Review the assignment due date

NIBLETS FINAL PROJECTION

- Team Number: 12
- Team Name: Niblets
- Team Members: Claren Ogira, Patrick Zhao, Praise Ndlovu

- GitHub Repository URL: https://github.com/upenn-embedded/final-project-s25-niblets
- GitHub Pages Website URL: [for final submission]

Final Project Proposal

1. Abstract

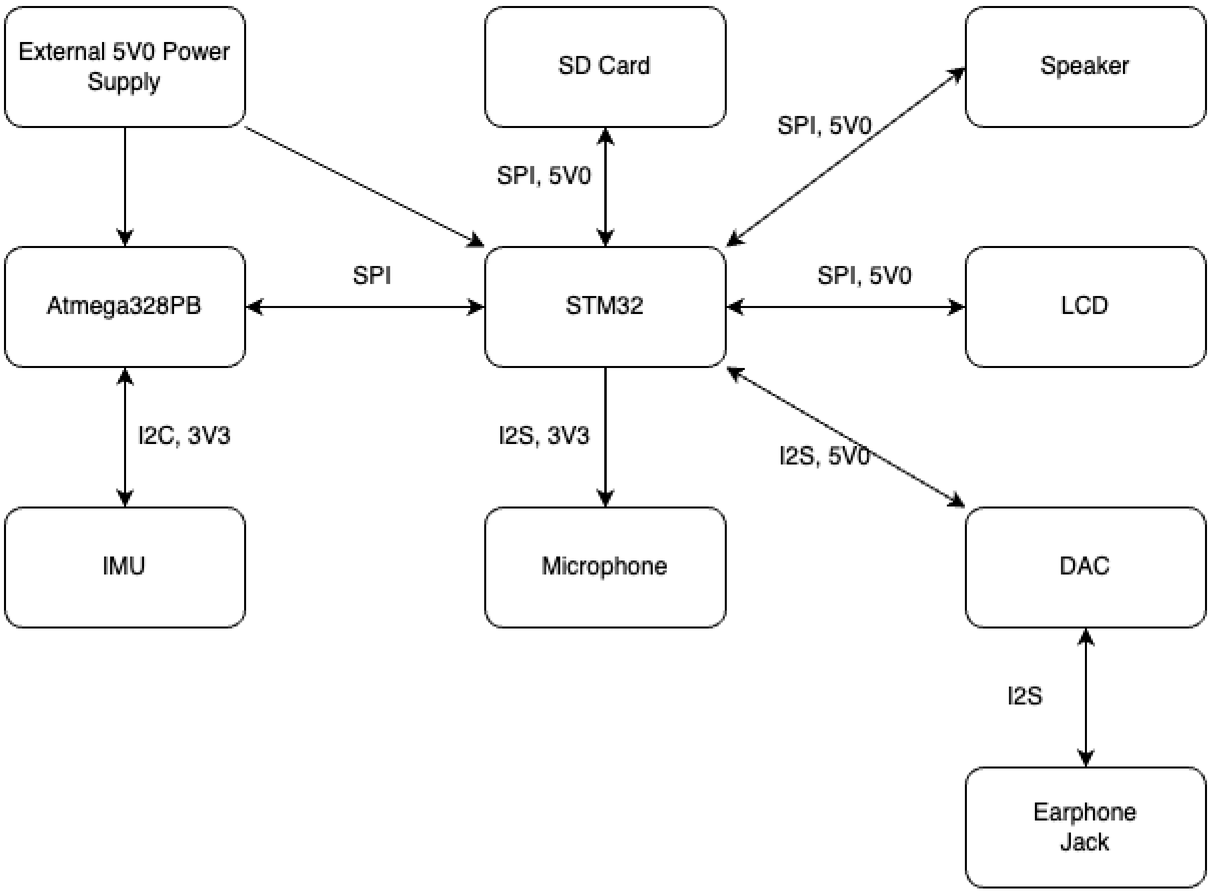
This project develops a digital audio device integrating an LCD, microphone, SD card, speaker, and microcontrollers for seamless audio playback, recording, and user interaction. The system ensures efficient communication, real-time performance, and reliable data storage. Validation includes display responsiveness, audio quality, and peripheral interfacing to optimize functionality and usability.

2. Motivation

This project builds a standalone digital audio device for recording, playback, and user interaction. It's a hands-on dive into writing graphics, mastering serial protocols, and making hardware talk seamlessly. The goal? A fully functional, ready-to-use device—while leveling up in embedded systems and low-level programming!

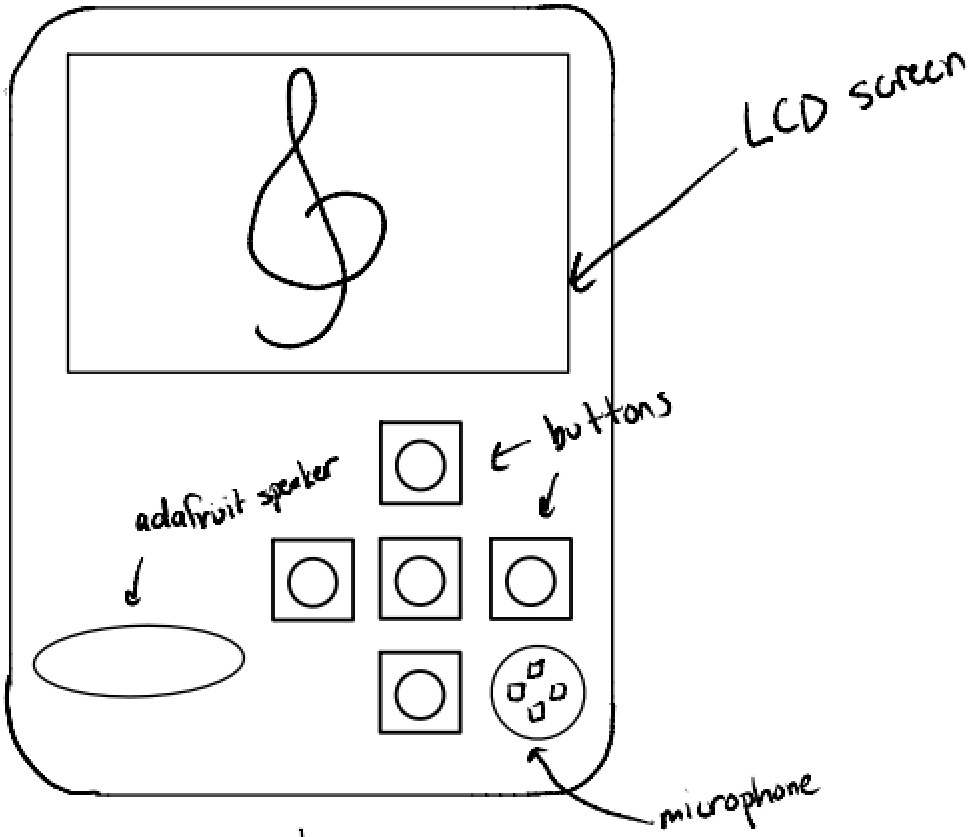
3. System Block Diagram

Show your high level design, as done in WS1 and WS2. What are the critical components in your system? How do they communicate (I2C?, interrupts, ADC, etc.)? What power regulation do you need?

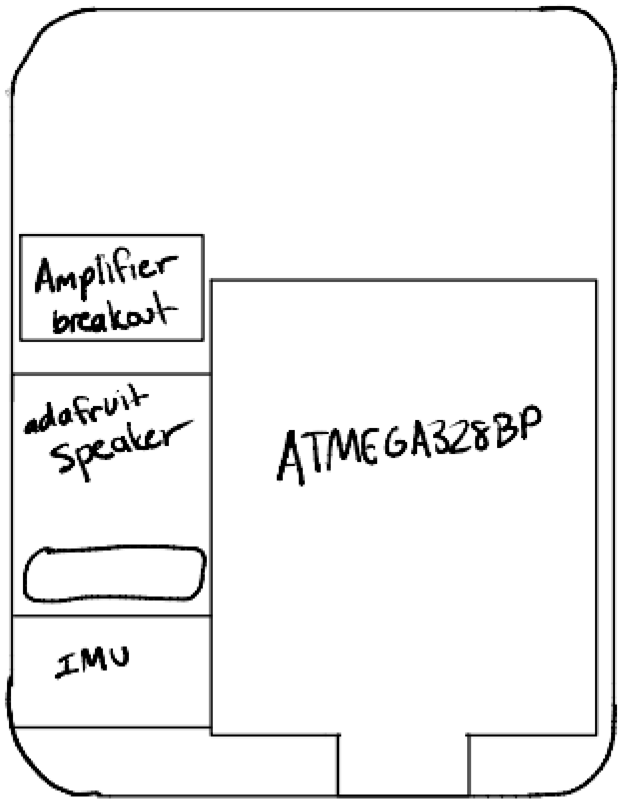


4. Design Sketches

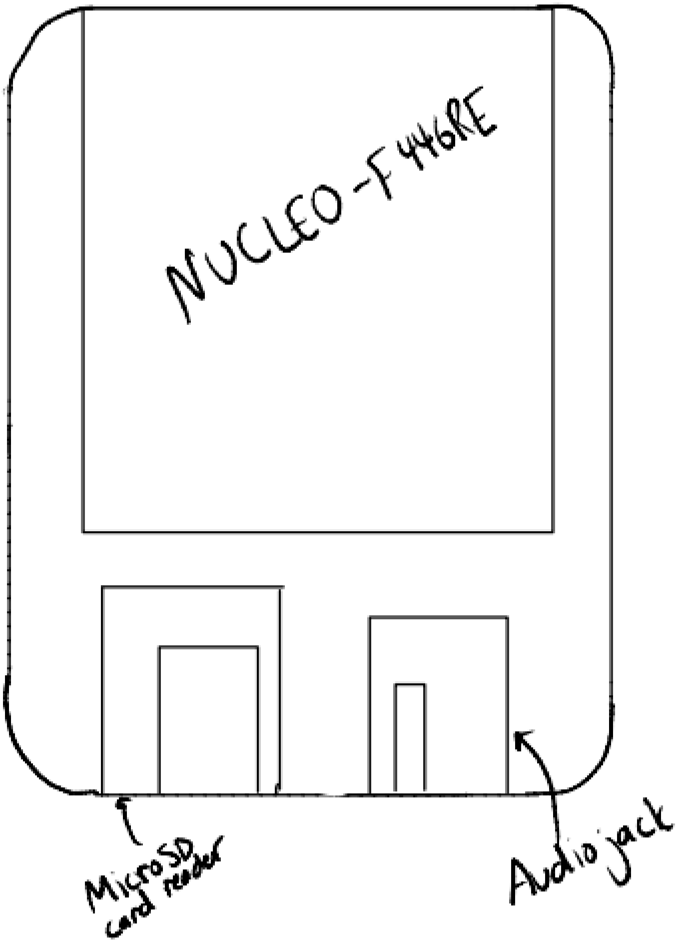
What will your project look like? Do you have any critical design features? Will you need any special manufacturing techniques to achieve your vision, like power tools, laser cutting, or 3D printing?

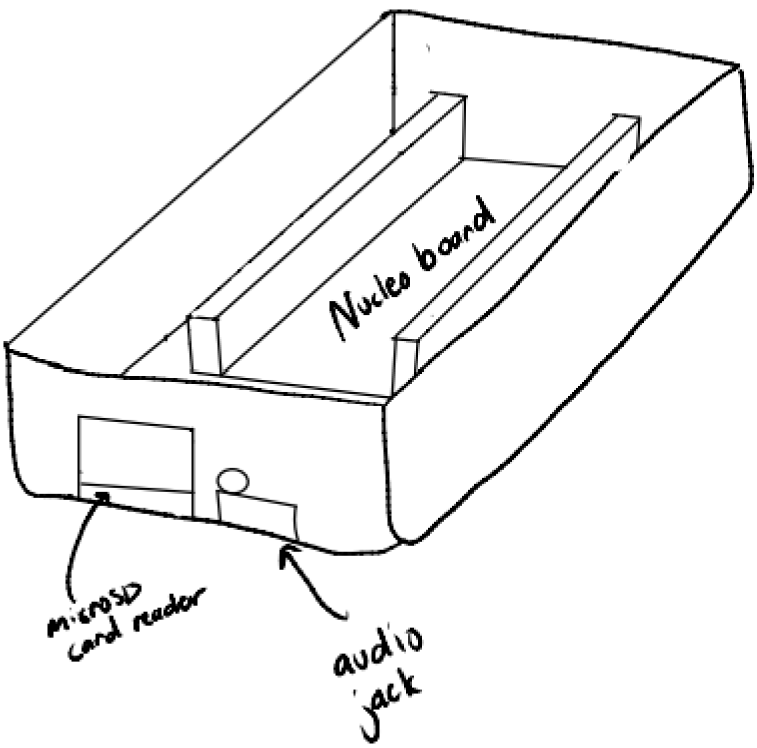


Layer 1



Layer 2





We wll need 3D printing to print out a case for fitting all of our components in. We may also require screws to place our components in.

5. Software Requirements Specification (SRS)

Formulate key software requirements here. Think deeply on the design: What must your device do? How will you measure this during validation testing? Create 4 to 8 critical system requirements.

These must be testable! See the Final Project Manual Appendix for details. Refer to the table below; replace these examples with your own.

5.1 Definitions, Abbreviations

Here, you will define any special terms, acronyms, or abbreviations you plan to use for hardware

5.2 Functionality

ID	Description
SRS-01 - LCD Display Functionality	The LCD must accurately render graphics and display the current state of the iPod, including menus, playback status, and user interactions.**Validation**: Verify display clarity, responsiveness, and update rate during different operations.
SRS-02 Microphone Recording Capability	The microphone must capture audio input with sufficient clarity and store it in a digital format when enabled.**Validation**: Record test samples and analyze audio quality, noise levels, and latency.
SRS-03 SD Card Storage & Read/Write Operations	The SD card module must support reading and writing data efficiently, ensuring reliable storage of music, recorded audio, and system logs.**Validation** Conduct read/write speed tests, check file integrity, and test compatibility with different SD card sizes.
SRS-04 Speaker Audio Output	The speaker must play sound accurately and at a sufficient volume without distortion when provided with an audio signal.**Validation** Measure frequency response, output power, and signal clarity under different playback conditions.
SRS-05 Microcontroller (MCU) Interfacing	The microcontrollers must communicate seamlessly with each other and with peripherals such as the LCD, microphone, SD card, and speaker.**Validation** Perform communication protocol tests (e.g., I2C, SPI, UART),

ID	Description
	measure data transfer latency, and ensure stability under various loads.
SRS-06 User Interface Responsiveness	The system shall provide an intuitive and responsive user interface for navigation and control.**Validation** Test button responsiveness, menu transitions, and overall usability during different operations.

6. Hardware Requirements Specification (HRS)

ID	Description
HRS-01 - LCD Display	The LCD display must have a minimum resolution of 480x320 pixels and support clear, legible display of menus, playback status, and user interactions**Validation**: Verify display clarity, resolution, and update rate during different operations.
HRS-02 - Microphone	The microphone must capture audio with high fidelity and low noise, providing clear input for recording and voice commands.**Validation**: Record test samples, analyze audio quality, noise levels, and latency.
HRS-03 - SD Card Module	The SD card module must support reading and writing at fast speeds to efficiently handle large files such as audio and system logs.**Validation**: Conduct read/write speed tests, check file integrity, and test compatibility with different SD card sizes.
HRS-04 - Speaker	The speaker must provide clear and undistorted audio output, with sufficient volume for playback in typical environments.**Validation**: Measure frequency response, output power, and signal clarity under different playback conditions.
HRS-05 - Microcontroller (MCU)	The microcontroller must manage the system's components, handle user input, audio processing, and communication between peripherals.**Validation**: Perform communication protocol tests (e.g., I2C, SPI, UART), measure data transfer latency, and ensure stability under various loads.

ID	Description
HRS-06 - Power Supply	The power supply must support efficient power management to ensure optimal device operation and battery longevity.**Validation**: Measure power consumption under different workloads and test battery performance over extended usage.
HRS-07 - Buttons & User Interface Controls	Physical buttons or touch interface must allow the user to navigate the system and control media playback and settings.**Validation**: Test button responsiveness, UI transitions, and overall usability during different operations.

6.1 Definitions, Abbreviations

Here, you will define any special terms, acronyms, or abbreviations you plan to use for hardware

6.2 Functionality

ID	Description
HRS-01	The audio player shall support playback of audio files from the SD card, with support for common formats such as MP3 and WAV.
HRS-02	The device shall include a play/pause button that toggles audio playback without delay.
HRS-03	The volume control shall allow the user to adjust audio output levels from silent to a maximum of 85 dB.
HRS-04	The audio player shall support track navigation, allowing the user to skip forward or backward between audio tracks.
HRS-05	The device shall display the current track information, including title and duration, on the LCD screen.
HRS-06	The speaker shall output clear and undistorted audio at all playback levels.
HRS-07	The audio player shall automatically resume playback from the last position upon power-on, if no other track is selected.

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.

Some components that we will need are the ATMEGA328PB which we intend to utilize with the IMU since we already know how the two devices interface given our worksheet experience. We want to use the NUCLEO-F446RE as our main process since it is able to process mode data, which will be required when we are storing and playing music data, and since it has more pins that will allow us to interface with all of our components.

Some output components that we are using are our adafruit speaker, which we find to be standard, an audiojack to allow earbud use and an LCD display touch panel. We chose the LCD display with the touch functionality because of both its touch functionality and because of the bigger display and its wider range of colors.

Our choice of the omnidirectional microphone both reflected on the device's small size and the device's ability to internally process the data that it receives with its internal DAC. We realized that an internal DAC within the microphone would save us tremendous time when building out our device. The same internal DAC reasoning was used in our selection of the amplifier used.

Finally, the IMU we chose reflected an IMU that we are already familiar with and we chose larger buttons for a more consumer-oriented feel.

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/1H9uE5A8rThZXxHaYCtSz2vRuI-nbI0XPYlmzn1kPBnA/edit?gid=2071228825#gid=2071228825

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.

We will demonstrate our project in class. The end product will be small enough to fit on a table. The device may be connected to an external power source to supply power to the system. A quiet setup may be required to hear the audio produced by the device, and to record audio using the microphone.

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	MicroSD Card Reading. LCD turning on with basic graphic functionality. Microphone recording voice. IMU sending understandable data to device. Speaker should be playing some audio that is controllable by both processors. Serial protocols all should be completed as well	Claren - Microphone voice recording, Praise - LCD turning on with basic graphics, Patrick - Speaker playing audio and SD card reading
Sprint #2	Start integration of components. Get audio jack working with amplifier. Start playing music through SD card read through speaker and be able to store recorded audio. Buttons should start controlling board features. LCD screen should display a more understandable graphic. CAD of case should be designed.	Patrick - Design CAD of case. Praise - Playing music through SD card and selection through LCD screen. Claren - Connect audio jack with amplifier and play music through earbuds.
MVP Demo	All components connected and fit inside case. Working minimal demo of song selection and music playing through device.	Patrick - Assembling case, soldering needs. Praise - Connect all of the software components together to play audio. Claren - Work on graphics of LCD screen to allow song selection and better experience.

Milestone	Functionality Achieved	Distribution of Work
Final Demo	Hopefully little to no work left here. Finish any features that were left unfinished from MVP. Program better graphics in the processor for the LCD	Patrick, Praise, Claren - Work on assorted problems to debug at finish ine.

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)?
    1. Show how you collected data and the outcomes.
5. Have you achieved some or all of your Hardware Requirements Specification (HRS)?
    1. 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?
    1. 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.