Product: BraillEd

Team: UpliftEd

Abstract

Our product, BraillEd, will be composed of inter-connected keyboard, braille display and speaker/microphone functionality, allowing for inputs and outputs to be freely enabled and translated between to serve as a braille learning system.

From the hardware perspective, we have developed a standalone braille half-cell actuator to demonstrate the mechanical function of our braille display. While unrefined, it gives a clear outline of the premise of this sub-system, with indication of how it can be expanded into a full display while also meeting the robustness requirements of the full system. Additionally, time has been spent on integrating hardware components with our central controller in anticipation of the various systems being combined into a single device.

From the software perspective, we have developed speech to braille translation. Speech recognition was completed using VOSK – an open-source speech recognition toolkit that converts speech to text. The recognised real-time text was integrated with our implementation of text to contracted braille translation. This output is useful to direct the movement of the braille cells.

1. Project management update

1.1. Achievement of goals

We have mostly achieved our goals for this demo:

- Decide on a design for a braille display cell mechanism [achieved]
- Implement a mechanism for at least a single braille display cell [partly achieved]
- Implement speech recognition [achieved]
- Implement text to on-screen braille representation [achieved]
- Integrate speech recognition with on-screen braille representation [achieved]
- Produce interface to demonstrate the operation of the software [partly achieved]

Developing a braille cell mechanism took longer than we expected, which resulted in our prototype mechanism only working for half a braille cell, and without actual braille dots on the wheel. This is because we spent many hours implementing various alternatives such as ones using electromagnets. However, this is not time wasted because we now are more confident in our idea.

Furthermore, we decided it was not worth developing a full interface to demonstrate the operation of the software since this would only be useful for demo 1 and it would be just as effective to display the results of the software in a simple commandline format.

1.2. Coordination of progress

We coordinated our work towards these goals primarily through the use of GitHub issues and GitHub projects, as together these allow us to specify start and finish dates for each task, as well as assigning them to individuals so we know who is responsible for getting a particular task done.

To ensure quality of code we decided upon a policy of implementing changes in branches and submitting pull requests for other members of the team to review and approve.

1.3. Allocation of time

We divided our group into two subteams. The software team consists of Florian, Nikita, Nikodem and Poppy; and for this demo focused on the development and implementation of the speech recognition and text to braille translation, as well as the setup of the Raspberry Pi. The hardware team consists of Balint, Daniel, Ripley and Souparna; and for this demo focused on developing a braille cell mechanism and prototyping each mechanism idea. The project manager, Ol, focused mainly on project management tasks such as planning meeting agendas and helping decide on priorities for tasks, but also spent time researching whether a new ethics approval process that will be required for user studies.

Table 1 contains the estimated time spent on each category of task by each member of the team. Unfortunately we hadn't started keeping track of this data until we were aware it was required, so some of the figures are rough estimates.

1.4. Budget and technician time

So far we have spent £10.27 on materials for a box which will be used for our prototype for demo 2 and beyond. We have used 30 minutes of technician time for manufacturing the box, and 15 minutes preparing for an unsuccessful 3d print. This gives us a total of £10.27 and 45 minutes of technician time spent.

1.5. Outlook

At this stage, we do not feel the need to make any changes to our plan for demo 2.

Category / Person	Balint	Daniel	Florian	Nikita	Nikodem	Poppy	Ripley	Souparna	Ol
Whole team meetings	5	5	6	5	6	4	6	5	6
Software subteam meetings	0	0	2	2	2	2	0	0	2
Hardware subteam meetings	2	2	0	0	0	0	2	2	2
Workshops	3	3	2	3	3	0	6	4	4
Project management	0	0	0	0	0	0	0	1	16
Pitch	3	3	3	3	3	3	3	3	3
Plan	3	3	3	3	3	3	3	3	4
Market research	0	0	4	0	0	0	0	0	0
Prep for user testing	0	0	0	0	0	0	0	0	3
Hardware R&D	10	10	0	0	0	0	10	20	1
Hardware implementation	55	35	0	0	0	0	50	35	5
Raspberry Pi setup	0	0	0	3	15	0	0	10	0
Speech to text R&D	0	0	2	2	2	0	0	0	0
Speech to text implementation	0	0	0	4	2	0	0	0	0
Text to braille R&D	0	0	0	4	5	12	0	0	0
Text to braille implementation	0	0	0	4	10	8	0	0	0
Software component integration	0	0	0	2	1	0	0	0	0
Software testing	0	0	1	1	3	4	0	0	0
Hardware testing	1	0	0	0	0	0	1	0	0
Demo 1 presentation	3	3	2	3	3	0	3	3	4
Total Hours	85	64	25	39	58	36	84	86	50

Table 1. Estimated hours spent by each team member in each area of the project.

2. Quantitative analysis and testing

2.1. Hardware testing

We tested the complete hardware system, consisting of the rail, moved by a large EV3 motor, the carriage which houses a medium EV3 motor to move the actuating cog in contact with the disk of the braille cell, and a DC motor which turns the actuating cog, changing the current braille being displayed on the cell. The results of this testing are shown in table 2.

Test	Time(seconds)	Errors	Success
1	20.1	0	√
2	20.5	0	√
3	20.9	0	√
4	19.7	0	√
5	20.3	0	√
6	20.5	0	√
7	20.3	0	√
8	19.9	0	√
9	20.4	0	√
10	20.0	0	√
11	20.2	0	√
12	20.5	0	√
13	21.4	1	×
14	20.6	0	√
15	20.8	0	√
16	21.3	1	×
17	20.9	0	√
18	21.7	1	×
19	21.6	1	×
20	1:50	1	×

Table 2. Results for 20 tests of the hardware.

The test considered the accuracy of the carriage, that is, its ability to hit its mark and line up with the desired braille cell, the time it took for the carriage to move to the cell and back into its original position, and finally, the structural robustness of the individual components, whether any parts came loose or cogs that perhaps skipped. Failures regarding the accuracy and the robustness would be considered a failure of the test and recorded as an additional error point on the table.

For this test, the carriage was placed in position 0 and then moved to the current highest position, 10, to simulate the travel time of the worst-case scenario. Once the carriage was there, the cell actuation took place and the carriage was returned to position 0 for the timer to be stopped. This was repeated 20 times for reliability and several observations were made:

- The large Lego motor was increasingly misaligning the carriage after 13 tests
- The misalignment caused delays to the carriage being returned to it's original position

The error rate for the first 15 tests was 1/15, 6.7%. However, the error rate for the last 5 tests was 4/5, 80%. This is a clear indication that in a more developed system with multiple braille cells, it's not viable to use such imprecise motors.

2.2. Software Testing

We undertook a thorough examination of the accuracy of speech processing, employing VOSK[1] – an open-source speech recognition toolkit. Our decision to use this toolkit stemmed from its lightweight nature, and providing models

[1] https://alphacephei.com/vosk/

Expected Word	Misclassifications	Misclassification Rate	Average Confidence Rate
his	15/40	0.375	0.81
is	37/40	0.925	0.61
as	4/40	0.1	0.91
apple	2/40	0.05	0.93
red	40/40	1.0	0.8
green	4/40	0.1	0.94
alphabet	1/40	0.025	0.98
school	14/40	0.35	0.86
braille	20/40	0.5	0.88
ostentatious	10/40	0.25	0.93

Table 3. Misclassification Rate and Average Confidence Rate

weighing in at 40MB. We have decided to go with the lightest model, and test the performance. This evaluation will also serve a dual purpose, for when we compare the performance against other boards such as the Raspberry Pi 5.

The testing protocol involved each team member enunciating 10 words, repeated 10 times. VOSK's transcription provided additional metadata, including confidence labels indicating the model's certainty in its translations. Making use of our team's cultural diversity, we also assessed the toolkit's performance across various accents. The selected words were chosen based on their relevance to a child's journey in learning braille.

The average confidence rate and misclassification rate for each word are depicted in table 3, the former being summarised in figure 1. Notably, VOSK struggled with the words 'red' and 'is': exhibiting lower confidence levels in transcribing them. This observation aligns with the inherent complexities of these words, which bear resemblance to similar-sounding terms ('read' and 'his').

Despite these findings, we have opted to explore alternative options for Demo 2, such as the Raspberry Pi 5, paired with a more robust model. We intend to replicate our tests on the Pi 5 to facilitate a comprehensive comparison. Additionally, we will be measuring the duration of processing in our evaluation.

In terms of relevance, this test serves to gauge the efficacy of the chosen speech recognition toolkit within the project context, informing decisions regarding platform selection for subsequent phases. The results obtained provide valuable insights into the toolkit's performance across various accents and linguistic nuances, aiding in the refinement of our system.

Interpreting the results, the difficulties encountered with certain words underscore the necessity for robustness in speech recognition systems. It may be necessary to explore incorporating features such as spelling out words or prompting users to confirm the intended word, potentially leveraging machine learning techniques. Nevertheless, this test has equipped us with the foresight to anticipate and address challenges that may arise in the future.

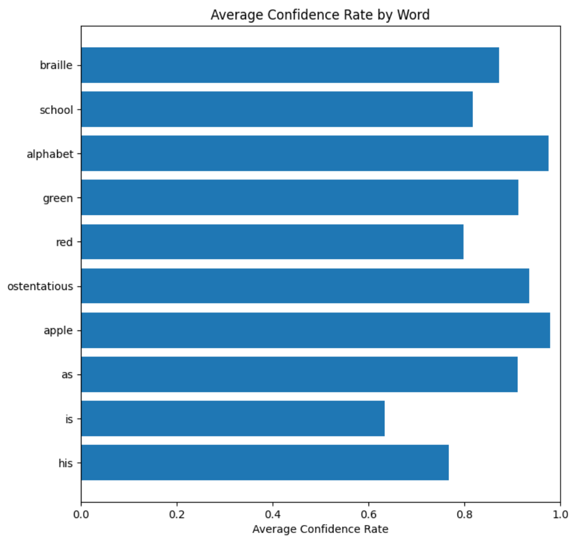

Figure 1. Bar chart of the average confidence label that VOSK assigned to the words

3. Budget

The total estimated budget of the system we presented in demo 1 is £374.36. This encompasses £337.76 for the EV3 and motors, £30 for the Raspberry Pi and about £7 for about 500g of Lego. We used the EV3 kit at this stage for ease of experimentation but by cutting it out we can control cheaper motors from the Raspberry Pi. This will significantly cut our product's cost down from demo 2 and beyond.

4. Miscellaneous

We have been planning to carry out user studies, but due to our product being disability-related, we may need to apply for additional ethics approval. We are currently awaiting a response from the ethics panel as to whether our study is covered by the existing approval.