Review the assignment due date

final-project-skeleton

Team Number: 31

Team Name: 404 not found

Team Member Name	Email Address
Pranay Choudhuri	pranay24@seas.upenn.edu
Haoran Liang	liang9@seas.upenn.edu
Kefei Bao	kefeib@seas.upenn.edu

GitHub Repository URL:

https://github.com/upenn-embedded/final-project-f25-404-not-found

GitHub Pages Website URL: [for final submission]*

Final Project Proposal

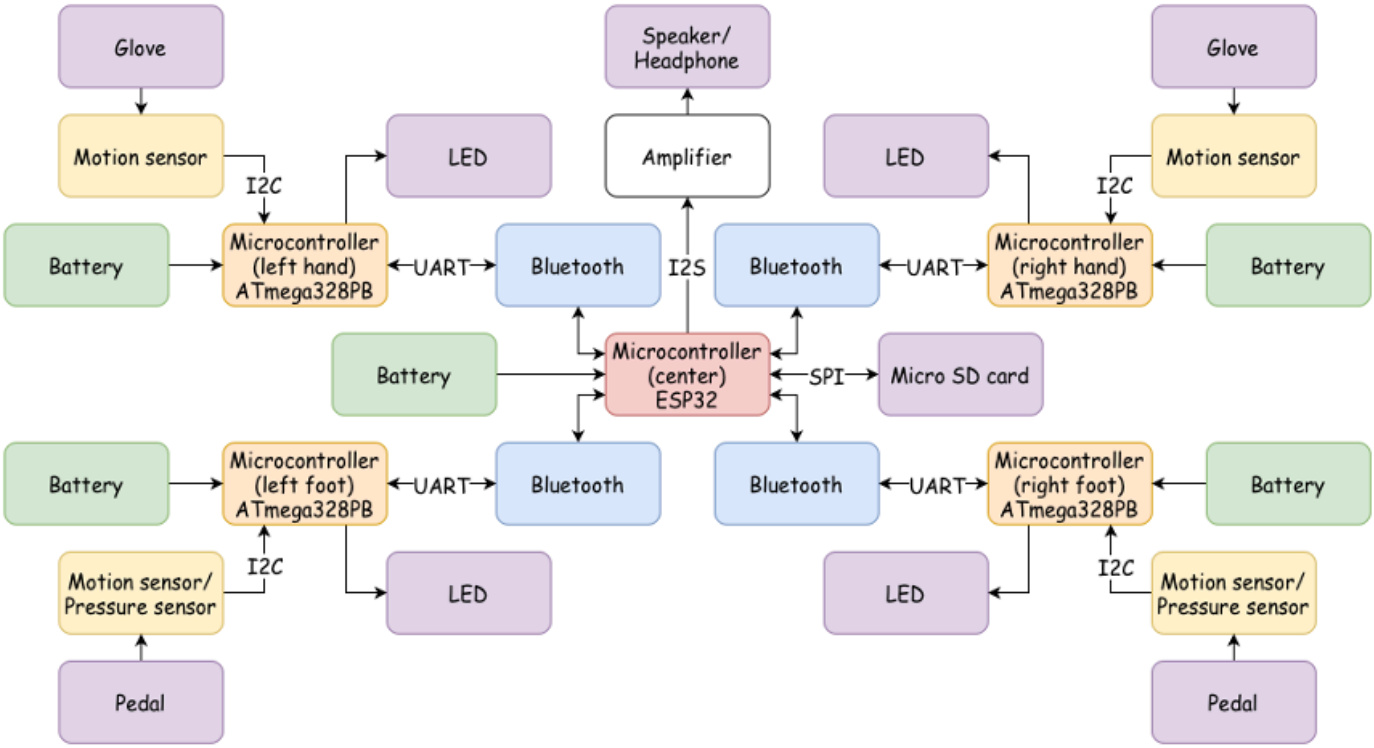
1. Abstract

Our virtual drum kit system is built around multiple sensor nodes, each equipped with a motion sensor, microcontroller, and wireless communication module, all connected to a central hub that serves as the heart of the kit. The hub houses a microcontroller and the necessary hardware to manage communication with the nodes and generate the corresponding drum sounds. Each node is designed to be worn on the hands and feet, and when a motion sensor detects a downward strike, it sends a signal to the hub, triggering the playback of the appropriate drum sound; creating a seamless, immersive drumming experience without the need for a physical drum set.

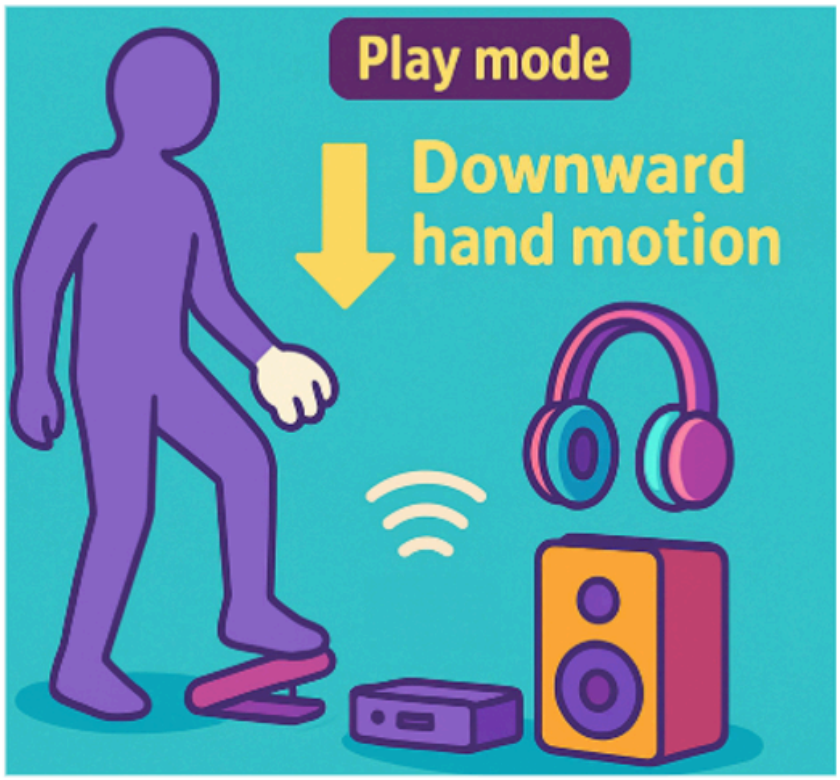
2. Motivation

It's often that we vibe so hard to the rhythm of a song that we instinctively want to drum along only to be reminded that traditional drum kits are expensive, bulky, and cumbersome to set up. Our Virtual Drum Kit bridges this gap by offering an accessible, portable alternative.

3. System Block Diagram



4. Design Sketches



5. Software Requirements Specification (SRS)

Software Requirements Specification (SRS):

1. Low latency in hit to sound
2. Reliable hit detection and low false positives on hit triggers
3. Support stable communication with multiple nodes and Seamless start up
4. Multiple sound effect being played depends on hand motion.

5.1 Definitions, Abbreviations

While we are talking about latency: from downward motion to sound being palyed.
While we are talking about trigger: downward motion that has acceleration larger than 2G
While we are talking about connection: Wi-Fi built in with ESP 32/Bluetooth built in for ESP32 and ATmega328pb

5.2 Functionality

ID	Description
Low latency	Hit to sound less then 100 ms
Reliable hit detection and low false positives on hit triggers	Any motion that is upward shall not trigger any sound, and only downward motion with larger than 2G acceleration is being considered as input.
Support stable communication with multiple nodes and seamless start up	Once power for all nodes are turned on, wireless communication shall auto engage and if faild, shall report error code based off the fact
Multiple sound effect being played depends on hand motion	Like a actual drum kit, our system detecting downward motion with direction and play different instrument recording correspondingly

6. Hardware Requirements Specification (HRS)

Hardware Requirements Specification (HRS):

1. Reliable wired connections that can sustain sharp movements
2. Minimal power consumption
3. Generate an audio that is loud enough
4. IMU responsiveness

5. Light weight

6.1 Definitions, Abbreviations

IMU = Sparkfun LSM6DS0 6DoF
ATmega = ATmega328pb
Amplifier = 1528-1696-ND
Speaker = 668-1234-ND
Power = 1000mAh Bank

6.2 Functionality

ID	Description
Reliable wired connections that can sustain sharp movements	Soldered connection in sensor and center node for final demo.
Minimal power consumption	With power bank of 1000mAh, the system shall work for 1 hour straight.
Generate an audio that is loud enough	Larger than 60dB.
IMU responsiveness	Any motion that is upward shall not trigger any sound, and only downward motion with larger than 2G acceleration is being considered as input.
Light weight	Hand node under 200g *2 , foot node under 300g *2, hand node, foot node and center hub and speaker 500g make it a less then 1.5kg kit

7. Bill of Materials (BOM)

What major components do you need and why? Try to be as specific as possible. Your Hardware & Software Requirements Specifications should inform your component choices.

In addition to this written response, copy the Final Project BOM Google Sheet and fill it out with your critical components (think: processors, sensors, actuators). Include the link to your BOM in this section.

https://docs.google.com/spreadsheets/d/1jcCn7PMXPBg7CsjnHoBdqLJhqJybd3HvZRBbgPiIp-4/edit?gid=2071228825#gid=2071228825

Node:
MPU: ATmega328pb
Communication module: HC-05(Bluetooth module)
• Low cost, low power Bluetooth module
Sensor: LSM6DS0
• IMU that we already have prior experience with
Power: Power Bank

Central Hub:
MPU: ESP32
• Provides enough compute power along with built-in wifi and bluetooth.
Amplifier: 1528-1696-ND (Digikey)
• Receives the audio from the MCU in digital format over I2S. Handles conversion and amplification
Speaker: 668-1234-ND (Digikey)
• Supports the output of the amplifier
Power: Power Bank

8. Final Demo Goals

How will you demonstrate your device on demo day? Will it be strapped to a person, mounted on a bicycle, require outdoor space? Think of any physical, temporal, and other constraints that could affect your planning.

Our final demo shall be fairly simple. We will strap on the components and play a groove while sitting on a chair.

9. Sprint Planning

You've got limited time to get this project done! How will you plan your sprint milestones? How will you distribute the work within your team? Review the schedule in the final project manual for exact dates.

Milestone	Functionality Achieved	Distribution of Work
Sprint #1	Acquire hardware and start setting up base code	Pranay:reading ESP and ATmega for wireless communication; Haoran: reading manual for motion sensor; Kefei: reading manual for amplifier and speaker

Milestone	Functionality Achieved	Distribution of Work
Sprint #2	develop full code for one node and the amplifier/speaker drive	Pranay: coding for wireless communication; Haoran: coding for motion detection; Kefei: coding for sound effect storage/call and amplifier/speaker driving
MVP Demo	scale up to 4 nodes/ Putting it all together	debugging together
Final Demo	Motion detect fine tuning and more sound effect	debugging together

This is the end of the Project Proposal section. The remaining sections will be filled out based on the milestone schedule.

Sprint Review #1

Last week's progress

Current state of project

Next week's plan

Sprint Review #2

Last week's progress

Current state of project

Next week's plan

MVP Demo

1. Show a system block diagram & explain the hardware implementation.
2. Explain your firmware implementation, including application logic and critical drivers you've written.

3. Demo your device.
4. Have you achieved some or all of your Software Requirements Specification (SRS)?
    i. Show how you collected data and the outcomes.
5. Have you achieved some or all of your Hardware Requirements Specification (HRS)?
    i. Show how you collected data and the outcomes.
6. Show off the remaining elements that will make your project whole: mechanical casework, supporting graphical user interface (GUI), web portal, etc.
7. What is the riskiest part remaining of your project?
    i. How do you plan to de-risk this?
8. What questions or help do you need from the teaching team?

Final Project Report

Don't forget to make the GitHub pages public website!
If you've never made a GitHub pages website before, you can follow this webpage (though, substitute your final project repository for the GitHub username one in the quickstart guide): https://docs.github.com/en/pages/quickstart

1. Video

[Insert final project video here]

- The video must demonstrate your key functionality.
- The video must be 5 minutes or less.
- Ensure your video link is accessible to the teaching team. Unlisted YouTube videos or Google Drive uploads with SEAS account access work well.
- Points will be removed if the audio quality is poor - say, if you filmed your video in a noisy electrical engineering lab.

2. Images

[Insert final project images here]

Include photos of your device from a few angles. If you have a casework, show both the exterior and interior (where the good EE bits are!).

3. Results

What were your results? Namely, what was the final solution/design to your problem?

3.1 Software Requirements Specification (SRS) Results

Based on your quantified system performance, comment on how you achieved or fell short of your expected requirements.

Did your requirements change? If so, why? Failing to meet a requirement is acceptable; understanding the reason why is critical!

Validate at least two requirements, showing how you tested and your proof of work (videos, images, logic analyzer/oscilloscope captures, etc.).

ID	Description	Validation Outcome
SRS-01	The IMU 3-axis acceleration will be measured with 16-bit depth every 100 milliseconds +/-10 milliseconds.	Confirmed, logged output from the MCU is saved to "validation" folder in GitHub repository.

3.2 Hardware Requirements Specification (HRS) Results

Based on your quantified system performance, comment on how you achieved or fell short of your expected requirements.

Did your requirements change? If so, why? Failing to meet a requirement is acceptable; understanding the reason why is critical!

Validate at least two requirements, showing how you tested and your proof of work (videos, images, logic analyzer/oscilloscope captures, etc.).

ID	Description	Validation Outcome
HRS-01	A distance sensor shall be used for obstacle detection. The sensor shall detect obstacles at a maximum distance of at least 10 cm.	Confirmed, sensed obstacles up to 15cm. Video in "validation" folder, shows tape measure and logged output to terminal.

4. Conclusion

Reflect on your project. Some questions to address:

- What did you learn from it?
- What went well?

- What accomplishments are you proud of?
- What did you learn/gain from this experience?
- Did you have to change your approach?
- What could have been done differently?
- Did you encounter obstacles that you didn’t anticipate?
- What could be a next step for this project?

References

Fill in your references here as you work on your final project. Describe any libraries used here.