Hands-Free DJ

Team 12
Yifei Teng, Jie Du, Ningkai Wu

Problem Statement / Objective

- Offer an alternative to classical DJ Station which is expensive and unportable
- Provide a small and easy-to-use home party music player

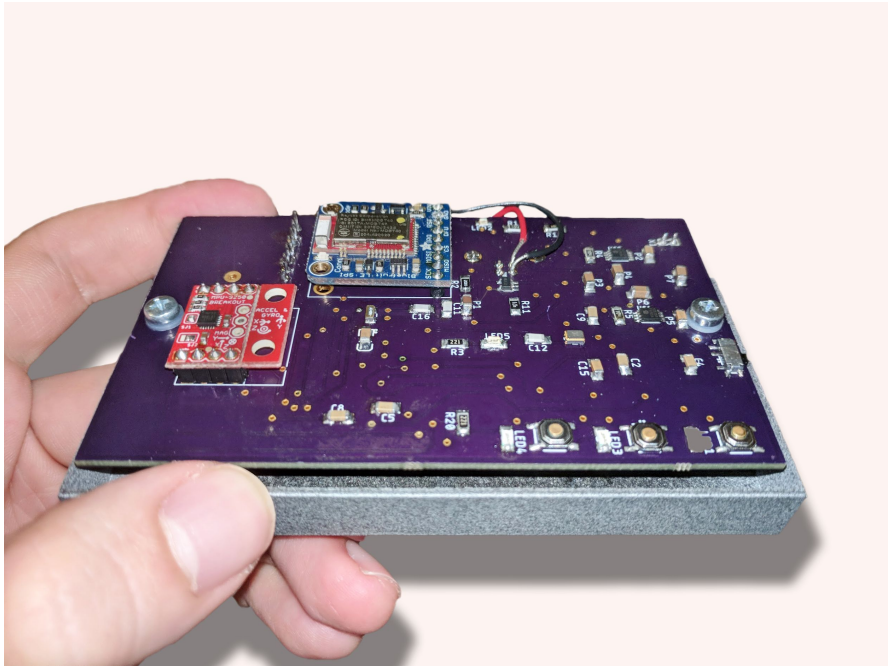
$1,999.00
AmericanMusical.com

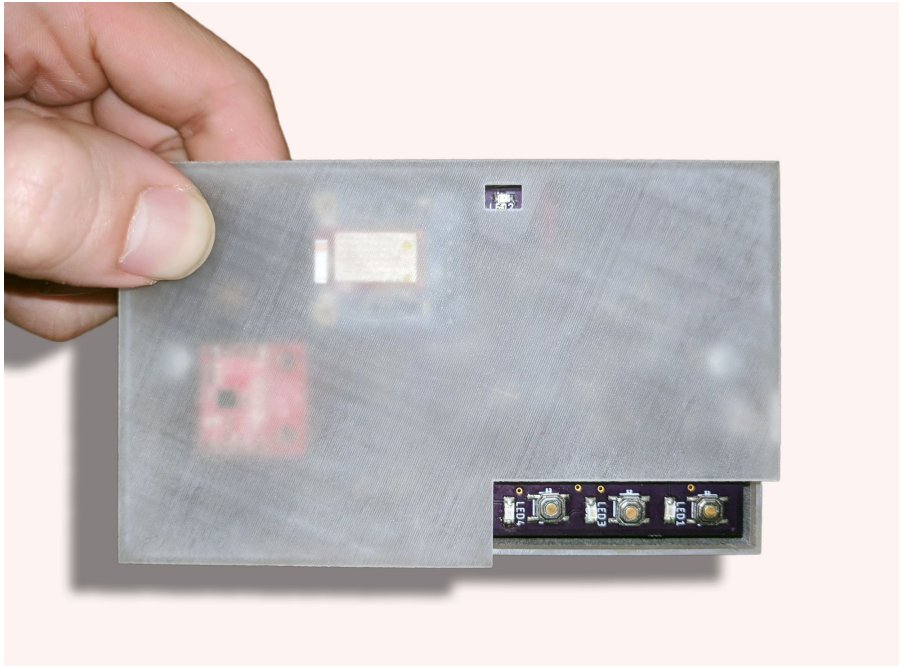
$1,199.00
AmericanMusical.com

Introduction

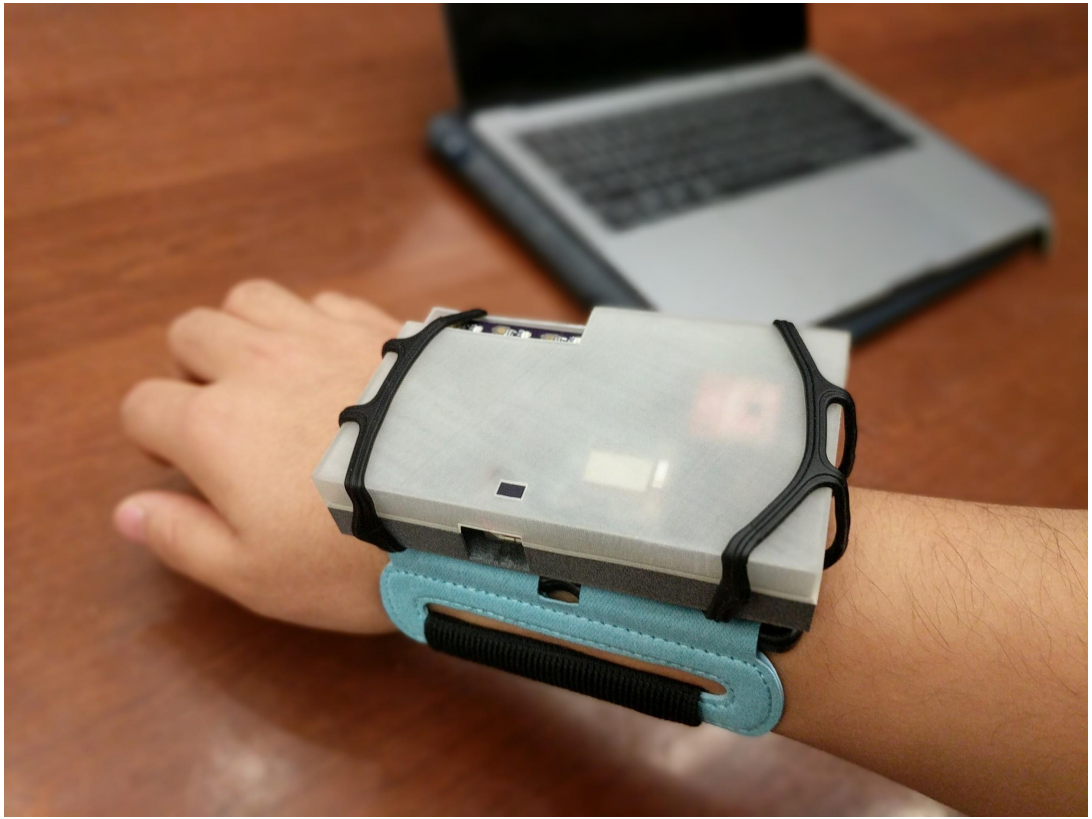
- Remote-controlled DJ package
- By moving the wrist, the DJ will be able to remotely change the song
- Can also apply various audio effects via a DSP pipeline
- Wrist-wrapped gesture sensor
- Talking to an iPhone app which controls the music

What does it look like?





What does it look like?



What does it look like?

Gesture Logic

All modes: Turning of wrist maps to texture change. (Like turning the texture knob on a guitar amp). Slicing your hand horizontally outwards through air moves to the next song. Slicing inwards goes to the previous song.

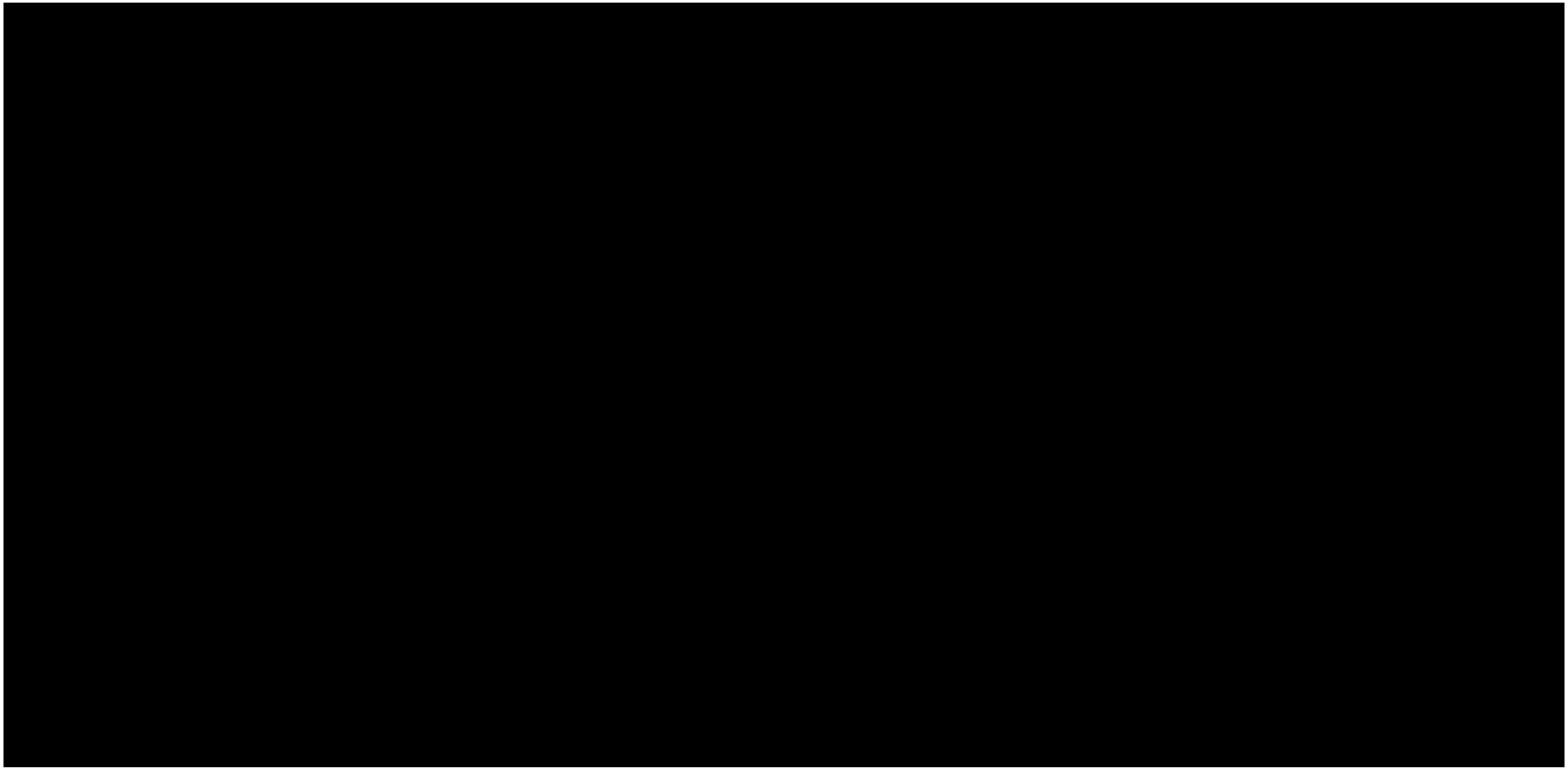
Mode 1: Height maps to pitch. Up/down rotation of palm maps to reverb.

Mode 2: Sudden upwards rotation of palm maps to activating looping, sudden downwards rotation maps to stopping the looping.

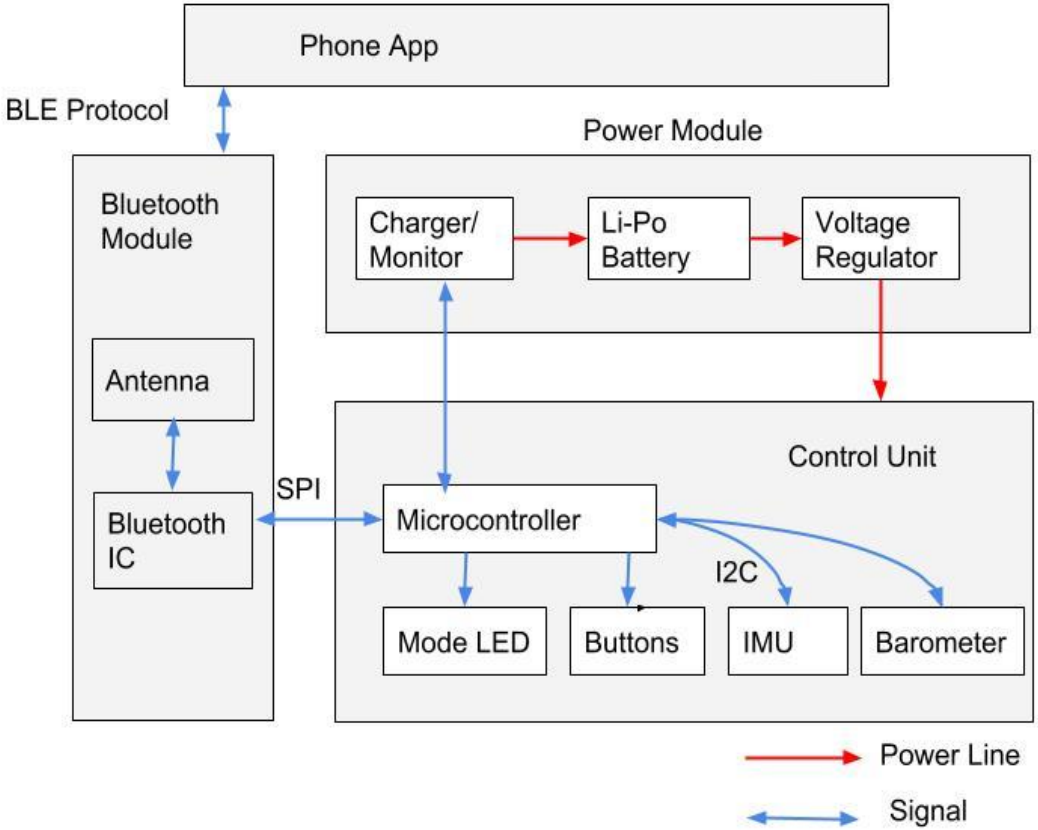
Mode 3: Height maps to wah-wah effect.

We'll use buttons for switching between modes, and for on/off.

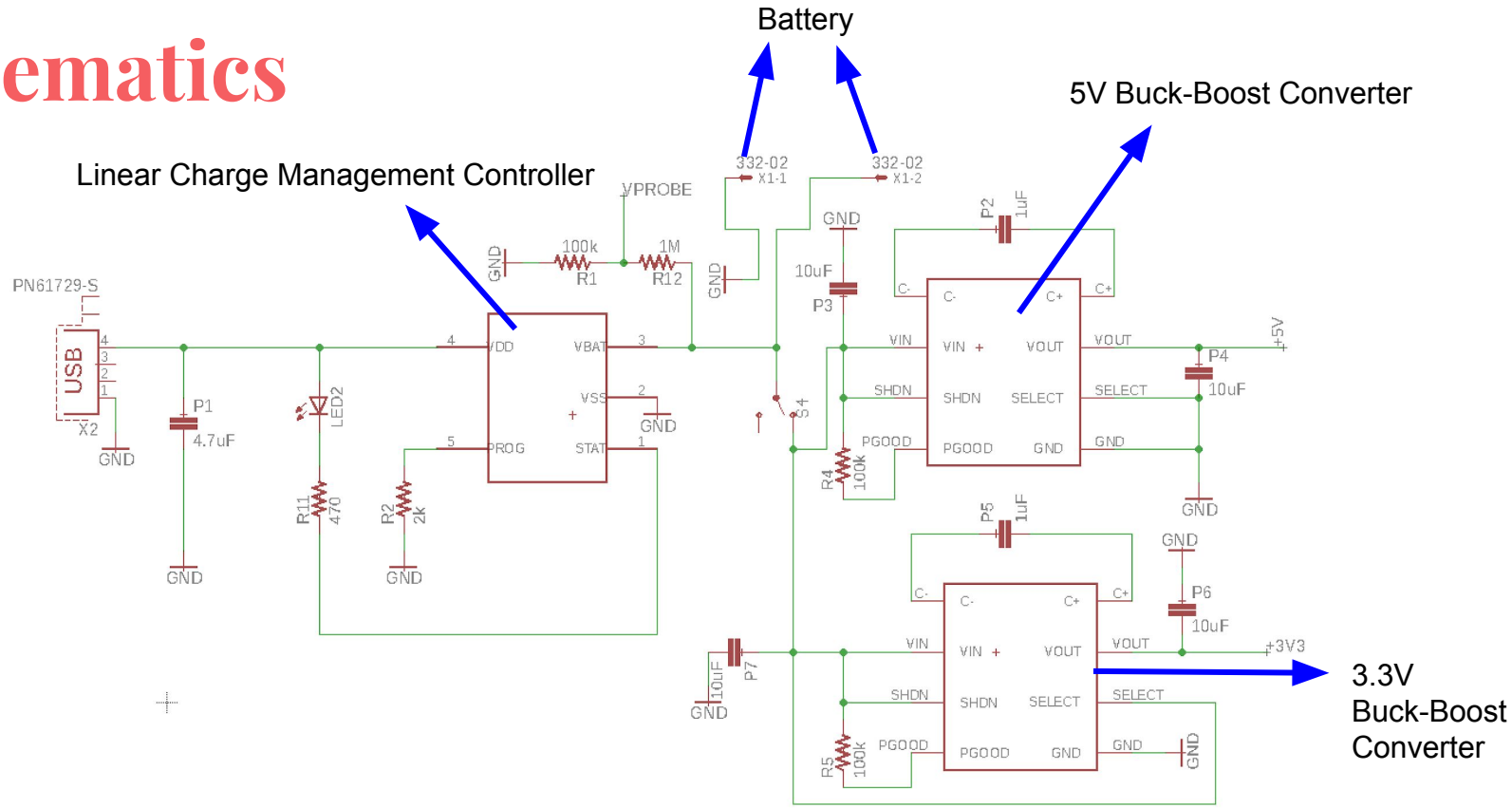
Project Demo



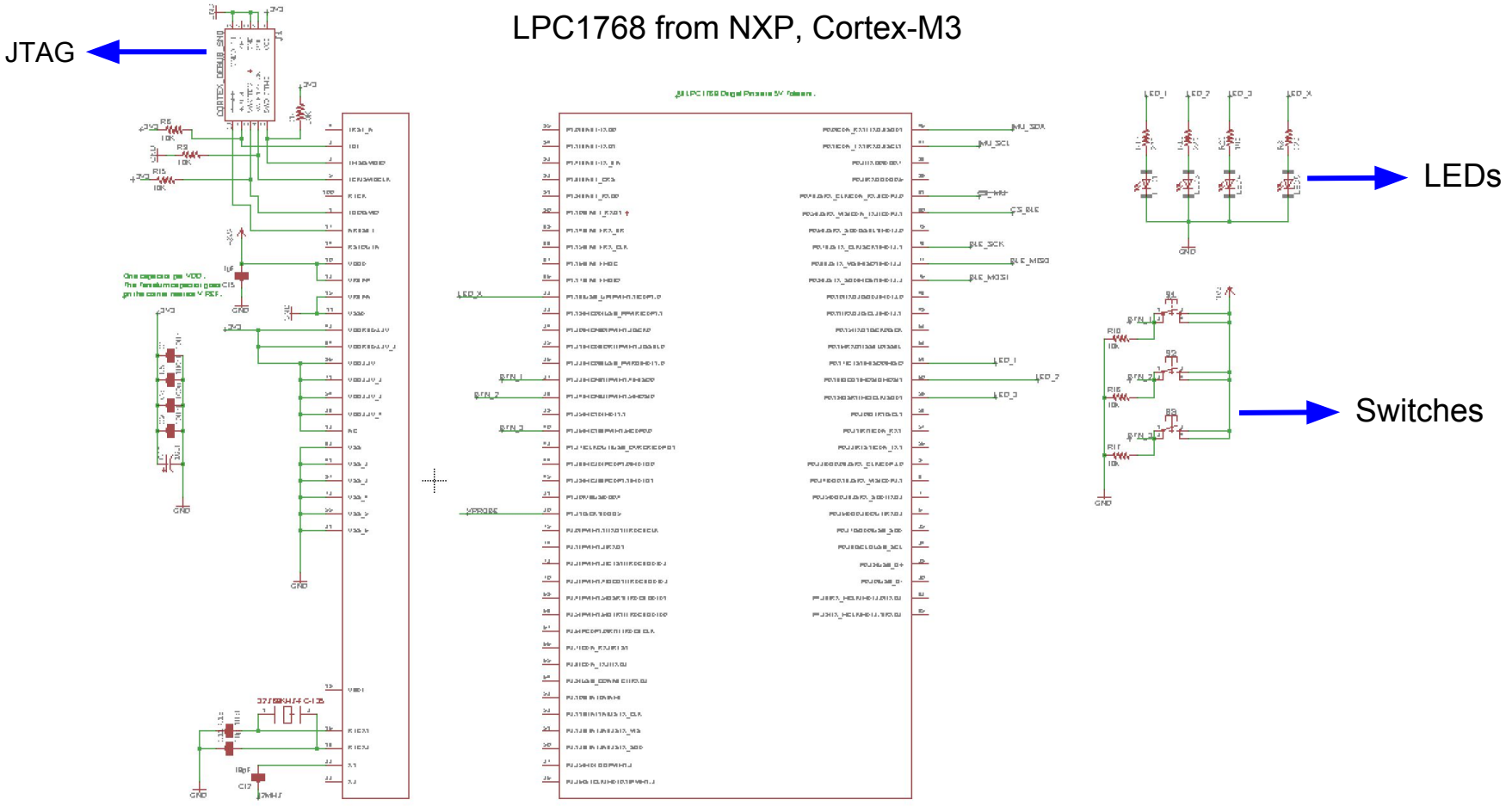
Block Diagram



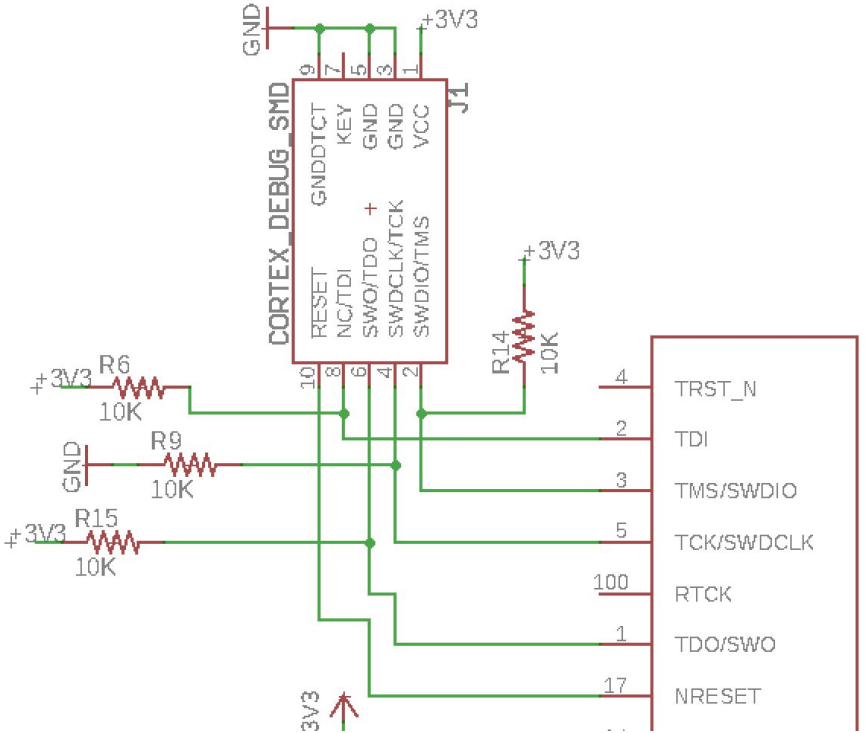
Schematics



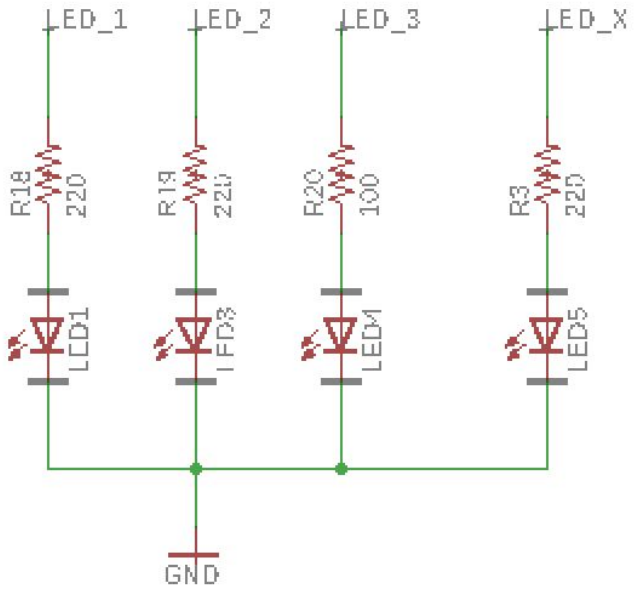
Power Module


JTAG
LPC1768 from NXP, Cortex-M3
LEDs
Switches
Microcontroller Module

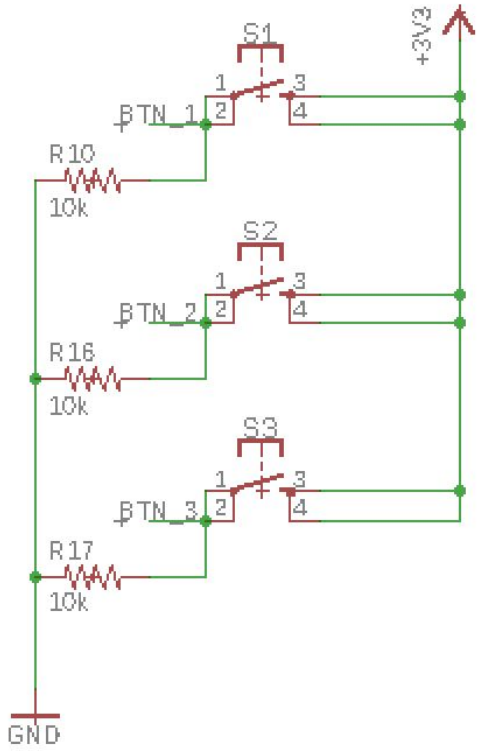
JTAG


GND
+3V3
CORTEX_DEBUG_SMD
J1
GNDDTCT
KEY
GND
GND
VCC
RESET
NC/TDI
SWO/TDO
SWDCLK/TCK
SWDIO/TMS
R14
10K
R6
10K
R9
10K
R15
10K
TRST_N
TDI
TMS/SWDIO
TCK/SWDCLK
RTCK
TDO/SWO
NRESET
3V3

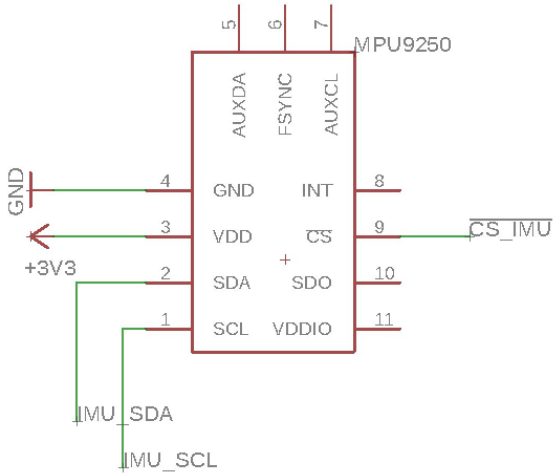
LEDs



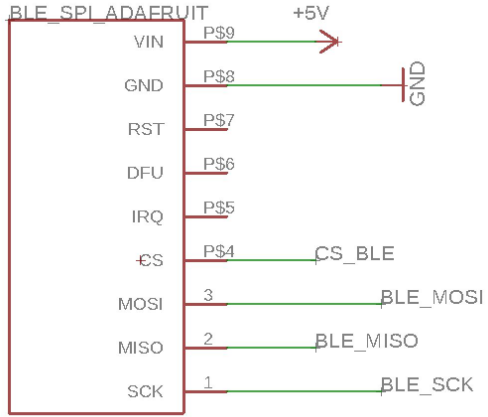
Switches



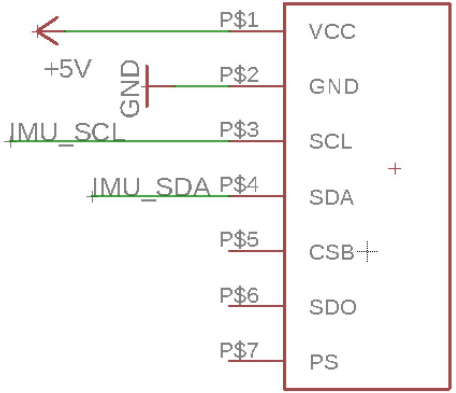
Peripherals Schematics



Inertial Measurement Unit

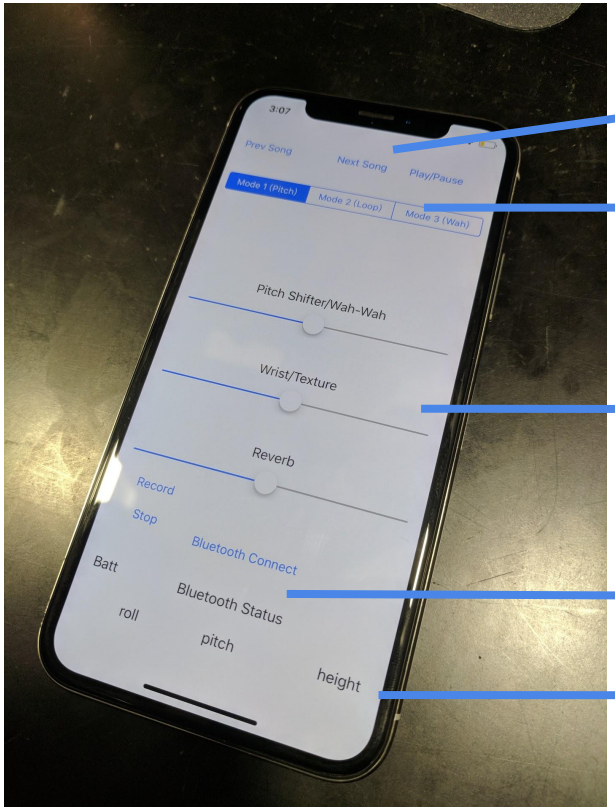


Bluetooth Unit



Barometer Unit

iPhone App



Previous Song, Next Song, Play/Pause

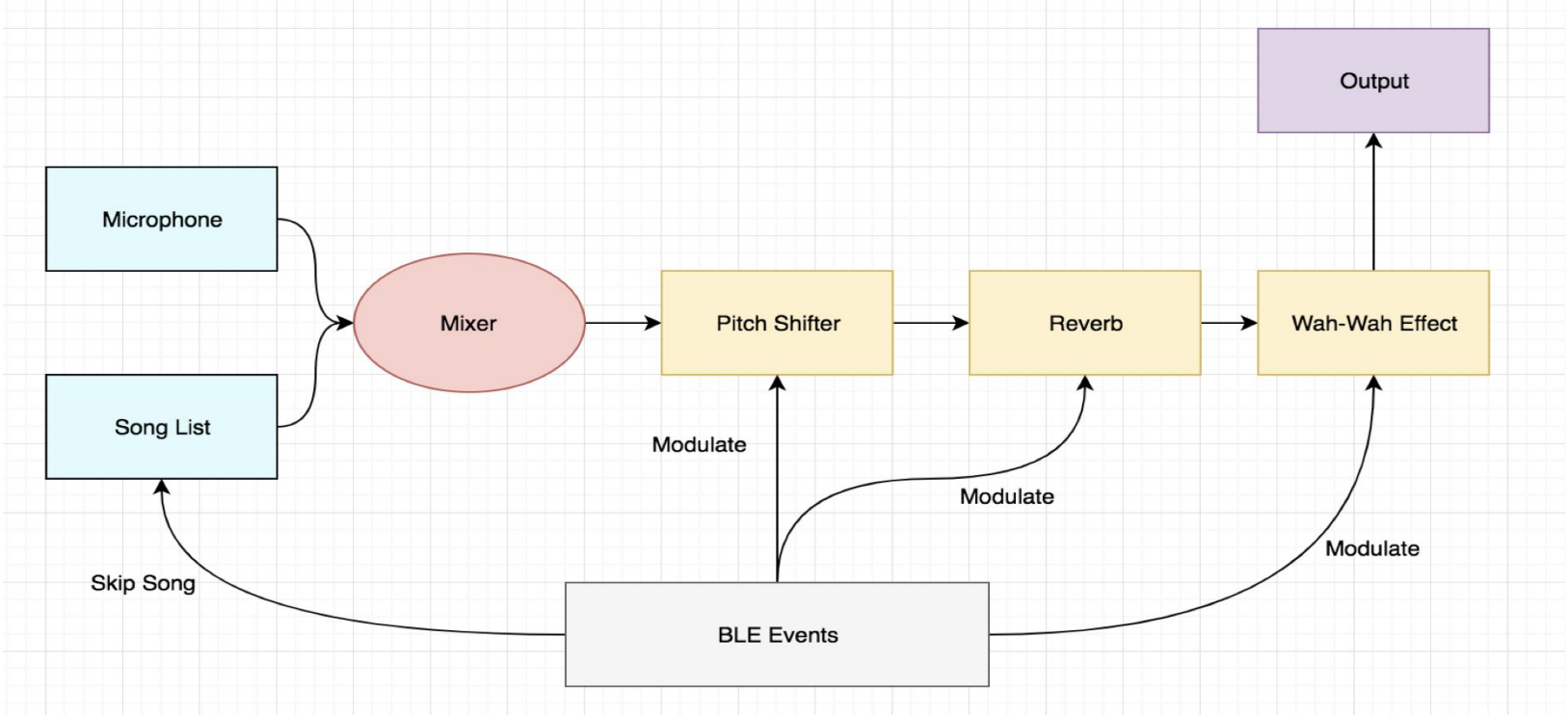
Mode Switch

Different Sound Effects

Bluetooth Connection Button and Bluetooth Status

Roll, Pitch, Height Value From IMU and Barometer

Audio Processing Architecture



Gesture Detection Implementation

Yaw

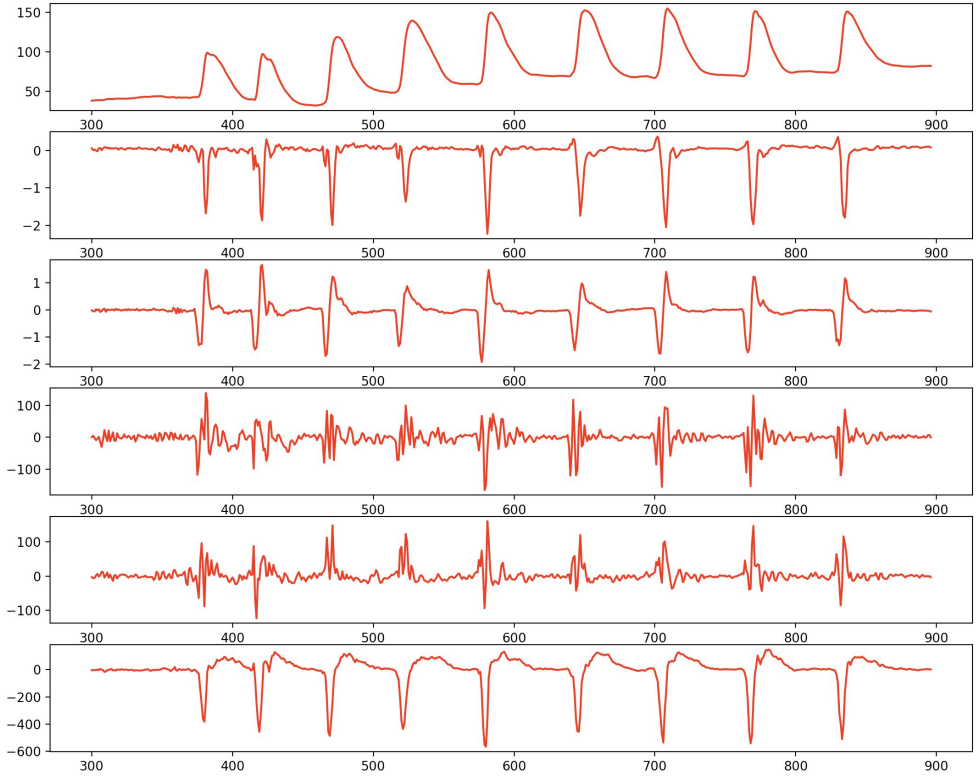
Ax

Ay

Gx

Gy

Gz



Data logging for “next song” gesture (horizontal swipe)

X-axis is time-steps in units of 20 milliseconds

Implementation for “Next Song”

```
139            if (yawDiff ≥ 60) {
140                // check gyroscope
141                bool gzNegative = true;
142                bool pitchOk = true, rollOk = true;
143                int gzLessThan200 = 0;
144                for (int i = 1; i < 9; i++) {
145                    detection_t hist = detection_queue[(detection_queue_ptr - i + 10) % 10];
146                    if (hist.gz ≤ -200) gzLessThan200 += 1;
147                    if (hist.gz ≥ 20) gzNegative = false;
148                    if (hist.pitch > 30 || hist.pitch < -30) pitchOk = false;
149                    if (hist.roll > 30 || hist.roll < -30) rollOk = false;
150                }
151                if (gzNegative && gzLessThan200 ≥ 2 && pitchOk && rollOk) {
152                    nextSong = true;
153                }
```

High-Level Requirements

- Requirement 1: The device will detect the orientation of the hand with an accuracy of +/- 10 degrees.

- Requirement 2: The device will detect “one-shot” gesture events e.g. skipping songs by horizontally throwing the wrist outwards with a success rate greater than 90% and less than two false positives in ten minutes.

- Requirement 3: The device will measure the relative change in height with an accuracy of +/- 20 centimeters.

Verification 1: Orientation Accuracy

Requirement : The orientation estimation algorithm must be stable within 10 degrees after randomly rotating for 10 seconds.

Verification Steps:

1: Set up IMU and microcontroller on a breadboard. Continuously log the orientation values to disk via UART.

2: Fix the breadboard on hand and after stabilizing, randomly rotate hand in Roll, Pitch, Yaw for any degrees for 10 seconds.

3: Rotate hand to the original stabilized position.

4: Compare the difference in Roll, Pitch, Yaw values by inspecting log and if all of them are within 10 degrees then the requirement is met.

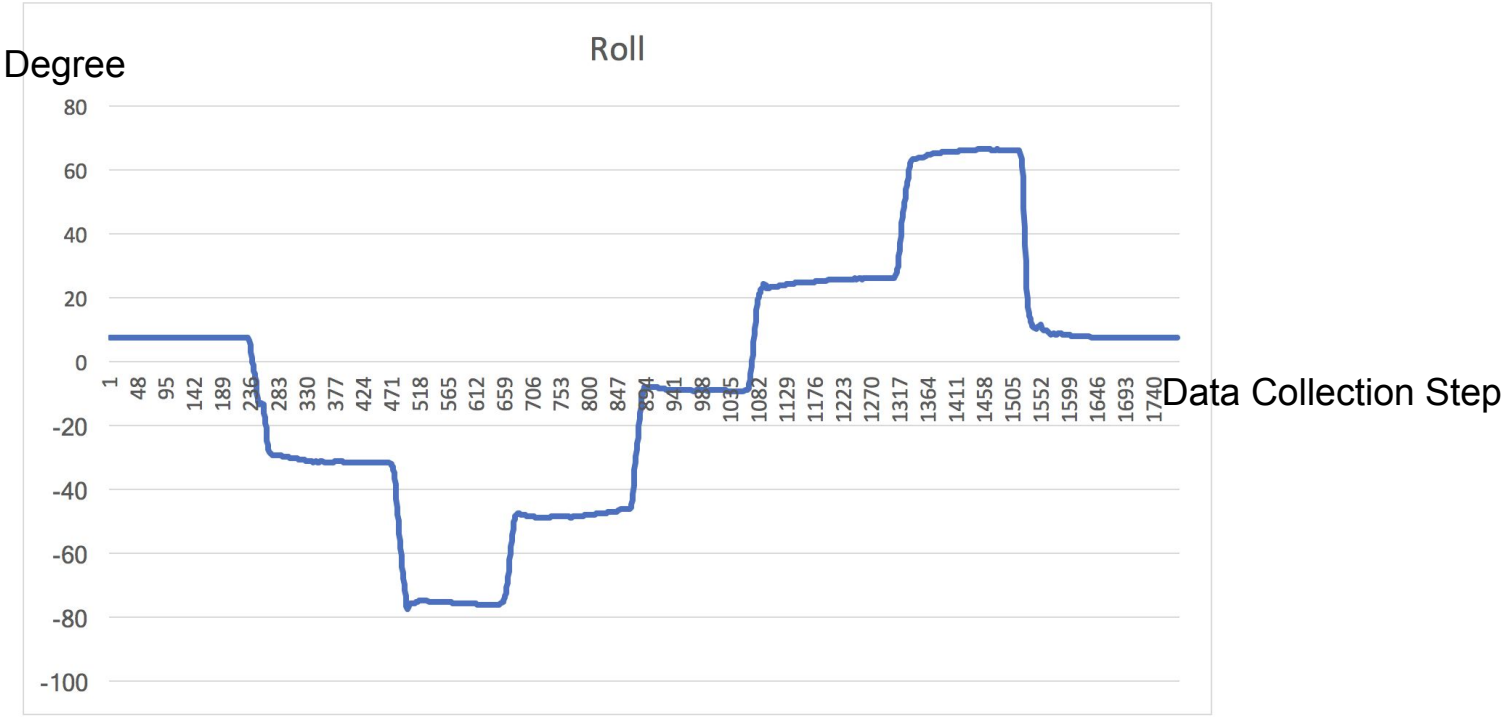


Fig 8. Roll against time-steps as measured from holding the breadboard at different angles

Verification 2: Gesture

Requirement : The device will detect “one-shot” gesture events e.g. skipping songs by horizontally throwing the wrist outwards with a success rate greater than 90% and less than two false positives in ten minutes.

Verification: Perform one-shot reliability testing by recording the number of identified gestures.

Test / Duration	No. Identified	No. Failed	No. False Positive
50 / 183 seconds	46	4	0

Success Rate = 46/50 = 92%

Verification 3: Barometer Verification

Requirement: The device will measure the relative change in height with an accuracy of +/- 20 centimeters.

Verification:

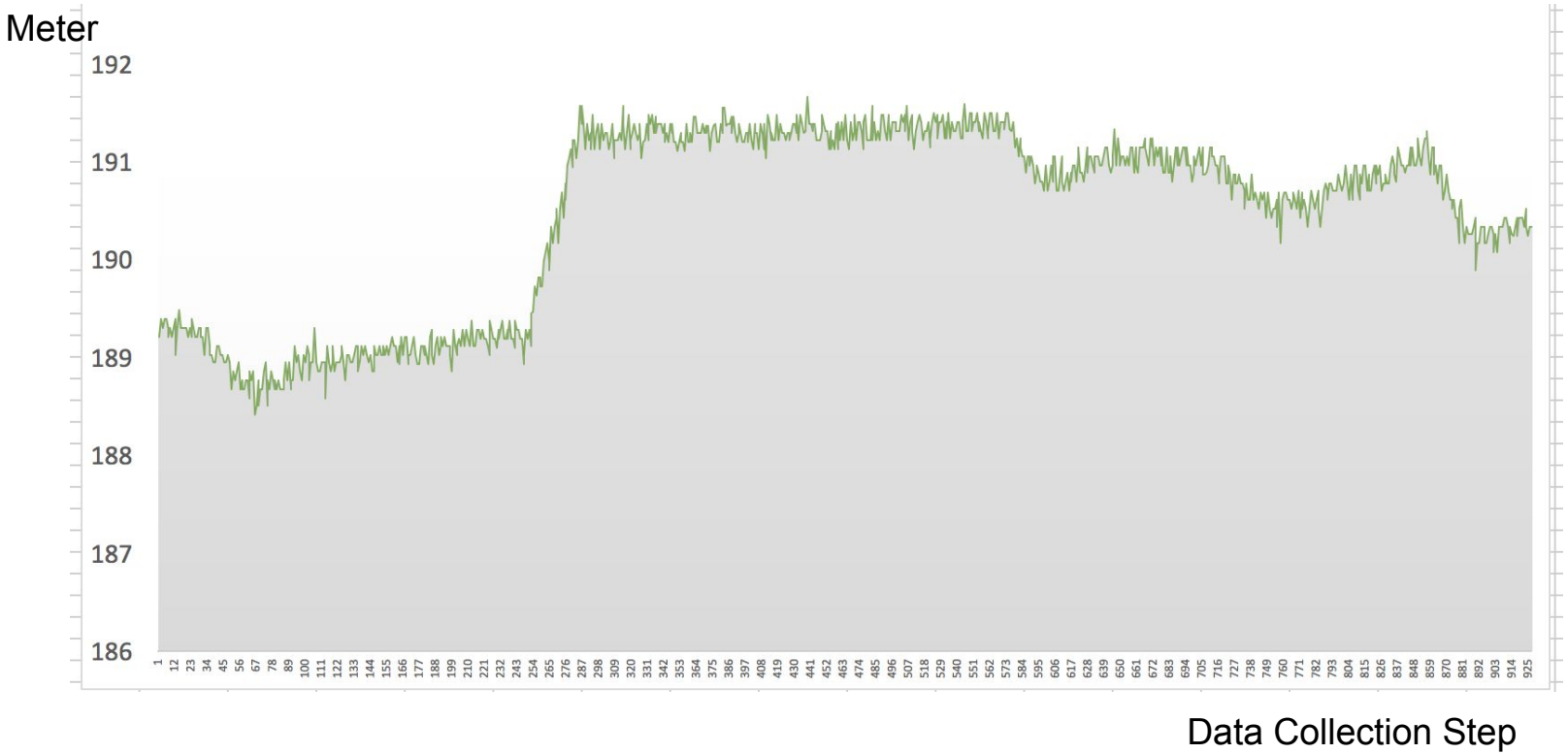
1: Set up barometer and microcontroller on a breadboard. Continuously log the height values to disk via UART.

2: Fix the breadboard on ground and stabilize for thirty seconds.

3: Move the breadboard 2.0 meters above for another thirty seconds.

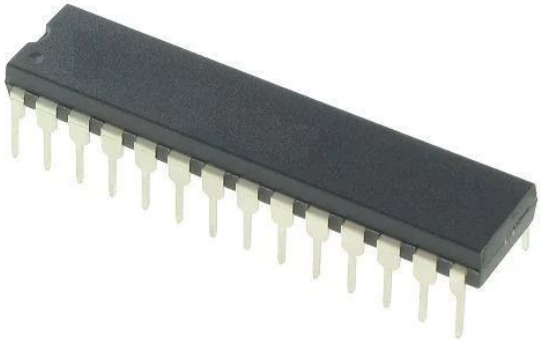
4: Compare the difference in height values by inspecting log. If the difference is within 0.2m then the requirement is met.

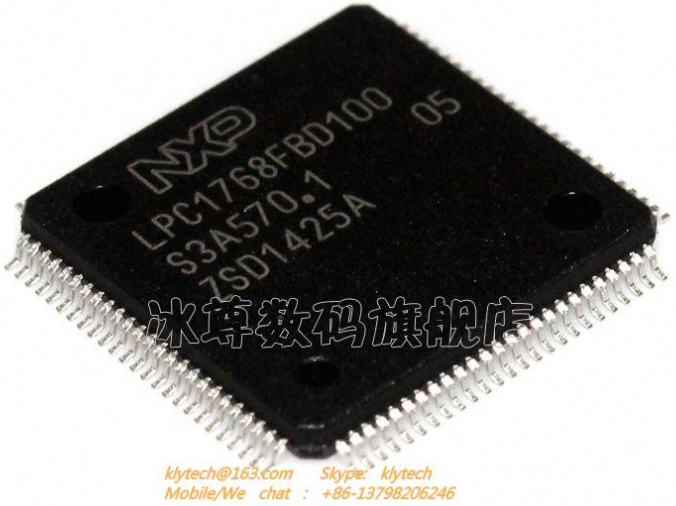
5: Repeat 1 to 4 for head to ground.


Meter
192
191
190
189
188
187
186
1 12 23 34 45 56 67 78 89 100 111 122 133 144 155 166 177 188 199 210 221 232 243 254 265 276 287 298 309 320 331 342 353 364 375 386 397 408 419 430 441 452 463 474 485 496 507 518 529 540 551 562 573 584 595 606 617 628 639 650 661 672 683 694 705 716 727 738 749 760 771 782 793 804 815 826 837 848 859 870 881 892 903 914 925
Data Collection Step

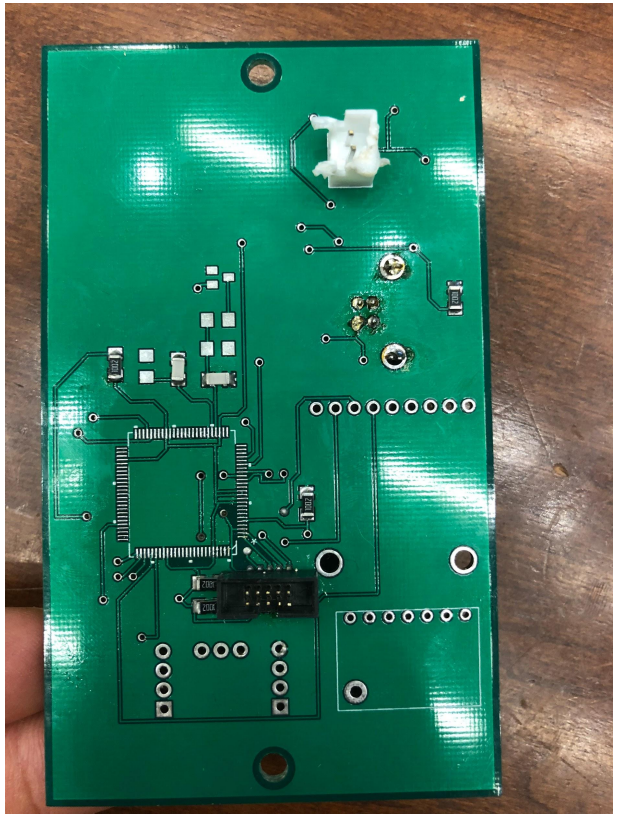
Challenges

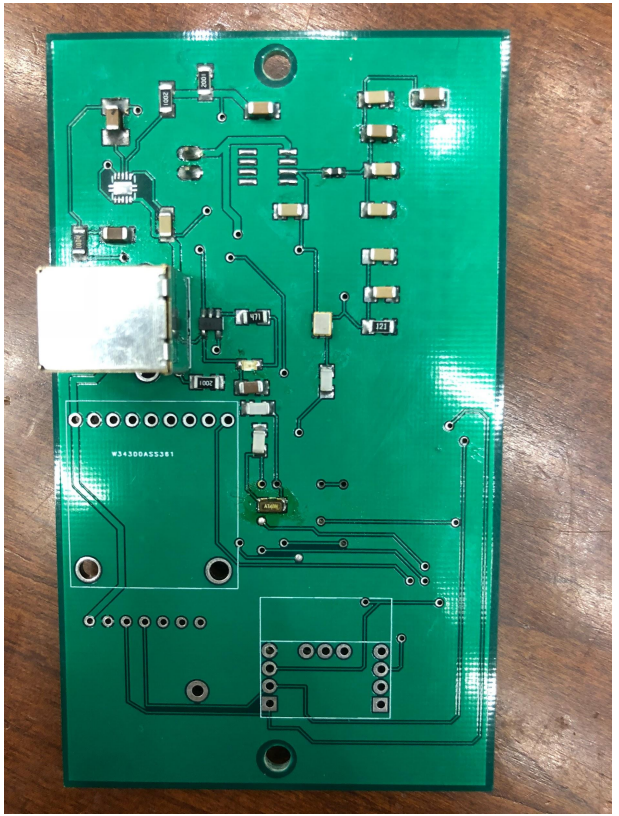
1: Soldering of Microcontroller



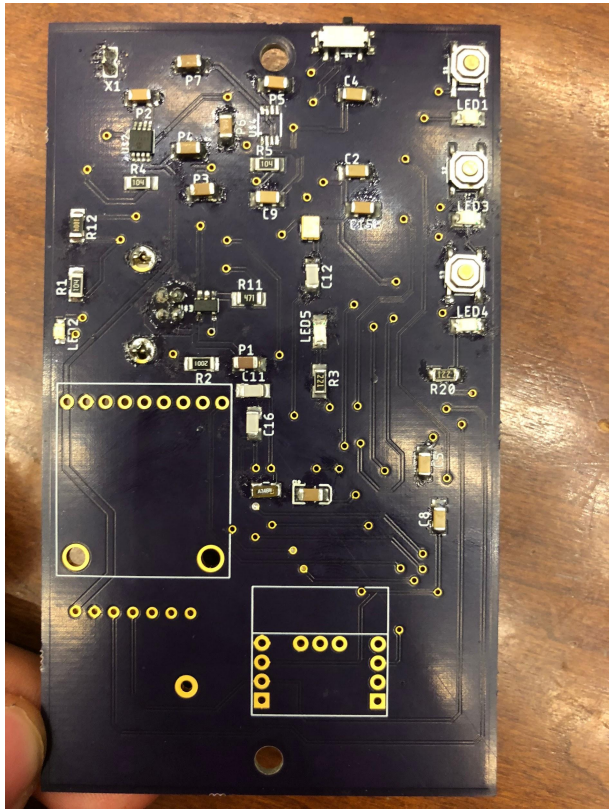


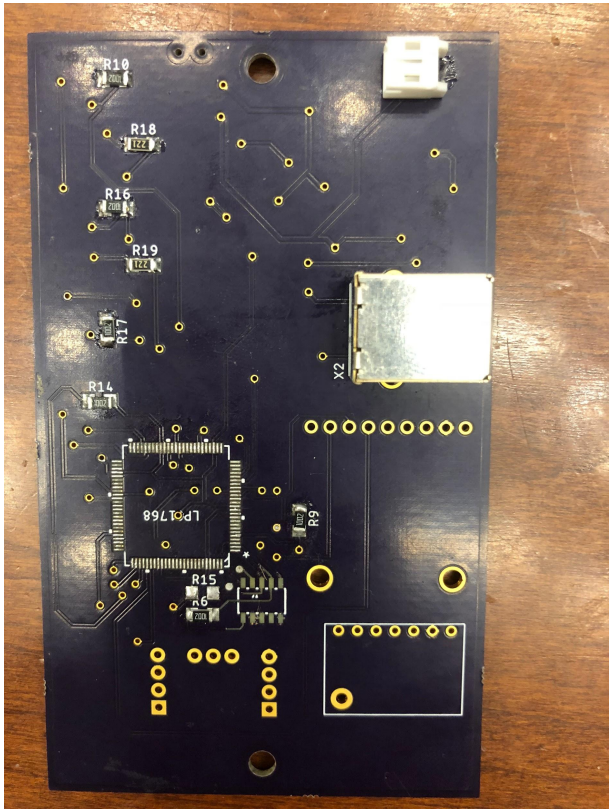
First Soldered PCB





Second Soldered PCB

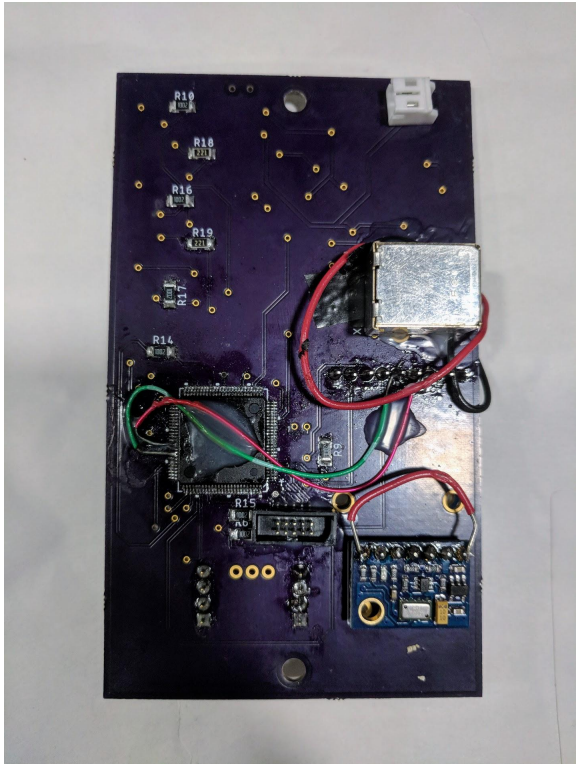




Final Soldered PCB

Front

Back



Challenges #2

- Programming Keil
- Multiple points of failures (PCB? Keil Settings? Programmer? …)
- Communication Failures
- Unknown Errors Happens All the Time

Future Work

1: Redesign PCB

2: Make the device smaller

3: Add more audio effects

Thank you!

Q & A