IDD HW2 - Text Entry Device

Joshua Kay

GitHub Repo: IDD - HW2 Text Entry Device

A text entry device was created using ten switches and the multi-tap technique with a re-designed layout, as shown in figure 1. The device was constructed by soldering the buttons to a PC board with copper through holes, using wire for the electrical connection. The keys and enclosure were engraved and cut by a laser cutter. The PC board was glued to the wooden enclosure top, and the keys were tapped to the tops of the buttons with standoffs for extra support.

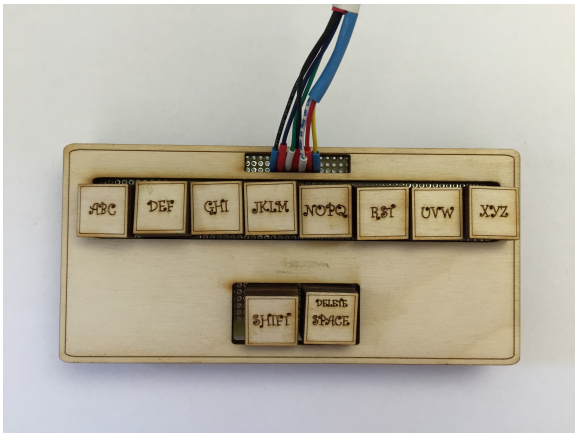

Fig. 1. Redesigned multi-tap text entry device

Each button in the top row of the text entry device represents either 3 or 4 letters, and similar to traditional multi-tap implementations, pressing a button cycles through the characters written on that button. However, in contrast to multi-tap used on mobile devices that are laid out in a grid to conform to a smaller form factor, the redesigned multi-tap text entry device allows for the user to touch all ten keys with all ten fingers at the same time. The result is a higher characters per second input rate than a traditional multi-tap keypad layout. Each key is marked with the characters that button cycles through. The two keys pressed with the two index fingers were chosen to contain 4 characters due to the increased dexterity of the index finger compared to the other fingers. Additionally, the two buttons operated by the user's thumbs were chosen to be a shift key and a space/ delete key. Shift was implemented to allow for more character options to be implemented. A large factor in text entry accuracy involves visual feedback and the ability to correct incorrect keystrokes. To provide visual feedback, a simple graphical user interface was designed in Python that allowed the user to see what character was inputted with each keystroke as shown in figure 2. Additionally, a delete key was added by shifting and then pressing the space key. This allowed for quick corrections of incorrect keystorkes, ultimately improving the user's speed. A demo of the working device used to type "Hello world" and "Device design" is shown here:

Text Entry Device Demo Video Link

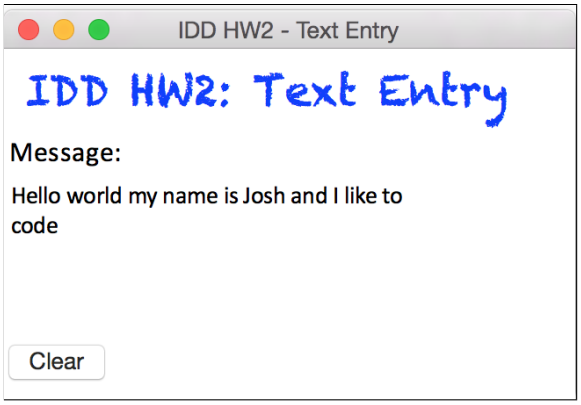

Fig. 2. Text entry GUI

Although the physical layout of the keys were arranged in two rows, the electrical connections of

each button were arranged in an array to minimize I/O pins used on the micrcontroller. The schematic for the text entry device is shown in figure 3. Each column pin was set to have an internal pullup. Next, a row was written low. If a column pin was then read low, the button at that index of row and column was pressed. The row pin was then written high, and the process was iterated for the next row pin.

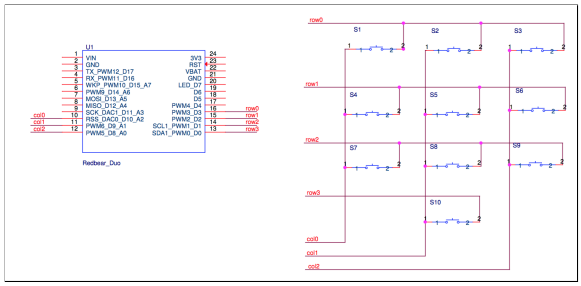

Fig. 3. Text entry schematic

To keep track of the character the user intended to input, an integer with the ASCII value for the intended character was stored in memory and sent over a serial connection to be converted into a character in the python GUI. Using an integer to keep track of the intended character allowed a simple algorithm to be developed involving the row and columns of the button as shown below:

$$i + 9\,row + 3\,column - shift$$

where i was initially set to 97, correlating with the ASCII value for "a" and shift was either 0 or 32 for uppercase or lowercase values. This character was only sent through the serial connection to a computer when a new button was pressed or when the user had not pressed the same button for a timeout of 2 seconds. Therefore, if a user moved to a new button, a new character was printed immediately. If the user wanted the next character to be one that shared a button with the previous character, the user needed to wait for 2 seconds before proceeding. A "-1" value was first sent over serial to indicate the python GUI to move to the next character. Then, the python code converted the integer value to the correct ASCII character. Delete was also implemented. When shift was pressed and then space was pressed, a "-2" was sent over serial, indicating a delete was pressed to the GUI.

By completing this project, I became more familiar with construction of enclosures using the laser cutter. I also learned how to make GUIs in python.