													
Make values returned from MCU usable in App	Kivan Daruwala													
Testing on LED PCB and configuring song notes with app and MCU	Jordy													
Finalizing algorithm to identify notes with pickups														
CDR Week	All													
Make functionality on application work with data from MCU	Kivan Daruwala													
Continue to improve algorithm speed and accuracy	Jacob													
Improve quality of notes sent to the microcontroller	Isaiah													
Developing final model of application	Kivan Daruwala													
Finalize algorithm	Jacob													
Develop songs and different modes for LED lights	Jordy													
Optimize LED configuration and finalize communication with App and MCU	Jordy													
Testing all subsystems and making sure they are working and able to communicate with one another	All													
FPR Week	All													

UMassAmherst
The Commonwealth's Flagship Campus

Thank you

Questions?