Carrina Dong
September 11, 2016
ME290U, HW2

Text Entry Device: The Hold n' Press Board

Documentation @ Github: https://github.com/carrinacat/hw2_cdong_chord

Video: https://youtu.be/eLS12vg6Lk8

Text Entry Method

The text entry method I chose was chording. I chose this method because I believe it had the most potential to optimize both speed and accuracy. Before I started the design, I evaluated the other options we had learned in class. Compared to methods such as multi-tap and D-pad, chording is considerably faster. Though it is slower than T9, T9 can be error prone regardless of experience level. I believe an experienced user, given an ergonomic and user-friendly chording text entry interface, can achieve speed and high accuracy.

To make the user interface easier for the user to navigate, I implemented chords in a 6 x 5 matrix format. There are two sets of 5 buttons: press buttons and hold buttons. The 6 rows are comprised of the 5 hold button options (which are held down), as well as a 6th "no hold" row (when no hold buttons are being held down.). The 5 columns represent the press button options.

Each row-column/press-hold combination maps to a separate character – there are thus 30 possible characters mapped using this method. It is important to note that in this method of entry, the hold buttons are always held down first before a pressing press button.

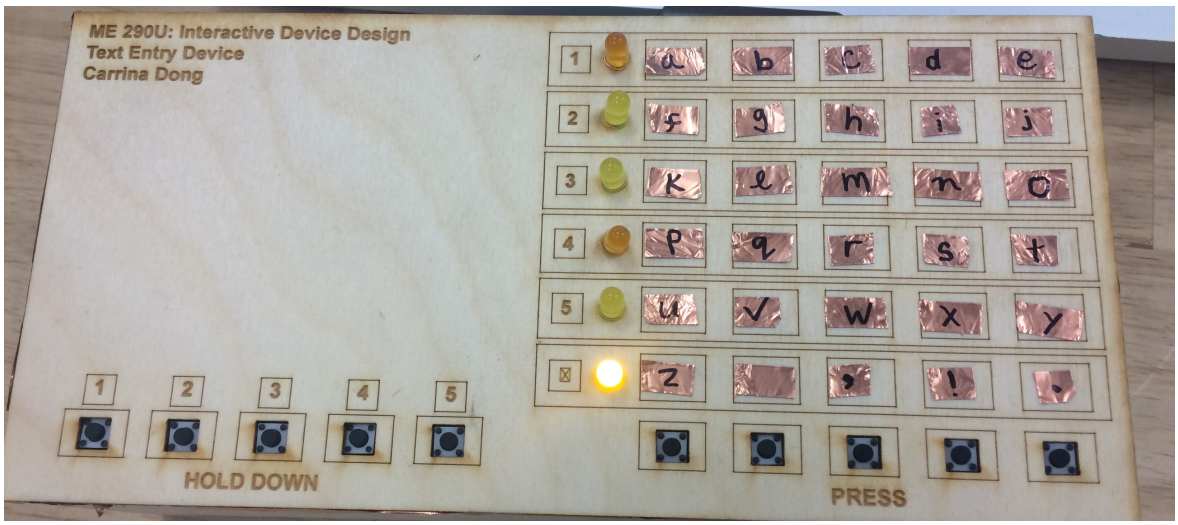


Figure 1: Board Layout

While the hold down buttons represent the rows of the matrix, they are laid out parallel to the press buttons. Ideally, the user would be able to rest both hands comfortably on the board during operation, as shown in Figure 2 below. It also facilitates better speed of text entry.

However, this layout can be confusing for the user for text entry. To help the user determine which row is active, LED indicators are mounted next to its corresponding row number. These light up whenever a hold button is held down. If no buttons are held down, then the “no hold” LED is by default on. The LED indicators also blink when a press button has been pressed.

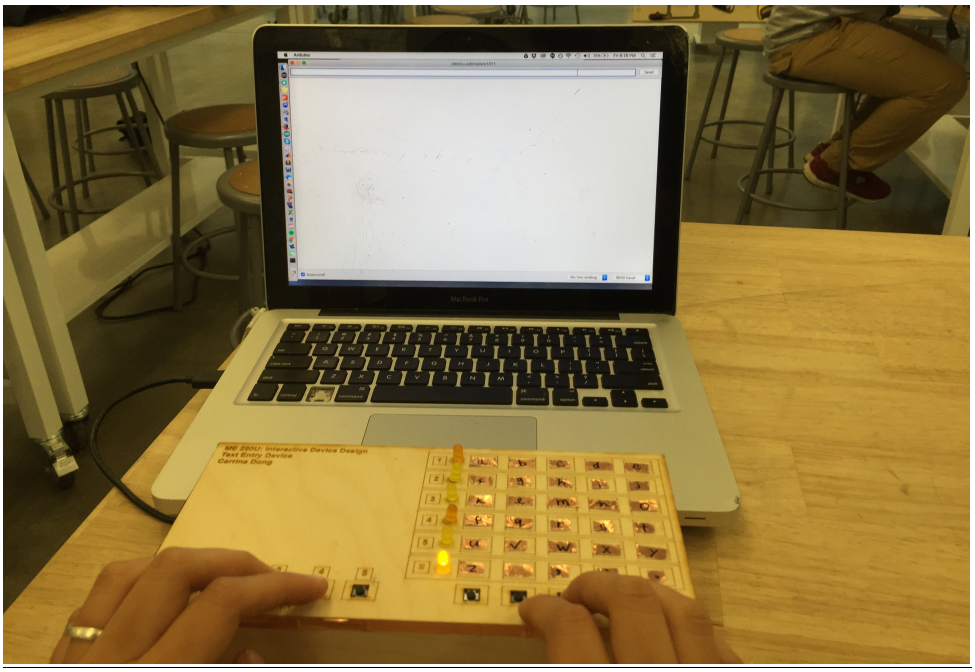


Figure 2: Hand layout during operation

To illustrate how this method works, these are the set of operations a user would use to type “Hi!” and the feedback the user would receive.

Operation	**User Action**	**Feedback Received**
Type “Hi”	Hold button #2	Row 6 LED turns off. Row 2 LED turns on.
	Press “h” button (3rd column)	Row 2 LED blinks. Serial Monitor prints “h.”
	Press “i” button (4th column)	Row 2 LED blinks. Serial Monitor prints “I.”
	Release button #2	Row 2 LED turns off. Row 6 LED turns back on.
Type “!”	Press “!” button (4th column)	Row 6 LED blinks.

Implementation

This text entry method was implemented through the Arduino IDE. If a hold button is pressed, it activates a corresponding while loop that checks and executes individual press cases. Conditional statements cover the press buttons with no hold button combinations. The pseudo-code is given below.

```
While hold button is pressed [x 5 for each hold button]
        Turn no hold LED off
        Turn corresponding row LED on
        Check all press button states [x 5 for each press button]
                If press button is pressed
                        Print character
                        Blink row LED
                        Delay 600 ms. (For blinking/debouncing.)
If press button is pressed [x5 for each press button]
        Print character
        Blink no hold LED
```

The full code is posted in the GitHub repository. The naming conventions and corresponding I/O pins are given in the tables below.

Hold Buttons		
Button #	**Variable**	**Pin**
1	B0	D0
2	B1	D1
3	B2	D2

4	B3	D3
5	B4	D4

Press Buttons		
Button #	**Variable**	**Pin**
6	B5	D5
7	B6	D6
8	B7	D7
9	B8	D8
10	B9	D9

LEDs		
Button#	**Variable**	**Pin**
1	LED_A	D10
2	LED_B	D11
3	LED_C	D12
4	LED_D	D13
5	LED_E	D14
[NO HOLD]	LED_F	D15

Wiring

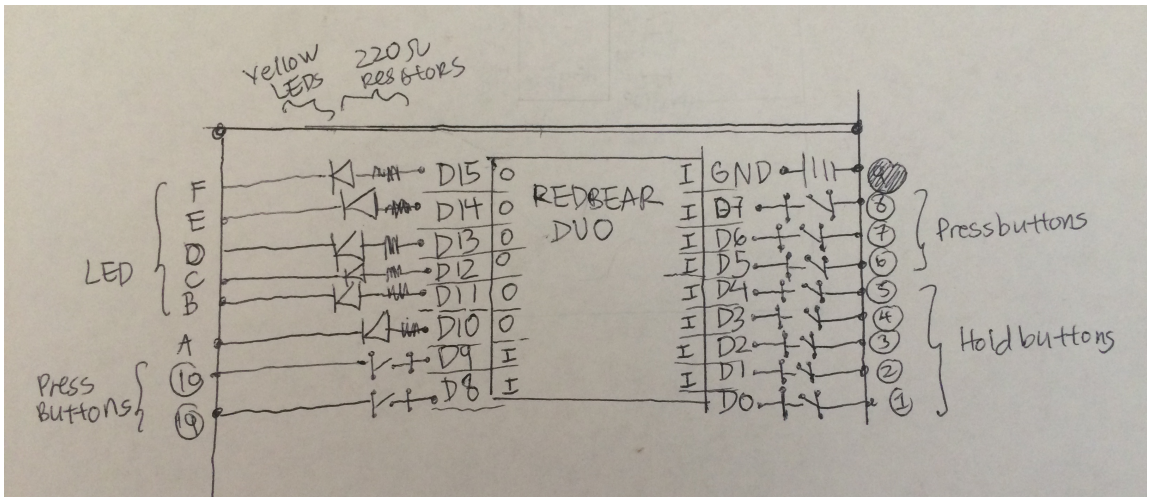


Figure 3: Wiring diagram: RedBear Perspective, input pins are in pull-up mode

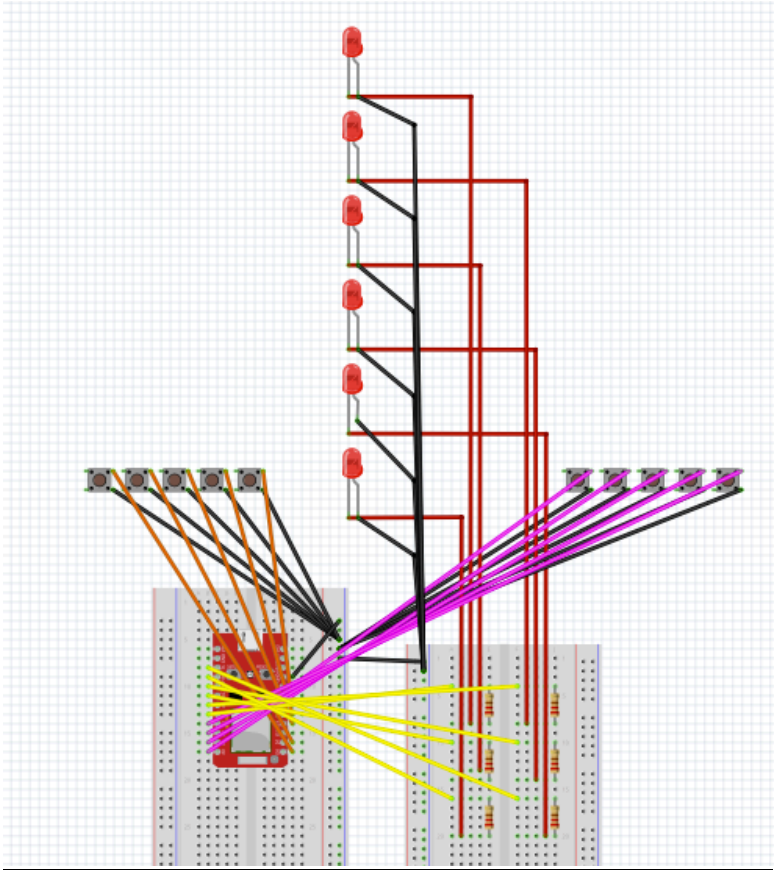
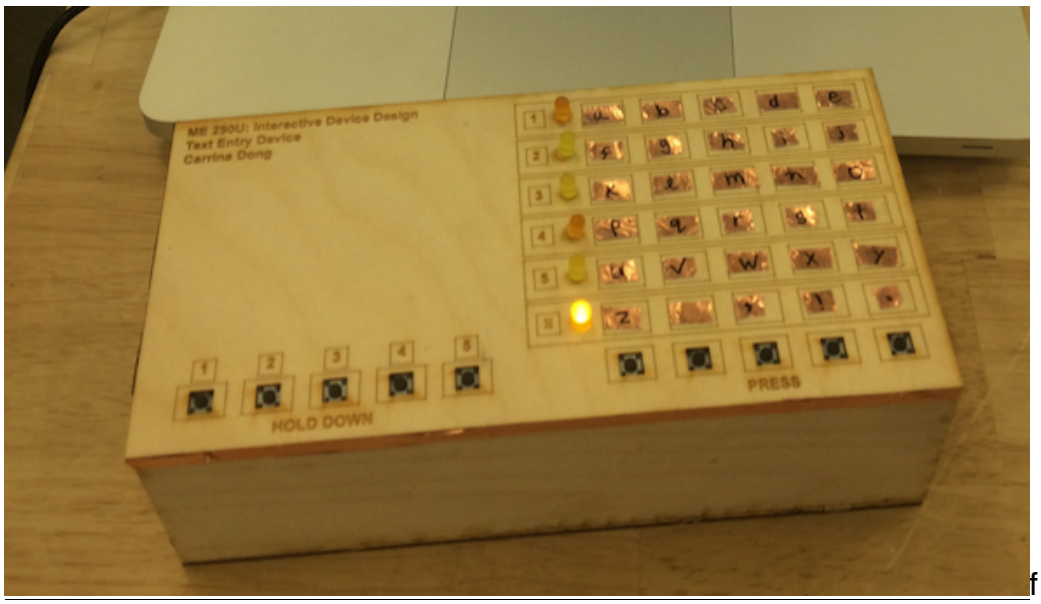


Figure 4: Breadboard

Fabrication & Assembly

The text entry method is mounted on a 8.75” x 4.25” x 2.6” enclosure. The top panel contains the switches, LED indicators and a character map. Wiring as well as the RedBear are contained within the enclosure. The back panel is open as a wiring exit (mainly for the USB cable.)

All pieces of the enclosure are made of 1/8” plywood, cut to size and engraved using the laser cutter. They were then joined together using adhesives. Copper tape stickers were used to label the character map – this was done for possibility of changing the layout of the characters in the future. For aesthetic purposes, copper tape was also used to line the edges of the box.



f

Figure 5: Enclosure of text entry device

Reflection

I learned many valuable lessons during this assignment. First, I learned the challenges of creating a text entry device. There are many trade-offs at play. For example, while fewer inputs were better for simplicity of fabrication and wiring, more inputs were potentially better for speed of text entry. I had to make decisions about which characteristics I valued in a text entry device, as it seemed difficult to “have it all.”

I also learned the importance of accounting for cable management and wiring. When I first built the dimensions of the enclosure, it was too small and was straining all the wires. Furthermore, I had to think about the amount of slack to use such that it wasn’t excessive but also was not constrained. If I could do this assignment again, I would think about this more carefully instead of after the fact. I would also maybe add strain relief.

Lastly, I learned the importance of making time for iterations – especially in designing human centered devices. I believe my prototype can be improved quite a bit in making the design more ergonomic and efficient, though additional time to test the design would be necessary. For example, I’d like to test the dimensions of the switch layouts for maximum long-term typing comfort, as well as the order of the characters for maximum text entry speed.

This assignment was a good of difficulty. It forced me to complete the required trainings for this class in order to complete the assignment. It also gave me a sense of the workload for this class and the time to complete tasks such as coding as well as wiring/soldering.