final-project-skeleton

- Team Number: 14
- Team Name: Stack Overflowers
- Team Members: Sofia Hedlund and Amanda Tapia-Garbelloto

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

Final Project Proposal

1. Abstract

In a few sentences, describe your final project.

We want to make a braille printer that the user can input text into with a keyboard and have it print out the text into readable braille bumps. We plan to do this by 3D printing braille letters onto a stamp that will then rotated according to the text input and stamped down onto the paper with a motor. This project will use power management, interrupts for reading from the keyboard, and timers to control the stepper motor.

2. Motivation

What is the problem that you are trying to solve? Why is this project interesting? What is the intended purpose?

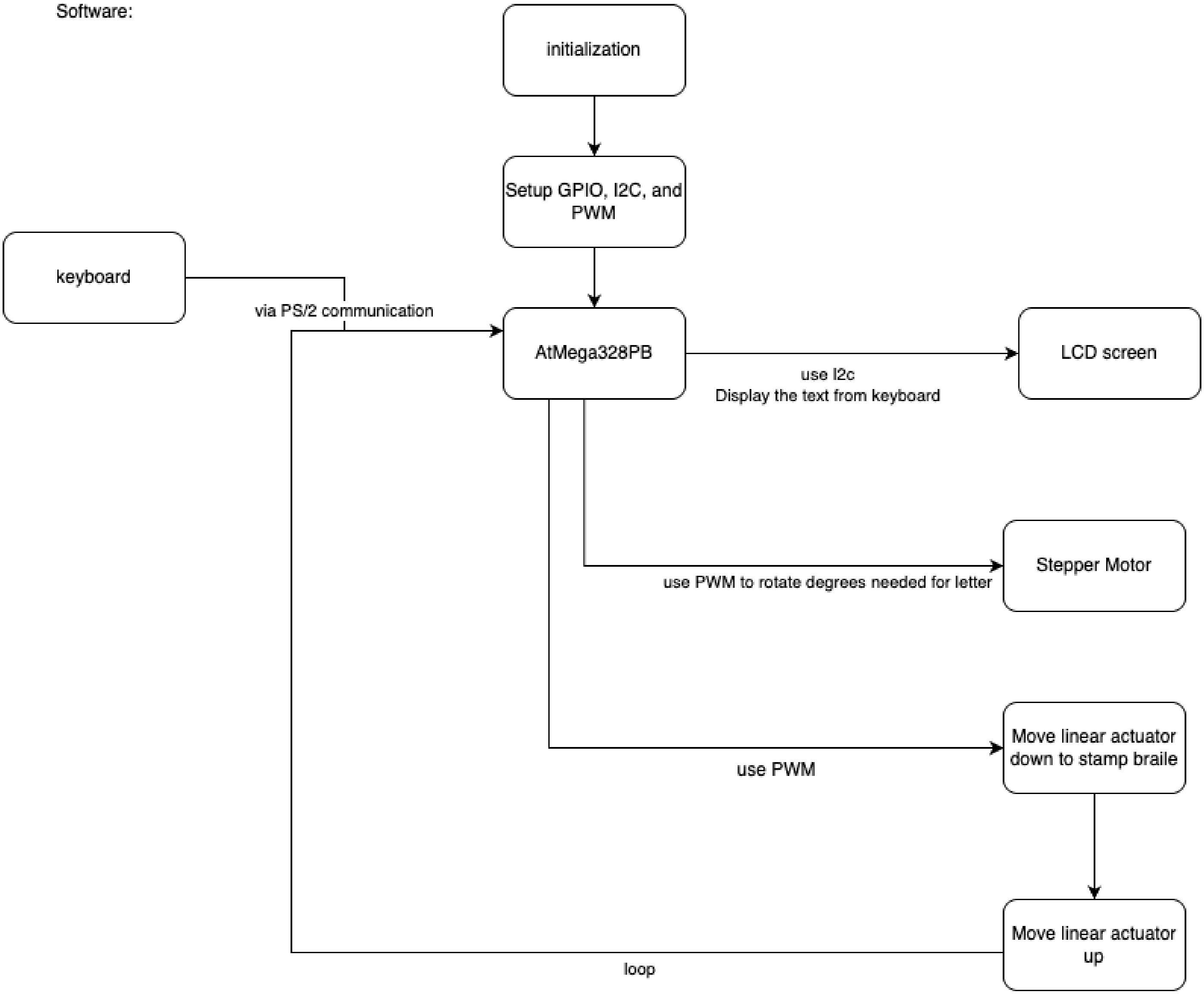
The project aims to enhance accessibility for people with reduced vision. By automatically transforming text into braille, people who can't see text are able to expand their integration to reading materials by possessing a personal device that can make any text available for them without requiring extra translators or specialized equipment.

3. System Block Diagram

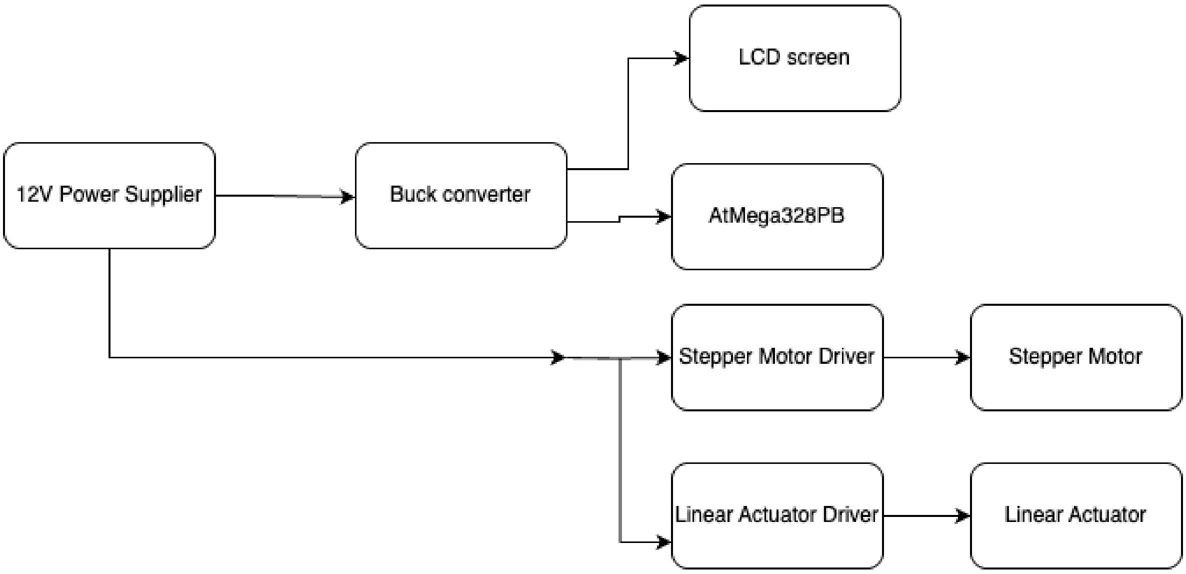
Show your high level design, as done in WS1 and WS2. What are the critical components in your system? How do they communicate (I2C?, interrupts, ADC, etc.)?

What power regulation do you need?

Software:

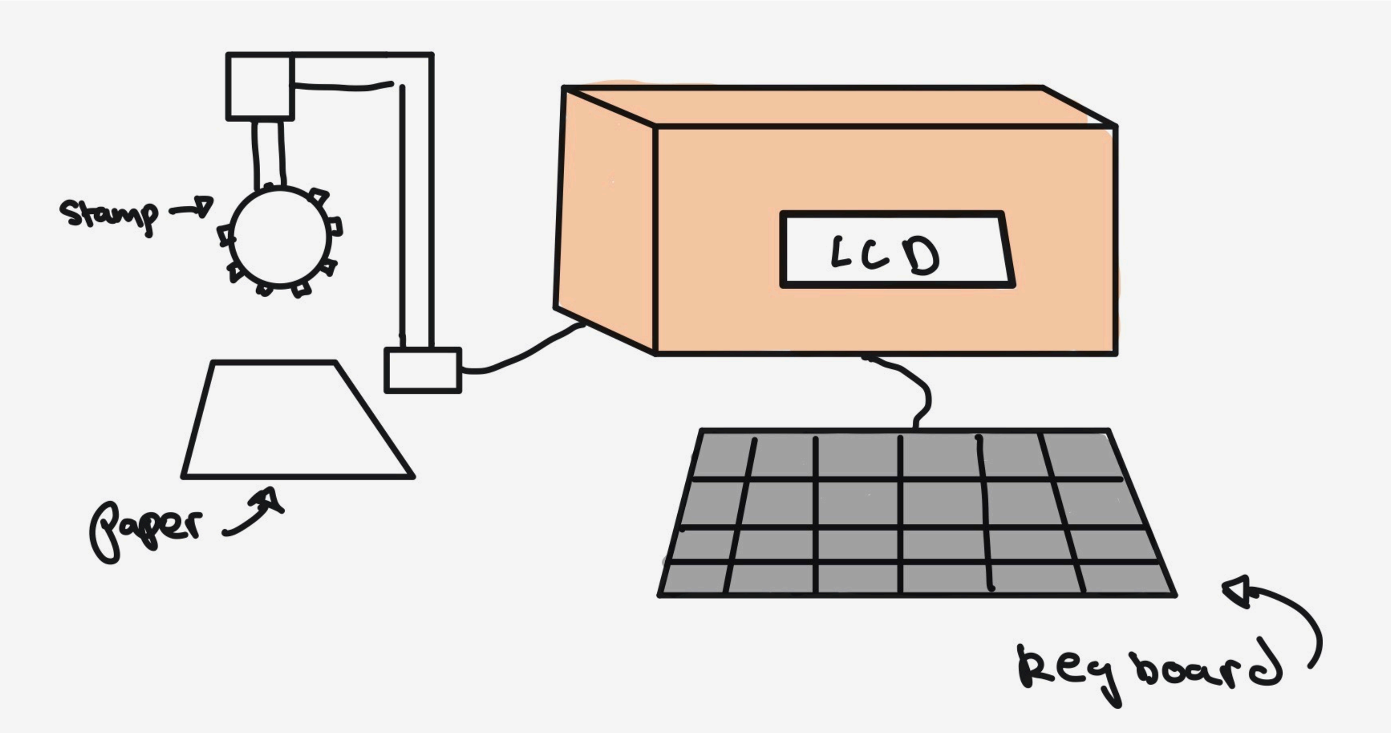


Hardware:



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?



We will need to 3D print the braille stamp to attach to the motor and also need some kind of container to hold the LCD and the wiring needed to power the system. We will also have to 3D print a stand to hold the linear actuator.

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

PWM- Pulse Width Modulation

ASCII- American Standard Code for Information Exchange

LCD- Liquid Crystal Display

5.2 Functionality

ID	Description
SRS-01	The microcontroller will process each letter of the text input and convert it to its corresponding braille.
SRS-02	This recognition will relate the letters to a certain degree of rotation in our stamp machine.
SRS-03	The system will then send PWM signals to control the rotation of our stamp.
SRS-04	Stamping should be at precise timing, so that the stepper motor can finish its rotation completely before stamping to prevent misalignment
SRS-05	The software will have debouncing to filter out noise from the keyboard.

6. Hardware Requirements Specification (HRS)

Formulate key hardware 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.

6.1 Definitions, Abbreviations

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

Stepper Motor- DC motor that has very precise conotrol that will help us avoid error when rotating the stamp to the position of the letter.

Linear Actuator- Moves in a straight line so that we can move the stamp up and down.

6.2 Functionality

ID	Description
HRS-01	The device will require a keyboard module for text entry, this will communicate with the ATmega238 via PS/2.
HRS-02	A LCD screen will display the characters typed into the keyboard..
HRS-03	A stepper motor to rotate our braille stamp.
HRS-04	A linear actuator will move the stamp up and down to imprint the letters on to the paper
HRS-05	The power system will need 5V and 12V to power the MCU, LCD, Stepper, and Linear Actuator.

6. 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.

The major components that we need are a stepper motor/driver to spin the braille stamp and a linear actuator/driver to move the stamp up and down. We chose our stepper motor in order have a lot precision over the amount of degrees the motor turns, so that the correct braille letter is stamped. We also chose for it to be 12V in order to ensure that it will have enough torque required to spin the braille stamp accurately. The driver will let us send simple STEP and DIR signals to control the stepper which will help us easily control the stepper. We chose a linear actuator to stamp down and a H-bridge so that we can move the actuator both up and down. Our budget was a main factor in choosing our linear actuator, because a lot of them were quite pricey (around 70$), so we chose the most affordable option we could find that fit with our system. We chose our keboard to be able to communicate over PS/2 because it is simpler to work with then USB communication which would likely need outside libraries. We will also use the LCD screen from Lab 4 and use the ATmega328Pb Xplained mini board.

BOM

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 device on demo day by showing it printing braille writing from the keyboard onto paper. Some physical constraints are that we need to have enough table space for the keyboard and the braille stamp and because we are stamping letter-by-letter, printing out a word or a short phrase will be more realistic than a full sentence.

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	-3D print braille stamp -Code the rotating for the stamp	3D printing- Amanda Code- Sofia
Sprint #2	-add keyboard/LCD screen -3D print parts to move stamp up and down -debug rotation issues	Keyboard/LCD- Sofia 3D print- Amanda Debug- both
MVP Demo	-create code to move stamp up and down -implement movement with linear actuator	Code/movement- both
Final Demo	-debug issues with motors	Debug- both

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.