Braille translator

Team 48
Ashmita Chatterjee (ashmita2), Matthew Price (mjprice2), Aayush Raj (aayushr2)
TA: Shuai Tang
Professor: Rakesh Kumar

1 Introduction

1.1 Objective

Being visually impaired in a world where the majority of tasks are completed by visually perceiving different objects, can often hamper the day to day functioning of the visually impaired. Technology has made great strides to make their lives easier through the use of products such as braille electronic note-pads, braille watches, etc. However, there is still a long way to go for technology to help them feel more normalized in public settings where they require to read menus, books, sheets of paper, etc.

In public places like libraries, restaurants or even grocery stores, the visually impaired face difficulties if there aren't braille versions available for these texts. Braille is a tactile writing system used by people who are visually impaired. Most public use libraries have a very limited selection of braille texts for users (highlighted in the article linked below [1]) and this limits people's potential and causes a dependency on other people. Since braille menus are very rare (as highlighted in this article linked below [2]), they often feel like they have to depend on others to order food or read a novel. In an attempt to empower the visually impaired further, we are building a hand-held real-time braille translator device which will give back some of their freedom by allowing them to read menus, sheets with normal text simply using this device.

1.2 Background

We plan to create a device that will be able to read text that visually impaired people can't and then produce that same text in a format that is perceivable to them. We plan to do this by having a scanner at the bottom of the rectangular device that will be able to scan any English text when placed on a surface with some legible text printed on it. This machine will then process the image data and produce a braille print on the braille display made using servo motors placed at the top of the device. This device should be controllable using buttons placed on the side of the braille display.

This device is different from other products available on the market such as the optacon because it has certain key components that make it a much more user-friendly product than some other devices. The optacon, for example, is a more bulky product and requires the user to move a tiny scanner across the page with one hand while the "read" vibrations produced on a big machine from the other hand. Our device takes a better approach to this problem by using micro servo motors that have a very small footprint and allow the device to be portable without making it too heavy.

Considering the use cases mentioned above, this device addresses certain key issues users face in scenarios like reading books at the library and menus at restaurants. The portability of a device for such uses is important to users as has been correctly identified by our colleague Abhijoy Nandi in his research when building the concept design for Samanaya[3], which is the basis for our design. Another interesting issue that this device addresses is that users are presented with a format for a text that is a lot more familiar to them - braille.

1.3 High Level Requirements

- The device should be able to scan a selected section of text in under 0.5 seconds.
- The device should be able to interpret relevant text with an accuracy of 75%. (interpret relevant text refers to the device being able to scan given text and convert it to a string of characters in the correct order)
- The device should be able to display scanned text in the correct order using a refreshable braille display with an accuracy of 90%.

1.4 Differences

While we are still using the same topic as our first project and we believe that after going through the design process once, we have found better ways to help us create a solution to a similar problem. We have made multiple design changes that will make this project significantly different from the last one. The first difference is that instead of using a camera in order to take a picture of the text, we are going to switch to a scanner. The main issue we had with the camera module was that the focal lengths of cameras in our price range were too large. This was going to make us have to put the camera as far as thirty centimeters away. With the scanner, we will be able to put it much closer to the paper, making the entire device smaller and handheld. Another difference is using the Raspberry Pi for the OCR. This will ensure that we have enough computing power for the task, also it will get rid of us potentially having to use an extra SD card to help support the microcontroller. Another big difference is switching to servo motors from solenoids. In our last project, we decided to use solenoids in order to keep the cost down, but this led to a huge problem with supplying enough power for a long enough period of time as each solenoid consumes a lot of power to push up a braille tip. Switching to servo motors can help with the power consumption and the number of characters we can display at a time. We believe that changing these design decisions has changed our project for the better and that it is worthy of redesigning.

2 Design

2 Physical Design & Block Diagram

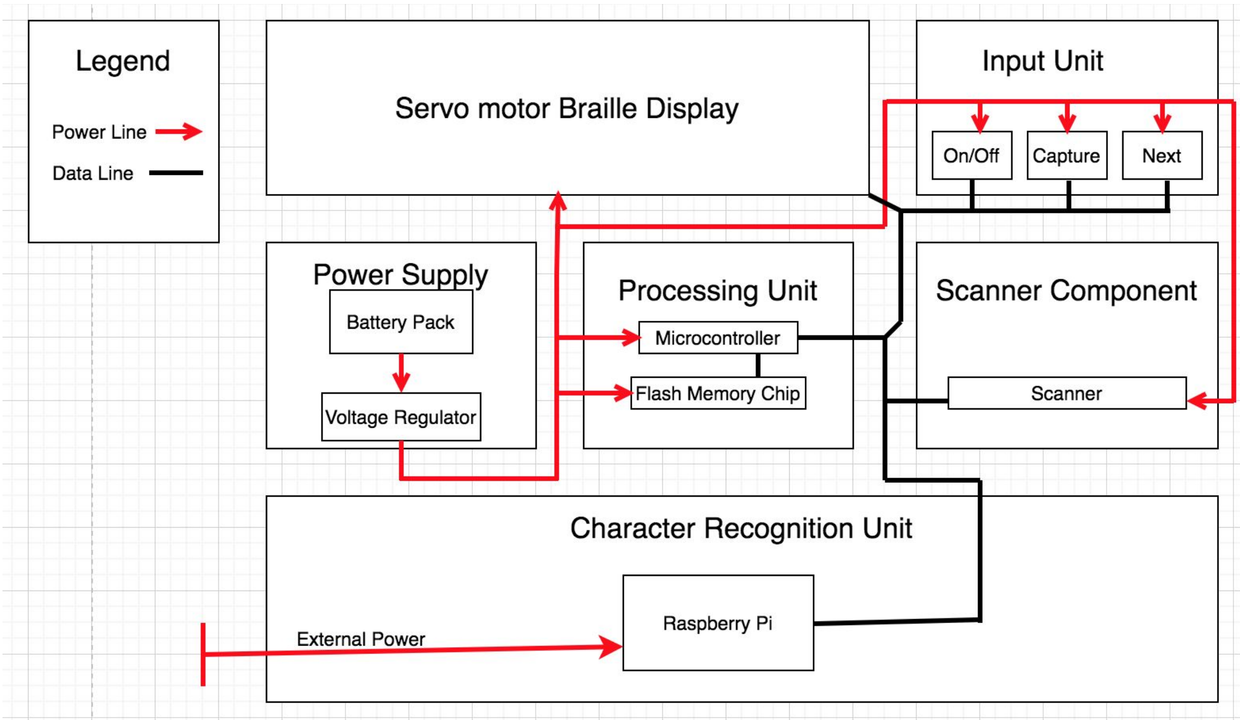

Figure 1: Block diagram for project

Figure 2: Concept design showcasing restaurant use [3]

2 Functional Overview

The functionality of the device is as described below:

- The device is powered on using the power button.
- The microcontroller receives various inputs from the input component like the power on/off button press, text capture button press, and the next button press. Once power on button press is received, the microcontroller will activate the scanner and the raspberry pi for any bootup procedures needed.
- The device can be placed over any surface with readable text and the capture button is pressed to allow the scanner to scan the text underneath the device.
- On receiving a capture button press, the microcontroller will send signals to the scanner to capture a block of text and will wait for the scanner to finish processing.
- Once the scanner returns an image, the microcontroller will send image data to the raspberry pi connected as part of the character recognition component for further processing and will wait for a return output.
- Once the character recognition component returns a string of characters to the microcontroller, the microcontroller will convert each character to its braille equivalent and send part of the selected text (up to 6 characters) to the braille display. The microcontroller will then wait for more inputs from the user.
- The braille display unit receives a digital signal from the microcontroller containing the string of characters of the deciphered text in braille. The digital signal is then supplied to the appropriate servo motors which are pushed up to form braille characters.
- On pressing the next button, the microcontroller will send the next set of characters to the braille display and will continue to wait for inputs.
- On receiving the power off button press, the microcontroller will power down the scanner, the raspberry pi, and all the other components.

2.1 Scanner Component

This component will be responsible for scanning displayed text on which the device is placed. This component should scan text and produce an image as an output that will then be sent over to the microcontroller component for further processing. We are using a scanner because we simply want the digital format of a 2D previously captured image and so a camera would be an overkill. We plan on using a 2D hand-held scanner that will be connected to the Raspberry pi via USB. A typical 2D-hand-held scanner should be around ~$35 which should be fine in our total budget (link to scanner : [6]). The power consumption of typical scanner is as follows:

12.0 W Operating
5.5 W Ready Mode
2.9 W Sleep Mode
0.5 W Power off

As you can see powering a 12 W scanner will be possible with any 12W wall power or adapter.

Our device should be able to scan a small block of text (~32 characters on a single line) within a reasonable amount of time (0.5 seconds) so that the delay between user input (capture button press) and device output is minimized.

2.2 Control Component

This component will be responsible for the correct operation of the entire device. It will handle the state machine for the entire device and ensure that each button press is handled correctly. The control component will mainly comprise of the microcontroller unit and the flash memory storage unit.

2.2.1 Microcontroller Component

The microcontroller will activate the scanner and raspberry pi on booting via USB and after it has detected a button capture the microcontroller will then send over the captured image to the Raspberry Pi for processing and deciphering the text in the image. Once the raspberry pi has deciphered the text using it will send a string of characters to the microcontroller which will then convert it into a digital signal and with the use of the servo motors the string of characters will be displayed in braille. The microcontroller that we have decided to use will be the 32 bit STM32F427AIH6 and will communicate with the scanner component via USB.

2.2.2 Flash Memory Component

The flash memory component will comprise of a SD storage card that will be able to store the text in the captured image in a buffer as we will not be able to display all of the deciphered text in the braille display unit at once due to budget limitations. To overcome this difficulty we decided to display only 4 characters and store the remaining characters in a memory buffer which the user can swipe through using the next button. We will be using a 256kB program storage however requirements may change if we require more storage.

2.3 Character Recognition Component

This component is responsible for converting an image to a string of characters. This component should include a raspberry pi connected to the microcontroller. This component will take an image as an input and convert it to a string of characters and return that string back to the microcontroller.

The exact technology we plan on using is OCR (Optical Character Recognition) on a Raspberry pi which can be used via the Tesseract OCR engine on the pi. The ability to recognize the full text in an image is what OCR does. We will be connecting the scanner to the input of the pi and the pi with the already downloaded Tesseract OCR engine should be successfully able to interpret the text in the image captured by the scanner.

2.4 Display Component

This component will be responsible for displaying the braille characters for the user to be able to read in real-time. It will be able to take the control signals from the microcontroller and turn those into the correct characters. We will be using servo motors that can be switched on easily, which will be able to push the characters up and down, creating the braille characters.

2.4.1 Braille Display Plate

This part will be where the motors are attached together. For each individual character, we will need six different motors, because each character is made up of six dots. We will also use a 3D printed armature, in order to assure that the dots are close enough together for someone to recognize them as a single character. The size of each motor will be the constraint for the number of characters we wish to display because we want the device to be handheld, so we want it to be small enough for a person to carry around.

2.4.2 Status Indicator

We will use one extra servo motor to push up when the device is powered on, and to be down when it is powered off. This is necessary, because the visually impaired users would not be able to see a light or any other indicator that could signal it being powered on.

2.5 Input Component

This component is responsible for correctly passing user input to the microcontroller. It consists of three buttons, “power on/off” button, “image capture” button and “next characters” button. For each button press, the input component is expected to pass a signal to the microcontroller that is then perceived as an interrupt by the system and is handled by the microcontroller.

2.5.1 Power Button

This button is necessary to toggle the power of our device. This will allow us to not drain the battery pack, considering it is a limited power supply.

2.5.2 Capture Button

This button will send a signal to the processor telling it to scan what is currently underneath the device, and then begin the process of converting the characters.

2.5.3 Next Characters Button

This button is necessary, because the amount of characters recognized by the device may end up being more that our device can display at once. If this is the case, the next character button will display the next characters that are left in the string.

2.6 Power Supply Component

This component is responsible for supplying the correct voltage to each of the different components of the design. The power supply needs to be able to supply different voltages to different components.

2.6.1 Battery Pack

Our device is to be designed as a portable device, making it easier for the users to bring it to the desired locations. This means that we are going to use a battery pack that can supply the necessary voltage. The motors will require up to 6 volts. We should use a 9 Volt standard battery to provide the necessary voltages.

2.6.2 Voltage Regulator

We will need a different voltage for the display component, the scanner, and then everything else. In order to ensure that each component is receiving the correct voltage, we will need three voltage rectifiers to take the input voltage from the batteries and turn it to the correct voltages for each component

2.7 Risk Analysis

There are some risks associated with the development of this device however the main few pressure points are the scanner, power supply unit and the servo motors. A few factors that come into play while deciding on a scanner for our device is the size of the scanner and it's compatibility with our selected microcontroller. If we are unable to find a scanner which is portable or small enough to fit in our hands then this device may be at a risk of being too bulky. The scanner should also be able to receive and transmit information to the microcontroller as this is crucial for the completion of our project.

We plan on using a battery pack to power our device however a few risks associated with that is the battery pack may not be able to supply enough power to the servo motors which should individually take up to 6 V, the scanner which should be operable at 9 V and the raspberry pi. We plan on mitigating these risks through trial and error.

3 Ethics and Safety

There are quite a few safety hazards that our project could potentially present. Our device might make use of a battery pack to power the device which is dangerous as batteries can leak. Leakage from batteries can cause the hazardous liquid to damage other electronic components present in the device and can also cause the battery to explode. Lithium-ion batteries are known to explode if overheated. Some other potential dangers are electrocution or burning from overheating of the battery or it's components. While it may not be possible to avoid these hazards completely, there are precautions that can be taken by us to possibly avoid some of these hazards. We can always make sure that the device is powered down before we make any changes in the circuitry and also remember not to touch any of the components of the device while testing as some of the components may be carrying more current than possible causing it to overheat and cause burns[4].

Our device is also meant to be used in public places like restaurants, libraries, grocery stores etc. One potential hazard of using a device in places like restaurants or grocery stores are spills. It is extremely easy to spill water or any other liquid in restaurants and if our device was exposed to such a spill there could be possible short circuitry of the device causing it to malfunction or even electrocute its user. To prevent such mishaps we could look into making some of the components of the device waterproof however that may increase the budget of the device. As a temporary fix we think that a waterproof case for the device may be enough to prevent damage[5].

Another potential hazard of this device as with any other electronic device, it should be kept out of reach for infants. Children using this device should be under adult supervision simply because children often misuse electronic devices as they are not aware of the precautions that one should take before operating such a device.

While there might be quite a few ethical issues with the project one major ethical issue could be with the copyrights to this project. There are other companies or organizations that strive to churn out products aimed towards the visually impaired and hence have already developed similar products. We aim to design and develop a unique product that will cater to the needs of the visually impaired while flaunting a new proposed design compared to pur previous project.

References

[1]https://www.nfb.org/sites/www.nfb.org/files/images/nfb/publications/bm/bm13/bm1303/bm130308.htm

[2]https://www.mic.com/articles/153739/how-american-restaurants-fail-to-accommodate-blind-diners

[3]https://www.abhijoynandidesigns.com/samanya

[4] Ieee.org, ‘IEEE Code of Ethics’, 2016. [Online]: http://www.ieee.org/about/corporate/governance/p7-8.html

[5] https://karacarrero.com/protect-laptop-water-spill-kids/

[6]https://www.ebay.com/itm/Konica-Minolta-Bizhub-C450-Optical-Scanner-Pwb-Ic-4037-0115-02/301343535976?hash=item4629797368:g:s~gAAOSw1x1UMtDg&fbclid=IwAR0T_c8JpqhRV8YuBi5gOTtQLqNVsOe306nweP6ba62Z5NNxDrsgfp8Zark