

E-Reader for the Visually Disabled

CS 580I Final Project Report

Nicholas Abate
Binghamton University

Tavish Srivastava
Binghamton University

ABSTRACT

Although E-Readers have increased the general population's access to texts, visually disabled users have very limited options to receive the same amount of access. We propose a novel solution to this problem by using mechanical solenoids to simulate a dynamic screen of brail text. In this project, we have created prototype hardware to represent dynamic brail, an android app that allows the user to select between a variety of texts to be displayed, and the required bluetooth configuration to connect the two.

KEYWORDS

E-Reader, brail, android, Arduino, solenoid

1 Introduction

Since the popularization of the Amazon Kindle in 2007, Digital E-Readers have become a widely used device. E-Readers use digital technology to allow users access to a vast variety of texts on a single device. Although these devices offer plenty of benefits to readers – such as increased availability of texts, customizable displays, and decreased storage space – there are few E-Readers that provide access to visually impaired readers.

We propose a novel solution to this limitation of E-Readers towards this demographic. Our proposed technology offers a digital display that can mimic the functionality of brail. We have created a prototype of this technology using mechanical solenoids that can dynamically present text to our user. With this solution, E-Readers can become accessible to a previously untapped demographic.

2 Software and Hardware Requirements

The hardware requirements are fairly simple; we need 6 solenoids to display one letter, one Arduino for logic, a Bluetooth module (compatible with the Arduino), NPN transistors, a button, and a 5v power supply.

In regards to software, we have separated our software architecture into three modules. This modularized approach to development allows for increased development speed. Our modules are as follows, Arduino logic and manipulation of solenoids, Android application for user control, and the Bluetooth connection between the prior components. Our

Arduino module requires the use of the Arduino IDE. This IDE allows us to code the logic and manipulation of the solenoids in C. When representing a letter, we first have to translate the letter to a list of six Boolean values (representing the brail letter). We then use the Arduino to manipulate the desired solenoids accordingly. Our Android Application is meant to simulate a virtual bookstore. We have three available texts for the user to choose from. Once the desired text is selected, the text data is sent to the Arduino via the Bluetooth module. Our Bluetooth module first creates the connection between the Android Application and the Arduino. After the connection is made, text data can be sent between the two hardware components.

3 Software Design and Architecture

We designed our software using a modularized architecture. We broke up our project into three main components and defined the interfaces between them prior to any development. This streamlined our development and allowed for a distributed workflow.

Figure 1 is a diagram of our system's software architecture. We have three components in our architecture. Each interface is predefined and outlines our system's control flow. The user experience consists of first selecting a text to be displayed on the hardware, then having that text physically displayed.

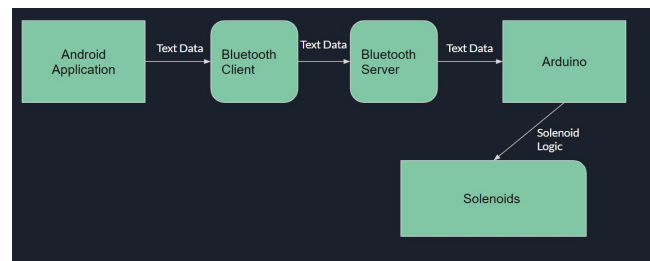


Figure 1: Architecture outlining software modules and interfaces

Our design of the hardware system is broken down into four main aspects. The first, the Arduino microcontroller. This microcontroller allows us to control the main logic behind our system; The logic being, when to activate a specific solenoid and when to leave a specific solenoid off. The next main aspect of our system is the solenoids. The microcontroller system is defined with six pins on the

Arduino (1,2,3,4,5,6). Each pin represents an individual braille “dot”. Figure 2:

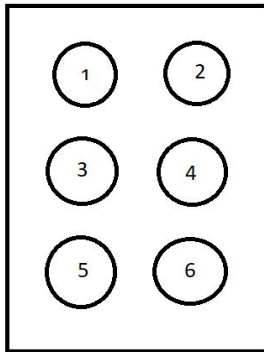


Figure2: Six motors representing six braille “dots”.

So, activating or deactivating certain solenoids decide which letter the system displays. One problem we faced was that the output of the digital pins of the microcontroller were not enough to power the solenoids. To overcome this, we used transistors. The transistors allowed us to use the logic from the Arduino digital pins and power the solenoids with another source of power.

The next aspect of our system is the button input. Due to financial restrictions, we were only able to build a system with a single letter. Therefore, we added a button input which allows a user to traverse through different letters, one at a time. The button is connected to the microcontroller. The microcontroller waits for a button input, and once it receives said input, the microcontroller outputs the appropriate letter. In addition, when the button is pressed again, the microcontroller outputs the next letter.

The final aspect of our system is the user API. For debugging purposes, we used our computer to send the user data to the system (serial connection) (user data-letters needed to be displayed). Our final system includes an android application which sends user data to the Bluetooth module. The Bluetooth module then passes the data onto the Arduino. Finally, the Arduino activates certain solenoids, and the letter is displayed.

4 Implementation

Android Application Development

We developed an interactive android application to allow the user to select which text will be displayed on the hardware. Due to our hardware limitations, we allow the user to select a word to be displayed on our hardware. The application takes the user’s selection and sends it to our Arduino using the Bluetooth module we created. This Bluetooth module first searches for the Arduino and creates the connection. It then allows text data to be sent from the Android application (the server) to the Arduino (the client).

Solenoid logic:

Our solenoid logic works as follows:

1. System will receive user input (Example: “TRY”).
2. If statements link a letter to an if block (figure3):

```

if(i==0) { //T
    digitalWrite(12,HIGH); //1
    digitalWrite(3,HIGH); //6
}
else if(i==1) { //R
    digitalWrite(11,HIGH); //2
    digitalWrite(3,HIGH); //6
}
else if(i==2) { //Y
    digitalWrite(13,HIGH); //3
}
    
```

Figure3: If blocks iterating through the letters T,R,Y.

3. The solenoids are appropriately activated/deactivated.

As discussed in the architecture section, we used another power supply to power each solenoid. We used a 5v supply to power each solenoid. As mentioned earlier, the Arduino provides the logic, and the transistor acts an amplifier-switch. Our solenoids together form a single letter (figure4):

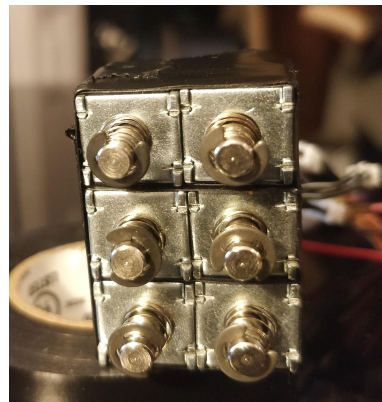


Figure4: Single braille letter.

Our circuit, consists of six solenoids, one microcontroller, one 5v power supply, six NPN transistors.

Next, a snippet of our circuit logic (Figure5):

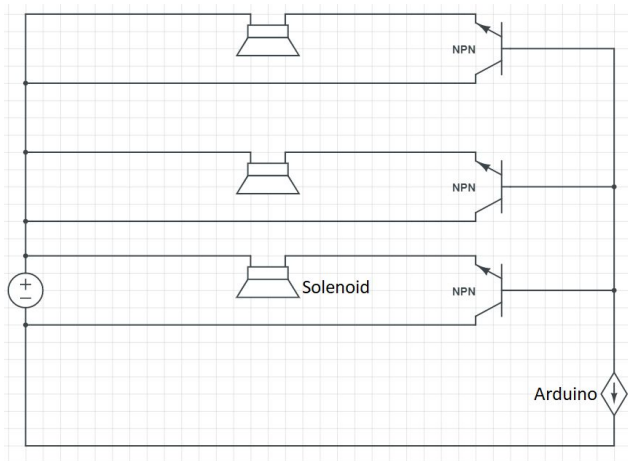


Figure5: Half of the circuit design (note that just connecting 3 more loops of tranistor-solenoid completes our design)

5 Results and Future Development

Our current application prototype is a limited version of the virtual bookstore we would like to create for our hardware. Due to the hardware restrictions, we are limited to the display of a single letter. This forced us to create custom texts that can be easily parsed while only viewing a single letter. In theory, our future hardware would be able to connect to any virtual book store and display the text accordingly. In regards to software, we would only need to update the availability of texts to the user.

Due to financial and time restrictions, we were only able to build a single letter. We added a button input as a compensation. Although the button achieves the purpose of our proof of concept, in the future we would like to see as much braille “dots” as there are on a regular braille book.

Although this was our original idea there are some, but extremely limited, similar implementations being worked on. A team from the Massachusetts Institute of Technology has excelled in their implementation. (1)

In a perfect world, where we are given enough time and resources, we would like to see a screen, 10- inches or larger, with the ability for each individual pixels in the screen to be “activated” or “deactivated”.

6 References

(1): McElvery.R, April 2017, MIT news feature, <https://www.technologyreview.com/2017/04/25/152177/mit-students-invent-simple-device-that-makes-printed-text-accessible-to-the-blind/>