Electronic Piano Teacher

Joe Breda, Cassius Peter, Matt Caswell, Aleksa Deric



Department of Electrical and Computer Engineering - Joe, Cassius, Matt, Aleksa

The Team



Joe Breda - CSE Note Recognition Software



Matt Caswell - CSE

Microprocessor programming and communication



Cassius Peter - EE LED Strip and hardware



Aleksa Deric - CSE

Android application programming

Problem Statement

Learning piano is a arduous and multifaceted task that is commonly supervised and reinforced by a piano tutor. These piano lessons can be expensive and difficult to schedule.

We propose a system that can autonomously supervise and reinforce the piano learning process by:

- Displaying notes to be played
- Administering haptic signals to the user on which finger to play with
- Perform error detection recognizing if the note being played is the correct note

Project Specifications

- **Instruction:** System should show the user sheet music, what note on the piano that corresponds to, which fingers to use to hit those notes, and detect errors
- **Synchronization:** All signals to the user for each note should arrive within 100ms of each other
- Accuracy of note detection: Correctly detect single note melodies as often as it can, with priority on most commonly used notes in middle octaves
- Latency of note detection: Note detection should provide live feedback. No more than 1 second of latency
- Usability:
 - **Non-obstructive:** System should not obstruct normal finger mobility
 - Battery life: up to 2 hours
 - Range: within 10 feet
 - **Retrofit-ability:** Any standard width (48 inch) 88 key keyboard
 - **Cost:** "Mass production" system must cost less than \$250 ~ 4-5 piano lessons

Block Diagram



MDR Deliverables

- Get all interfaces working
 - Defined and implemented communication syntax between phone and control board, and control board and glove boards
- Demonstration of microphone listening to piano, communicating to LED strip and lighting up the notes you played
 - All individual components are working, the full loop is not implemented.
- Collected training dataset of MIDI files with labels for musical key and possibly finger placement for finger to note model
 - Created a proof of concept MIDI file with labeled finger placement

Android application

-Can communicate with other bluetooth devices -Text converted to a serialized stream of bytes -Passed to the controller board

-HC-09 bluetooth chip

- -Actual throughput advertised: 6 KB/s*
- -Meets latency specification
- -Example format:

turns on 4 LEDs, vibrates 4 fingers, turns off 4 LEDs

y11,13,15,22,;f1,6,9,0,;n12,14,16,23,;

-Translates to about 38 Bytes/sec

-Tested with 52 bytes without issues

*https://engineering.purdue.edu/477grp19/Files/refs/SH-HC-08%20datasheet.pdf

Android Tablet

Sound from Piano

MIDI File

App Displaying Sheet Music, Sending info to Board, and classifying notes

Microphone

Bluetooth

Transmitter

Note information

Controller Board

Communicates over Bluetooth to Tablet using AT-09 Bluetooth chip

Communicates with Glove Boards using 433MHz transmitter chip

ATmega328P Microcontroller on an Arduino Platform

Both connections are seen as serial ASCII connections by the microcontroller





Glove Board

433MHz Receiver 5x 1cm Vibration Motors

ATmega328P Microcontroller on an Arduino Platform

Range: > 30 feet Power Draw: 100mW idle, 425mW with all 5 motors running Need a ~1 Watt hour battery (~200mAh@5V)

Both gloves receive the same signal. Left glove has fingers 0-4, right glove has 5-9







Board Communication Syntax

Tablet -> Controller Board

- Commands separated by ";"s
- Command consists of control character followed by comma separated list of numbers (trailing comma is required)
- Control Characters:
 - y: turn following keys ON
 - n: turn following keys OFF
 - f: vibrate the following fingers

Example: "y1,3,22,;f1,3,5,;n6,;"

Turn on keys 1, 3, and 22. Vibrate fingers 1, 3, and 5, and turn off key 6.

Controller Board -> Glove Board

 Send up to 10 digits (0-9) to indicate that the corresponding finger should be vibrated

Example: "46"

Vibrate fingers 4 and 6.

- Uses W2812s serial LEDs, 144/M
- Will be attached to a thin rigid bar, most likely wood, and placed over the piano keys
- Uses simple function to map LED to piano key



Department of Electrical and Computer Engineering - Joe, Cassius, Matt, Aleksa

Note Recognition Script

- uses PyAudio to stream audio input
- Finds 3 max peaks in FFT under the assumption that 3 largest peaks will be f1, f2, and f3.
- Estimates fundamental if and only if 3 peaks form harmonic series f1, f2, and f3 or any permutation.



Department of Electrical and Computer Engineering - Joe, Cassius, Matt, Aleksa

Fourier Transform of Piano Note

- "timbre" of piano has strong harmonics
- Upper harmonics are often lower magnitude than lower harmonics (but not always)



Finger Model

- Manually encoded fingering into MIDI file
- Velocity field in each note was previously unused
- Set velocity of MIDI note to 0-9 to indicate which finger to use



CDR

- Note Classification Running on Android
- Baseline Finger Placement Algorithm
- Glove PCB boards mounted on a non-final arm

sleeve and working

 LED strip mounted on rigid backing and programmed to align with piano keys

Timeline



Demo - Board Communication

- Phone app sending commands to Control board over bluetooth using defined syntax
- Control board executes commands and communicates over 433MHz transmitter to glove board
- Glove board executes commands

Demo - Note Recognition

How it works:

- Calculates first harmonics ~10 times per second
- if any 3 calculations return the same note in a row the prediction is confirmed
- Steams live audio signals
- does not output when no note is playing
- Occasionally PyAudio buffer overflows
- Does not work for lowest 15 notes

THANK YOU