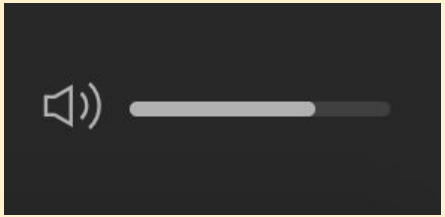
Gesture Recognition System for Music Playback

Jenny Li
Shana Mathew

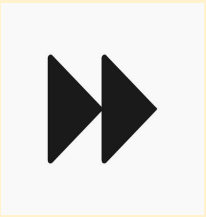
Overview

- Music is an important part of our lives. We can listen to music:
  - At home
  - On our way to classes/work
  - As we pset or work on our Vivado code
- But… the way we **control music** has largely stayed the same…



Volume scroll

Play/Pause

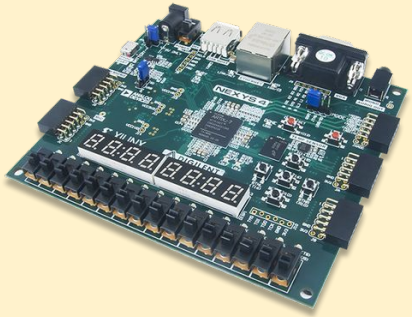


Fast forward/rewind

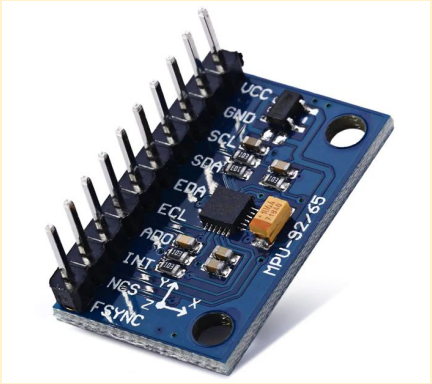
What about a new way to **CONTROL OUR MUSIC EXPERIENCE??**

Physical Parts

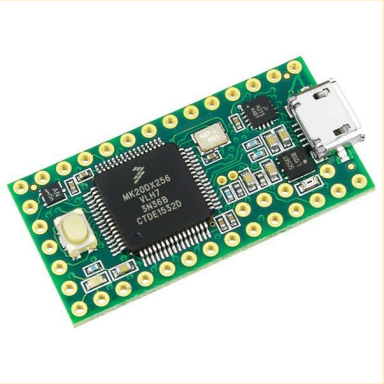
Hardware components:



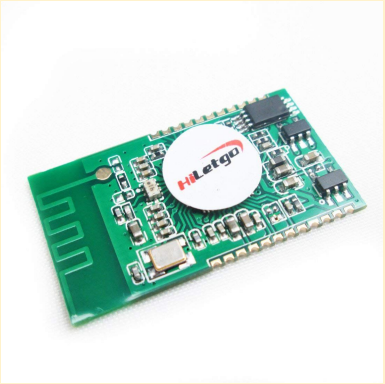
Nexys 4 Artix-7 FPGA



MPU-9255

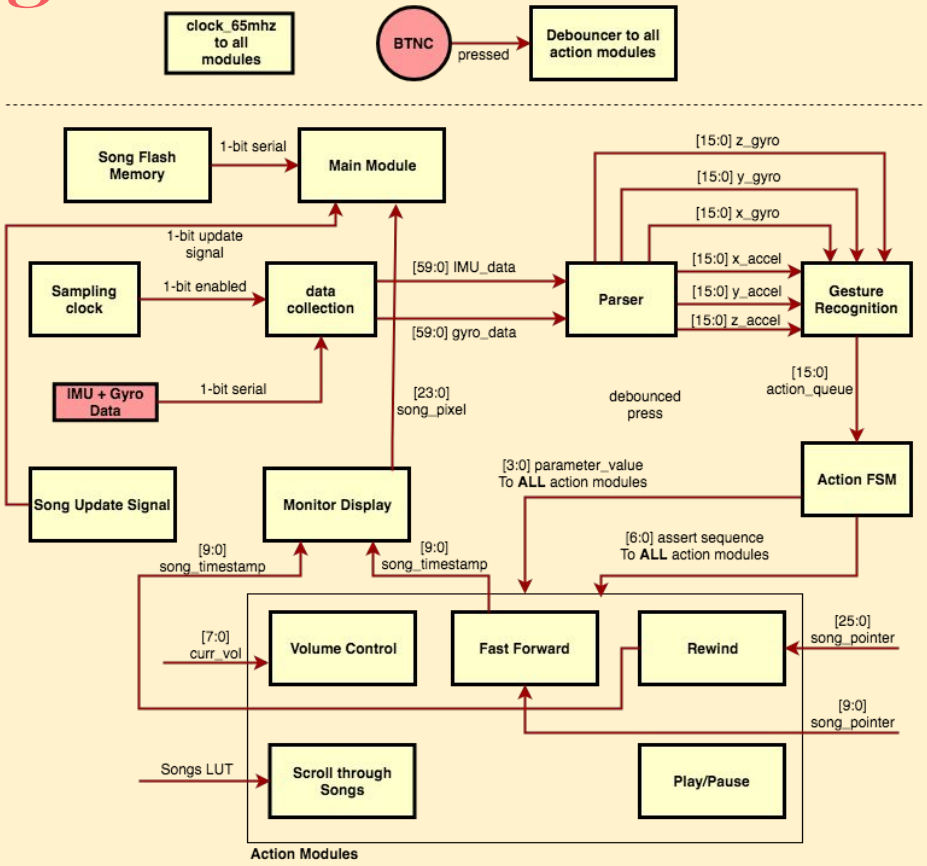


Teensy 3.2



HiLetgo XS3868

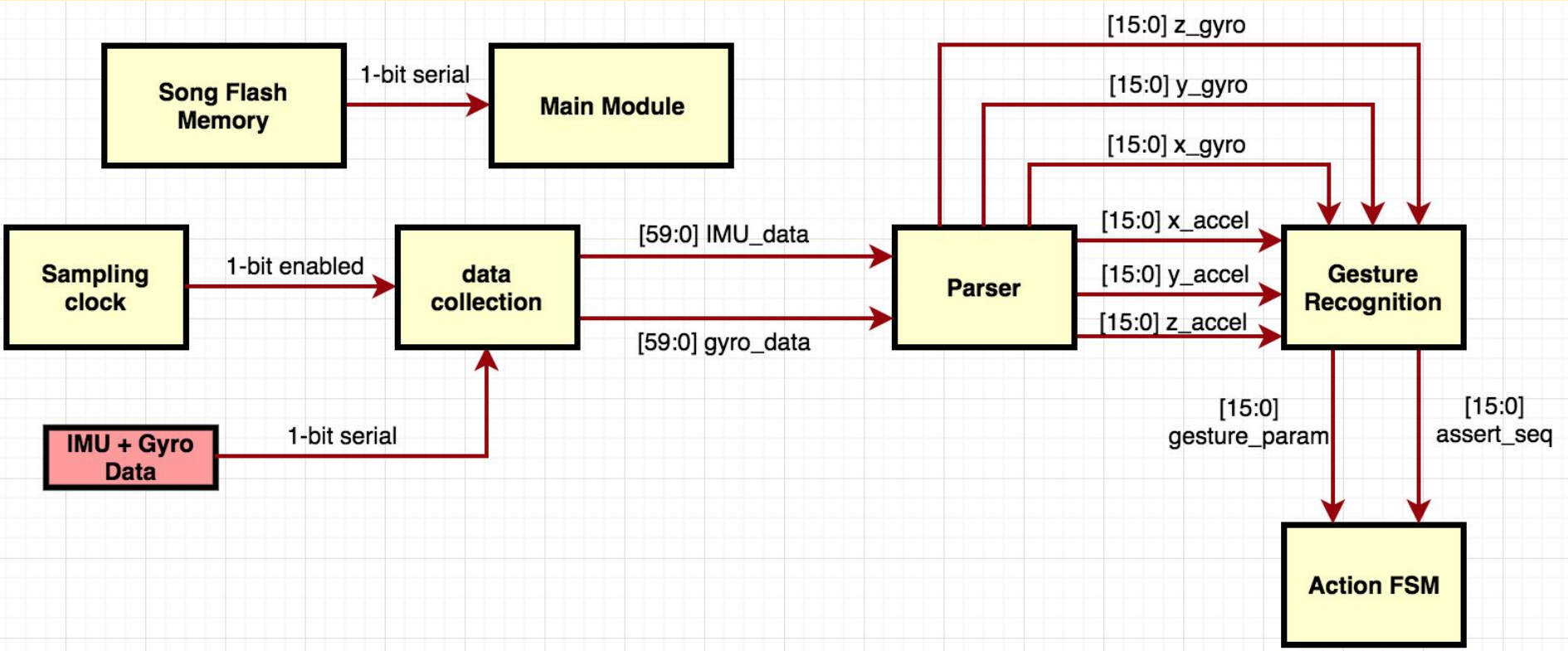
Block Diagram



Major Modules Breakdown

Gesture Recognition Side	Music Playback Side
- Data Collection - Parser - Gesture Recognition FSM - Action FSM	- Volume Control - Song rewind/fast forward - Scroll through songs - Play/pause

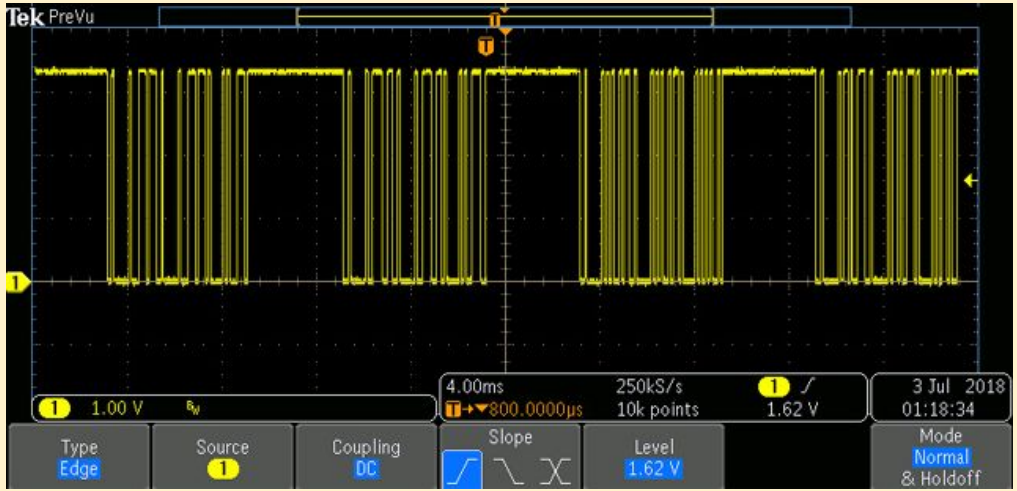
Gesture Recognition

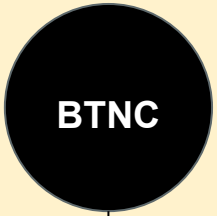


Data Collection + Parsing

Throwback to lab 5c!

- 60 bits x-, y-, z- accelerometer data
- 60 bits x-, y-, z- gyroscope data





While pressed

Default hand position

Move hand down

=

Noticeable change in rotation value around one axis compared to previous 16-bit register rotation value

Gesture Recognition

Acceleration values = Pinpoint orientation of user's hand

$\int_{\text{gesture start}}^{\text{gesture end}}$ **(Gyroscope values)** = Natural rotation of user's hand

Check both readings to determine what gesture was performed

Output 2 arrays: hard coded parameter values and assert signal sequence linked to each unique gesture

2 gestures in 1 gif!

Action FSM

Checks each action's 'activated' flag and makes sure no **constraints** are violated.

- Fast forward and Rewind should NOT happen at the same time
- Play/Pause should NOT happen at the same time as Fast forward and Rewind
- Scroll through songs should NOT happen at the same time as fast forward and rewind

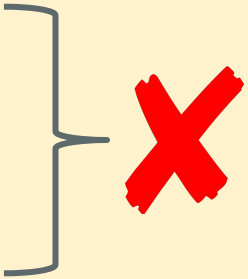
Ex: **Activated flags**: [playing_flag = 1, v_up = 0, v_down = 0, ff_flag = 1, rewind_flag = 0, song_sel = 0]

Parameter Val = 3'b100

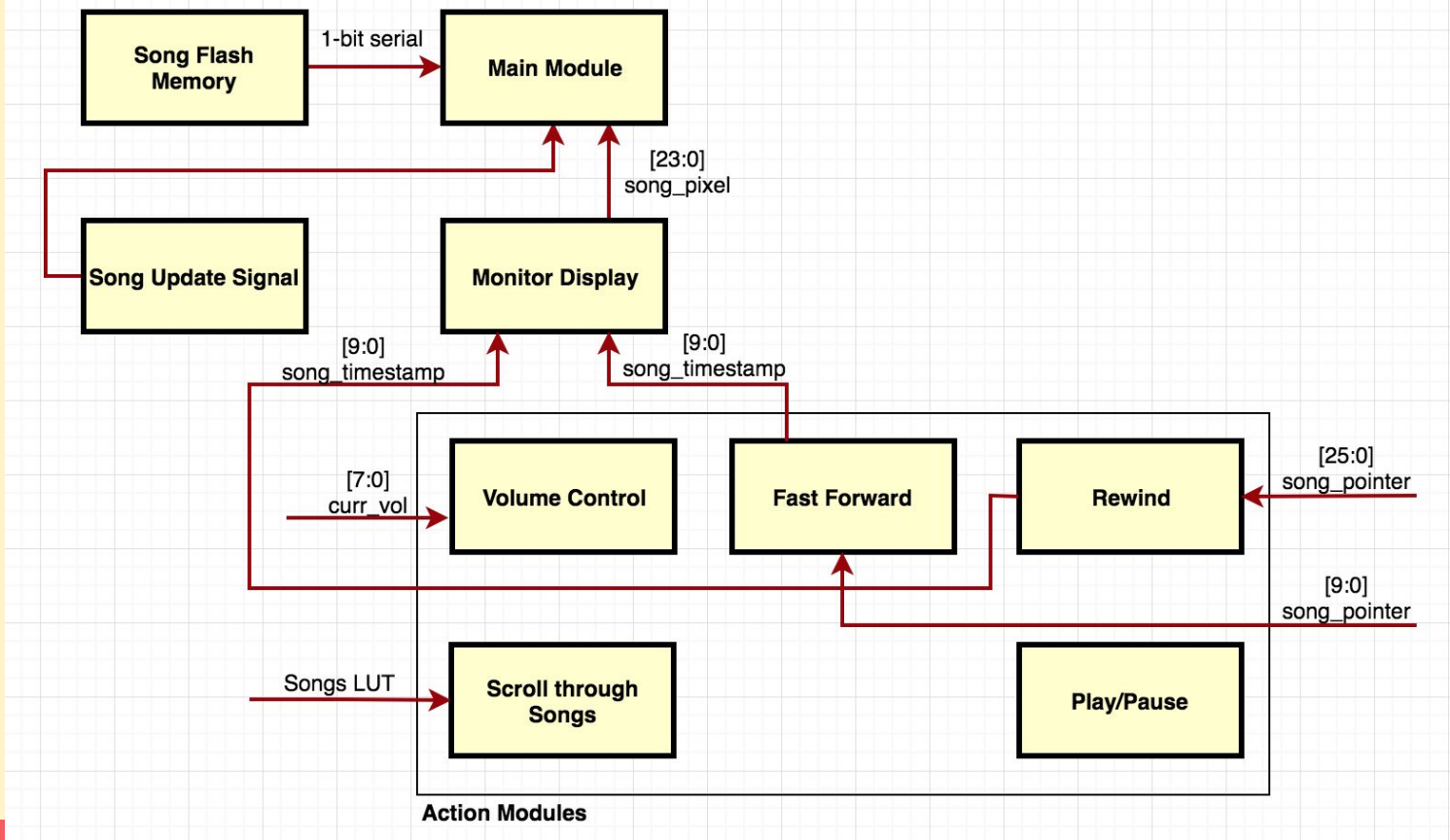
Assert sequence = 4'b1010

Translated Action = Rewind

Check constraints = CONSTRAINT VIOLATED

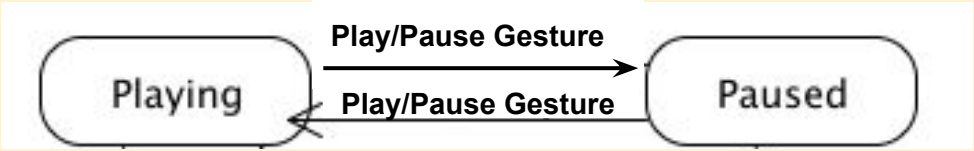


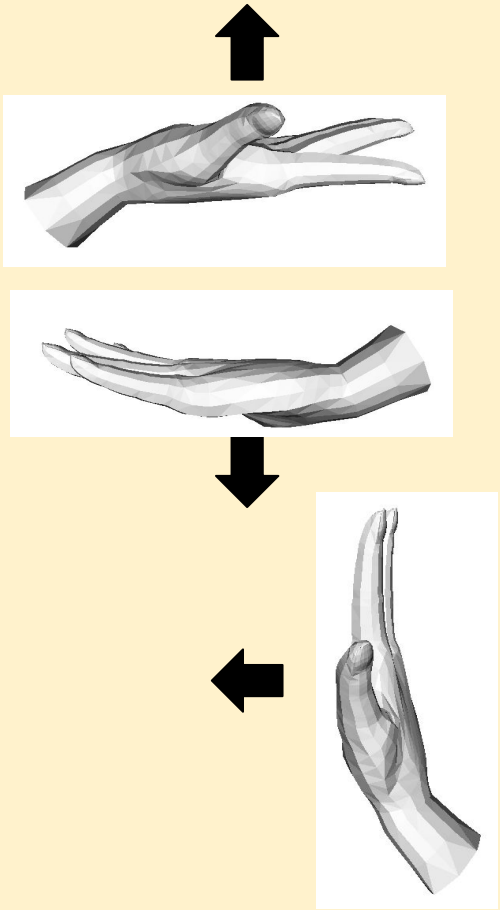
Music Playback



Action Modules

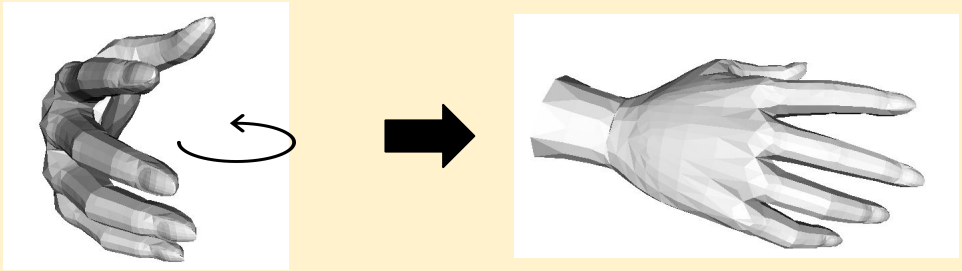
- Volume Control - increases or decreases current volume exponentially
  - our sensation of ‘loudness’ is logarithmic
  - an exponential slider makes sensation of volume variations linear
  - log(exp(x)) = x
- Play/Pause - flips the bit of the register representing the current state of the song



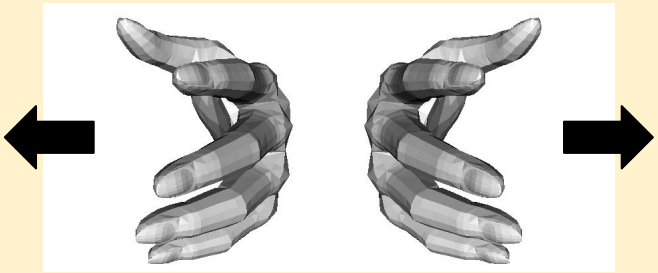


Action Modules

- Fast Forward/ Rewind - changes pointer to current song location to reflect +/- 10 seconds



- Scroll through songs - changes the pointer to the next/previous song's "start" sector in the SD card



Monitor Display

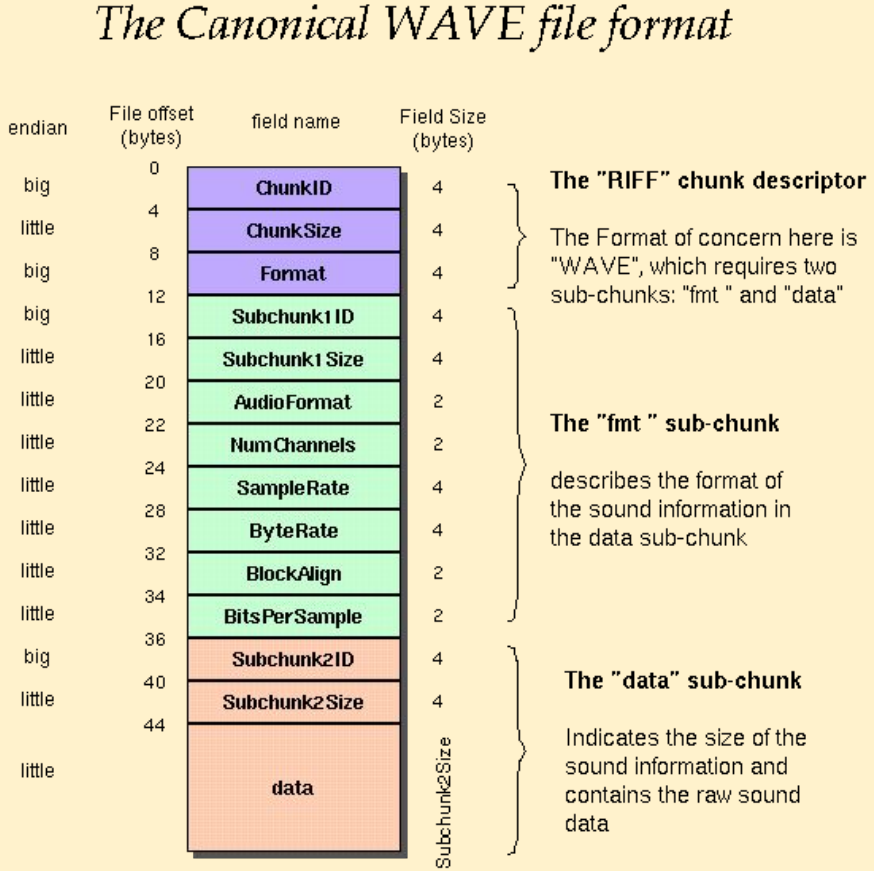
Reads metadata in header of WAV file to display:

- Name of song
- Total length of song

Also using saved register states, displays:

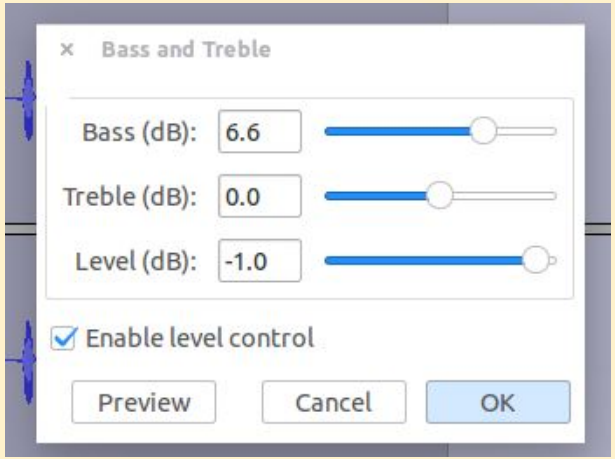
- Current position/progress in song

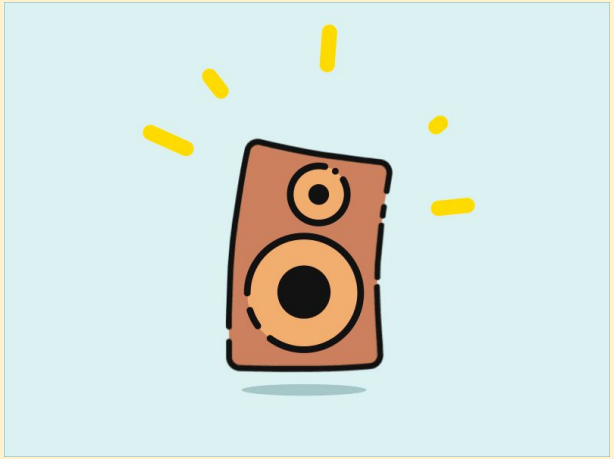
- Play/Pause state



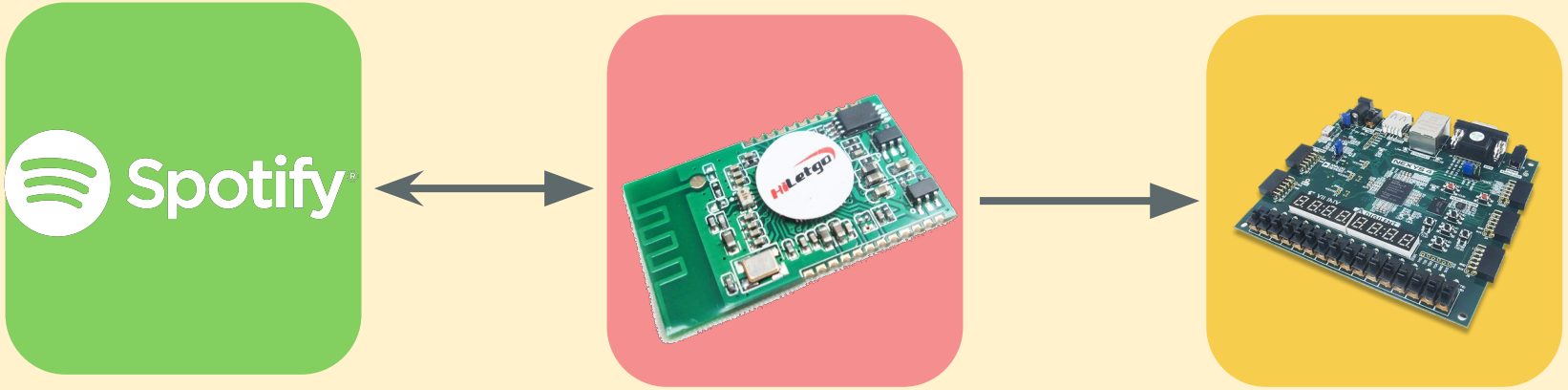
Reach Goals

Implementing bass/treble boosting → Amplifying certain audio input signals based on frequency ranges.





Spotify® Integration



HiLetgo XS3868
Bluetooth Module

Nexys 4 Artix-7 FPGA

Obstacles

- Error minimization (to account for varying gesture positions)
- Queue-ing of gestures and making sure they are executed in the right order

Timeline

Date	Task
November 16th	- Accelerometer & Gyro data collection module completed and tested - Basic song playback module from SD memory completed
November 19th	- Gesture Recognition module done and Action FSM module halfway implemented - Action modules written up and in testing phase - Start of testing all individual modules and planning to integrate modules together
November 26th	- Completion of testing individual modules. - Integration of all modules
December 3rd	- Testing and polishing phase - Cleaning up any functionality bugs
December 10th	- Project complete

Questions?