ECE 540/558 Project Proposal

Jump Rope Game

Chuck Faber, Brett Thornhill, Tiffani Shilts

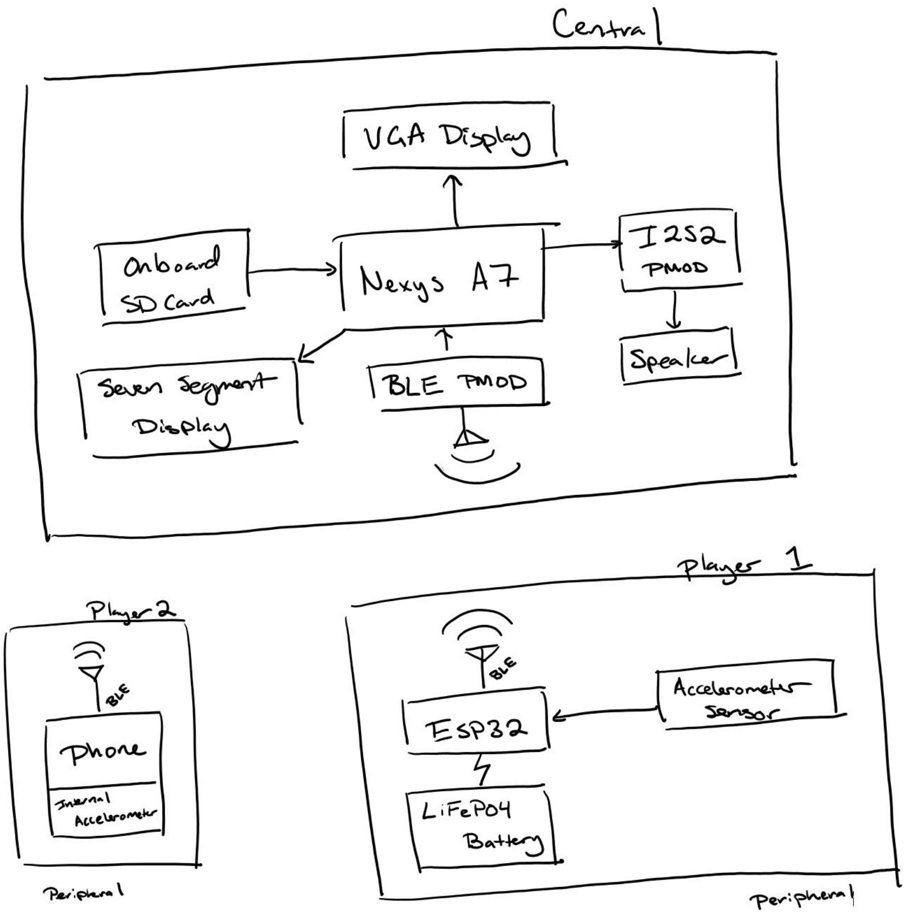
Project Description

Overview

For this project, we propose developing an exercise/rhythm game that requires the player to jump or duck based on the cues provided on a screen. We call this a “Jump Rope” game because the visual cue will be a line on a screen that moves to the bottom of the screen when a player is required to jump, and the top when the player is required to duck, like a real-life game of jump rope.

The actions required will be synced to the beat of one or several songs. In addition, we will implement a two player mode where the second player will use a phone as their wireless sensor running an Android app that sends the motion data from the phone to the game.

Capabilities & Hardware



The game will utilize the Nexys A7 to display the rope to the VGA display and the score on the seven segment display. It will also receive the BLE packets from the ESP32/Accelerometer unit through the BLE PMOD and publish the data to a software accessible register. Songs will be played from the onboard SD card reader to the speaker. Initially we will play to the mono audio output on board the Nexys A7, but a stretch goal will be to implement the I2S2 PMOD for stereo sound.

The hardware will open up software accessible registers for the rope location, a register in which the data from the accelerometer is published and readable by the software, and at least one other register that will allow the software to start the song playback.

Stretch goals of being able to play multiple user-chooseable songs will require the use of a register that allows the writing of a value which can be used to multiplex which song will be played by the hardware, and possibly to trigger the display of extra transient display features to the VGA such as player feedback ("Good", "Perfect!", "Boo", etc.).

One base goal to combine this project with our ECE 558 final project is to implement an android app allowing another player to join in the game as a second player. This app would use the phones internal accelerometer to measure the movement of the player, and it would send it over BLE using the phone's BLE capabilities.

Design Approach

Work Split

Chuck - Implementing BLE, Low-Level Hardware for display, and wireless sensor unit.
Tiffani - Implementing Audio and SD Card Low-Level Hardware
Brett - Implementing Game Software and Android App

Deliverables

Level 1:

Rope Game implemented with

- VGA display
- button input as "jump" or "duck"
- no sound
- score keeping on seven segment display
- start an Android app that measures acceleration data

Level 2:

Rope Game implemented with

- accelerometer sensor sending data over BLE
- mono audio
- one song
- Android app can communicate over BLE

- Android app displays second player score

Level 3:

Rope Game implemented with

- Multiple songs to choose from (⍰)
- Stereo audio using I2S2 PMOD (⍰)
- More display elements such as point or performance icons (⍰)
- Multiple point tiers for critical jumps and duck timing (⍰)
- Android app allows user to preview and select songs (⍰)
- Game displays high scores after song on screen and app (⍰)
- App allows user to create user profile (⍰)

⍰ - Stretch Goal

Milestones

Milestone	Due Date
Project Proposal	February 27th
Level 1 Goals	March 1st
Level 2 Goals	March 10th
Progress Report	March 12th
Level 3 Goals	March 15th
Final Deliverables	March 18th

Concerns and Contingency Plans

Our major concerns are as follows:

- We are unsure how to implement the PMOD hardware with the RVfpga processor in Vivado.
    - **Contingency Plan:** We could possibly develop using the Microblaze processor and Xilinx SDK rather than using PlatformIO and RVfpga. There are more resources online about how to implement PMODs as blocks using the IP integrator, though this raises several other concerns, such as the fact that we have no practice with this processor and would have to learn how to add several features that are already implemented in our current RVfpga architecture.
- There are not a lot of resources regarding playing audio through the Nexys A7, particularly from the SD card.
    - **Contingency Plan:** There are some resources that may help, but if they do not, we may have to implement the project without sound.
- Understanding the BLE communication may be too difficult to complete within the time-frame.

    - **Contingency Plan:** Should this occur, we could attempt to tie the accelerometer directly into the Nexys A7 with really long wires. (Not recommended because of tripping hazards). Should this also not be feasible, we could limit it to just using the push buttons to simulate jumps and ducks.
- We could run out of time.
    - **Contingency Plan:** We would simply submit the game with whatever features we have thus far completed. Our design approach almost guarantees that we will have some product ready by the deadline regardless of the limitations of features due to its multi-level approach towards building complexity.
- The Bluetooth communication may not work or with two players.
    - **Contingency Plan**: We could possibly attach an ESP32 to the Nexys A7 and use the BLE capabilities of it (instead of the PMOD), or run an MQTT server from it.