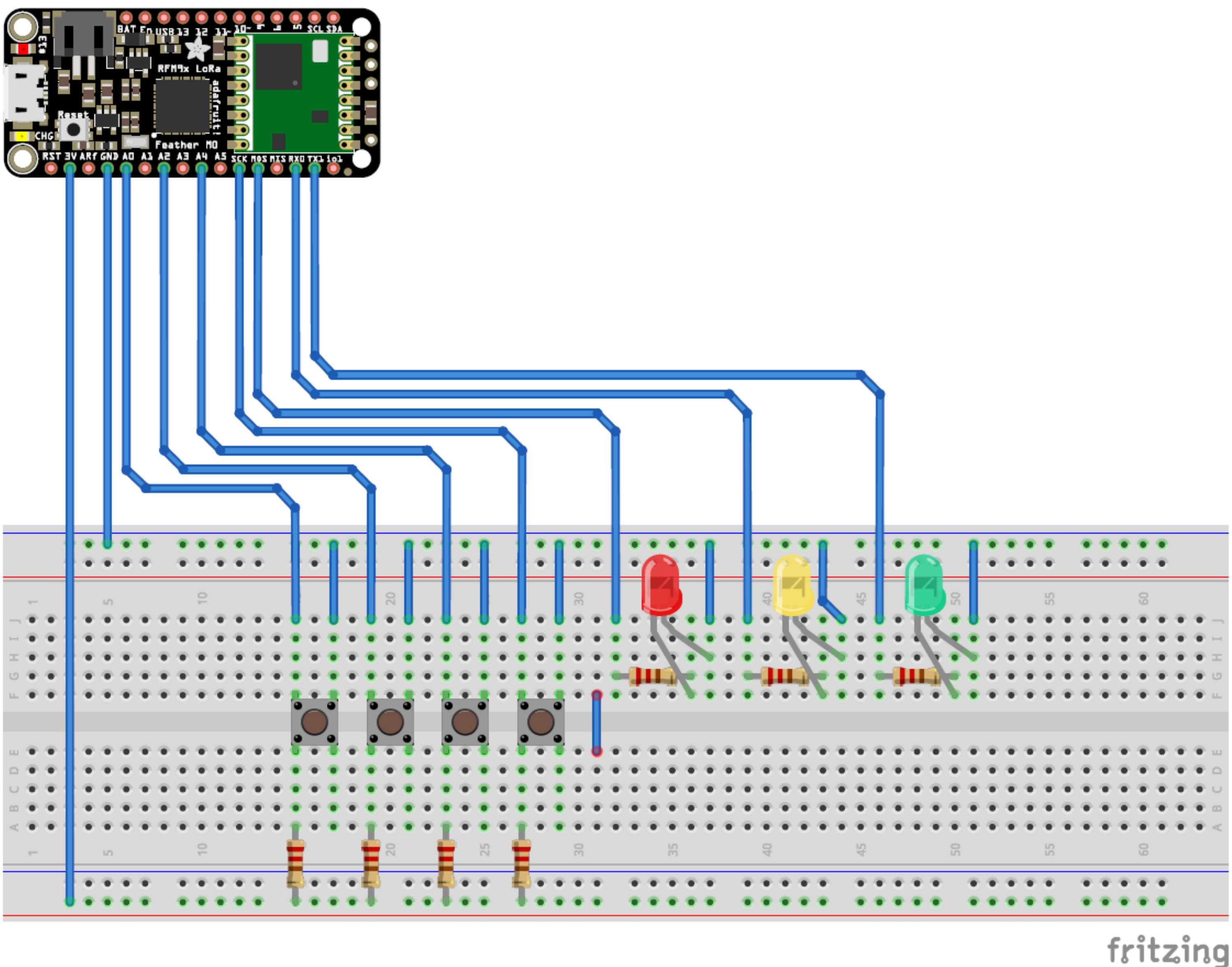
Interactive Device Design Summer 2018

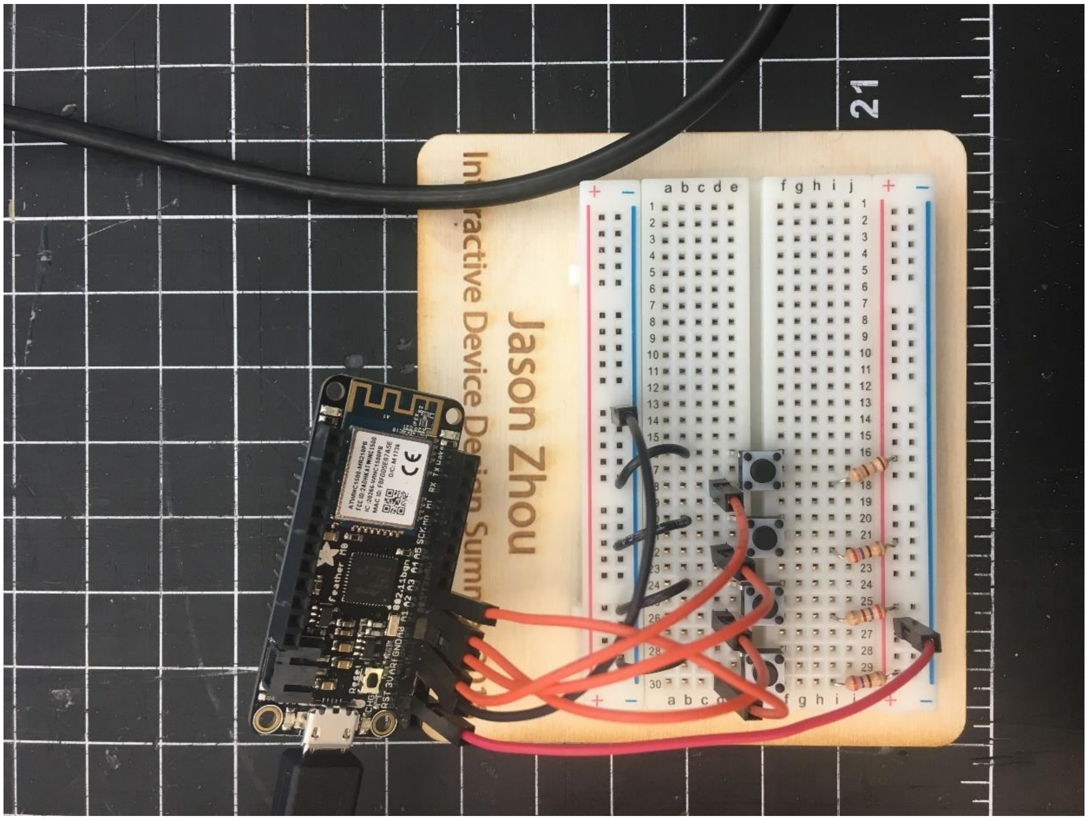
Homework 2: Text Entry Device

Jason Zhou & Cameron Riley

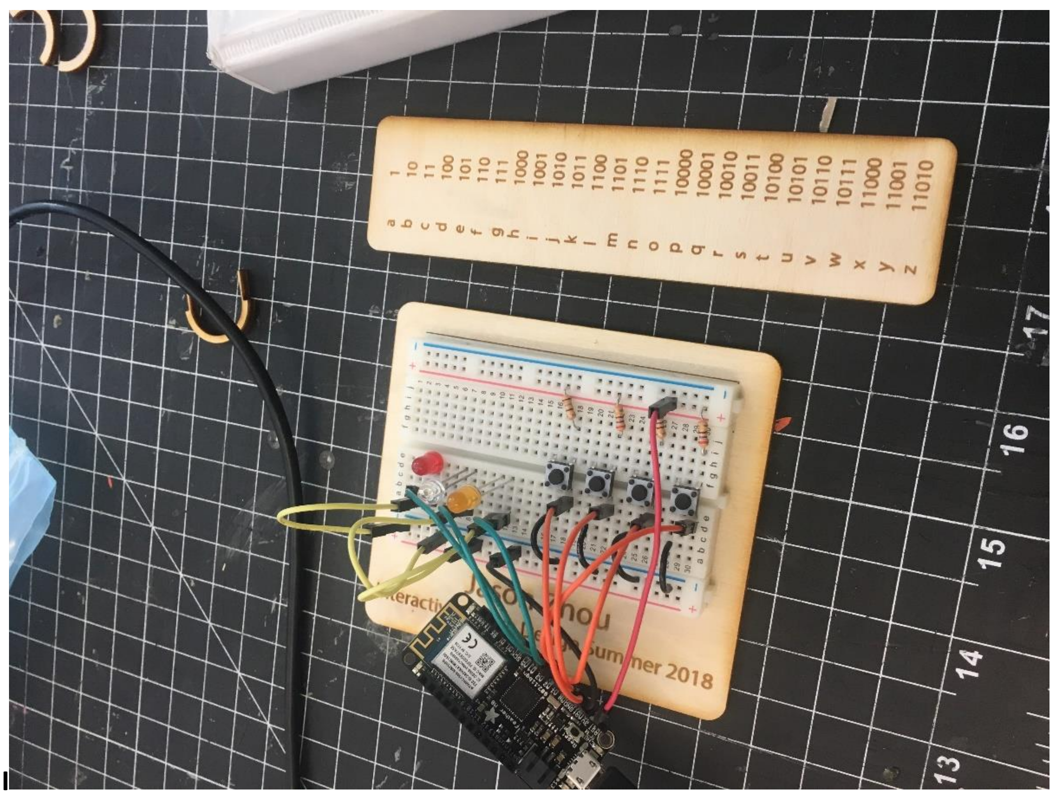
Wiring Diagram & Project Images:



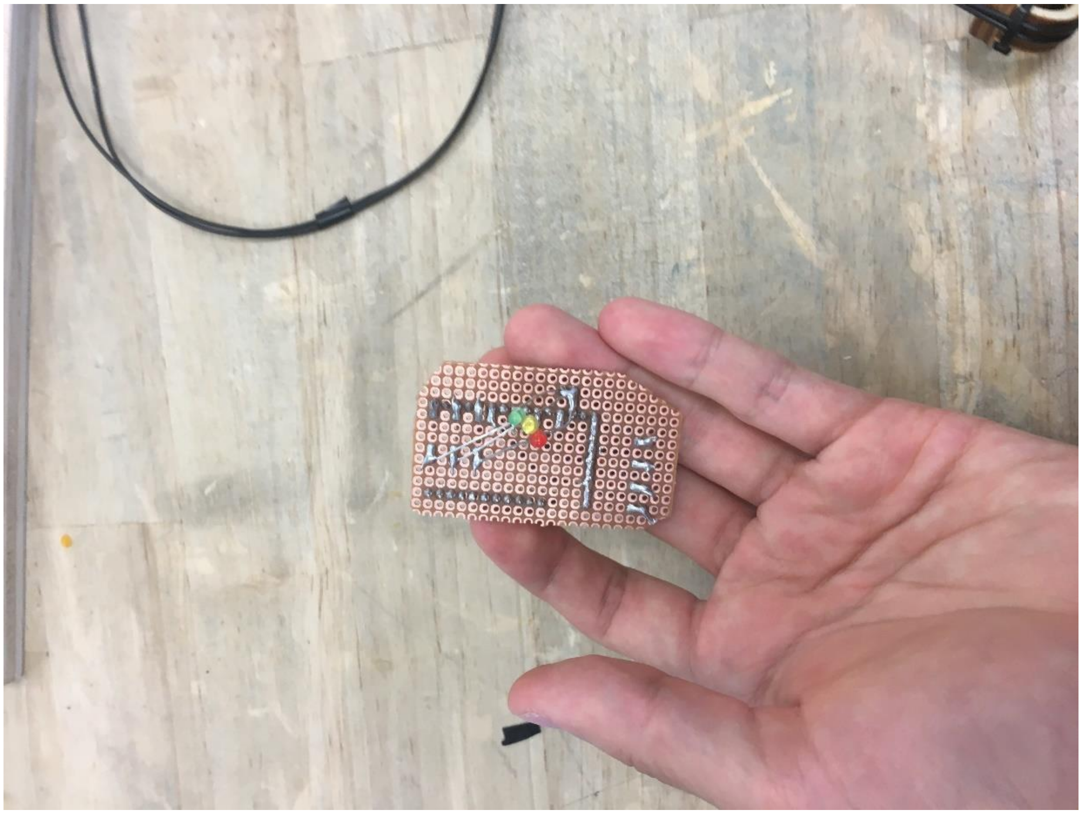
Initial breadboarding work of button mechanism:

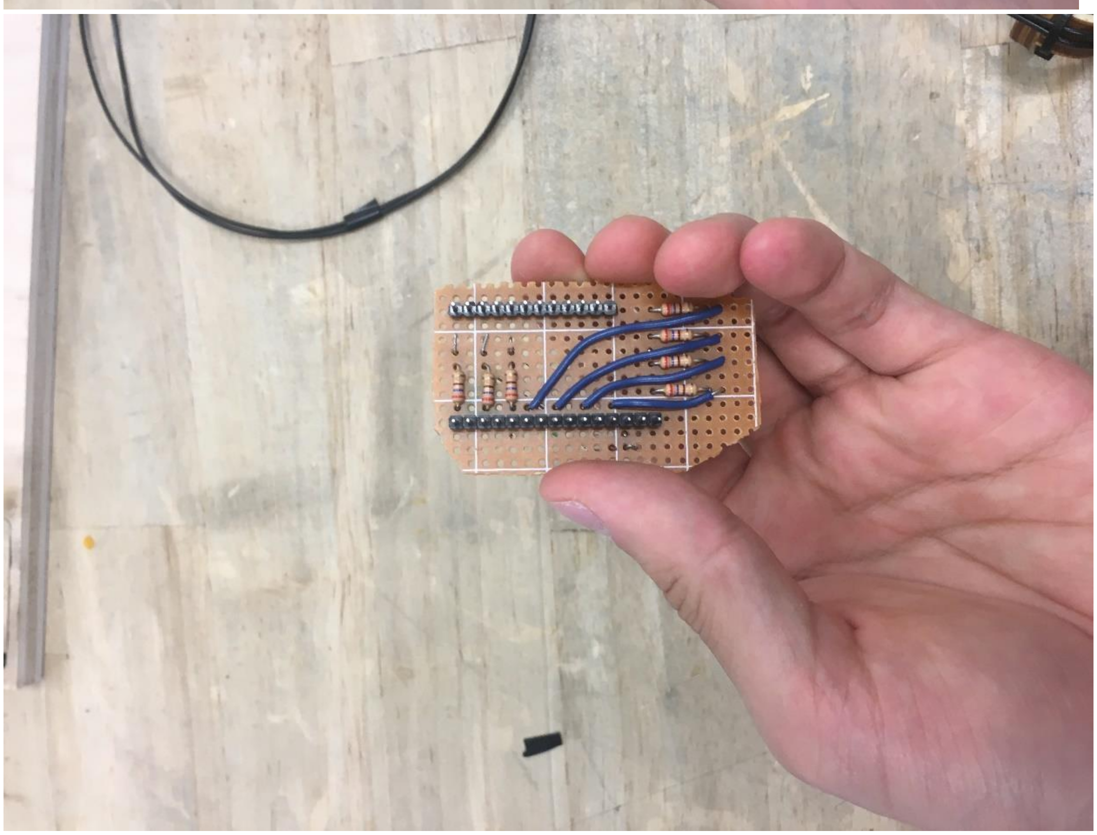


Final breadboarding work including instruction tablet and LED indicators:



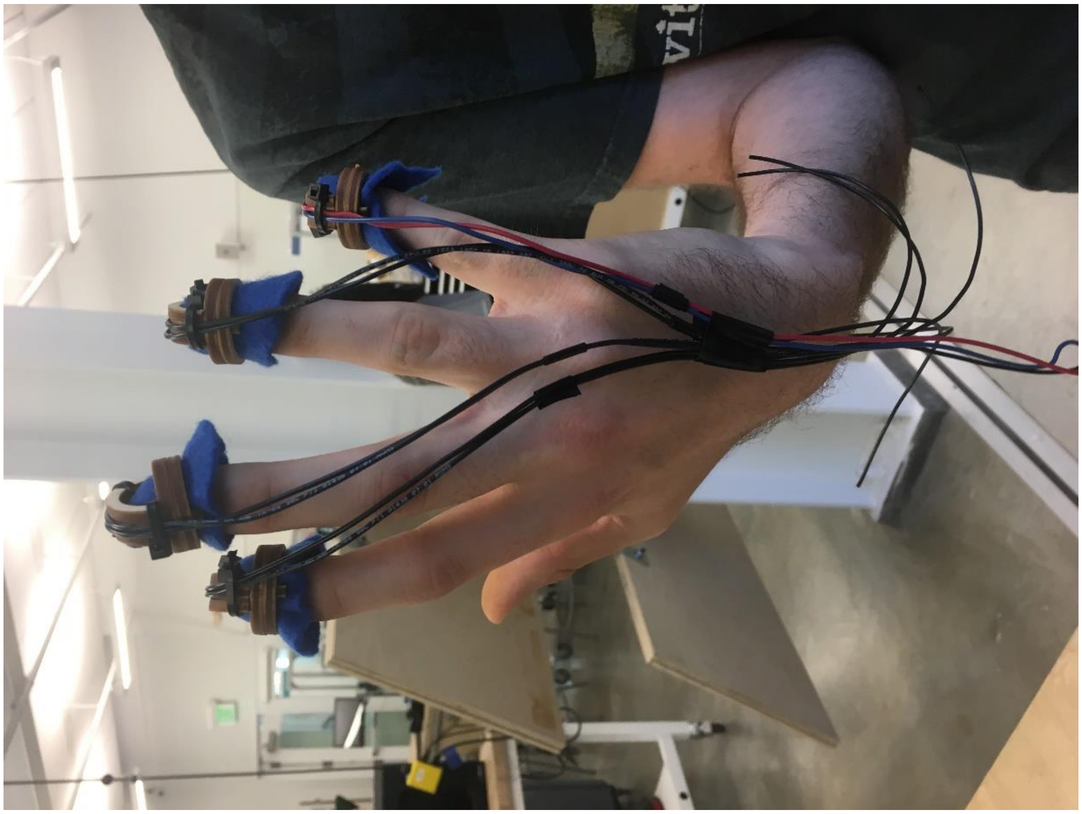
Front and back of perfboard circuit





Wrist strap work in progress:

Fingertip button mechanism work in progress:



Final Device Pictures:

Github URL:

https://github.com/IDD-su18/hw2-jasonxzhou

Writeup:

Our team decided to use a binary-to-letter text entry technique. This technique allows for the use of only four buttons: whitespace, enter, 1, and 0. This concept allows for a hand-interface device to be made to correspond to each button. The decision was made to use the thumb as a surface against each button was pressed. The four-button configuration was chosen because it could be operated with one hand, decreased the amount of Focus-Of-Attention (FOA) to one for expert users, and made efficient use of the allotted button count.

The device was constructed using the usual electronics: an Adafruit Feather microcontroller, wires, a perf board, colored, LEDs, 330 Ω resistors, four pin buttons, and solder material. These components were soldered onto the perf board using a bridge-pin configuration. Each button was wired with a pull-up resistor. The ergonomic parts that fit the fingers and forearm of the user were made using laser cut plywood that were modeled using SolidWorks and cut out along lines mapped in Adobe Illustrator. The finger cuffs were made by cutting out two ring parts with one central notched part and gluing them together (as in the pictures). Two wires were connected to each button to be affixed to each finger cuff along the back of the hand, curving over the front of each finger cuff. Fabric portions that were used for each finger were then stapled together and attached to the cuffs using hot glue. The wrist portion, including the box and wristband was glued together. The wooden chain was connected using cut paperclips through each of the holes and a rubber band through the end holes of each side of the chain. Finally, the Microcontroller and perf board were put inside the box and covered using the slide portion of the box.

For the electronics/code part: the code/project utilizes 7 microcontroller pins total, with 4 as input and 3 as output. The 4 input pins are connected using pull-up resistors to the 4 buttons that correspond to the user's finger. As each button is pressed, a change in voltage for that respective pin is detected. The index and middle finger buttons are mapped to the binary bits, the fourth finger is mapped to the enter/capslock key, and the fifth finger button is for entering a space.

The code initializes an integer variable called "key" to the value 0; this variable stores the letter to be inputted. Note that the value 0 does not correspond to any letter. The code now awaits input; if the index button is pressed, "key" is left-shifted by 1 bit (analogous to adding a 0 to the right of the current number). Similarly, if the middle button is pressed, "key" is left-shifted by 1 bit and incremented by 1 (analogous to adding a 1 to the right of the current number). The code will await user input until either the enter button is pressed, in which case the value "key" is converted to the corresponding letter (using a hardcoded translation function), or until the space button is pressed, in which case a space char is outputted and "key" is simply reset to 0. The capslock function is implemented by checking if "key" is 0 when the enter button is pressed (i.e. if a user simply presses enter without pressing any of the binary bit keys). Lastly, the code handles two types of input errors. If more than 5 bits are entered (even prior to the user pressing the enter key), the code will reset "key" to 0 and prompt the user to try again. If 5 bits are entered but the pattern does not match one of the hardcoded translations (as not all 5 bit numbers are used/mapped), then the code will prompt the user to try again as well.

The three LEDs (green, yellow, and red) indicate the device's state. The red LED toggles on/off according whether capslock is engaged or not; the green flashes to indicate a successful character entry, and the yellow flashes to indicate that an error has occurred.

Cameron: I learned that, through the prototyping process, it is much more efficient and uses less time to use existing materials (like gloves) in the design. Making the glove from final materials is more of a final step. Also, a difficult part of this design was coming up with a way to make the ergonomic portion of the device more universal (since each person has a different sized hand). I learned that narrowing down the end-user makes the process much simpler.

Jason: The biggest take-away I had from this project was the importance of planning ahead and accounting for unpredictable events. Going in to the project, I expected the code and the circuit to not be too time-consuming to construct, which it wasn't on the breadboard, but the process of translating the breadboard circuit to the perfboard took an enormous amount of time. The debugging process, the soldering process, and fitting everything into a small perfboard turned out to be much more challenging than expected. Thankfully, Cameron and I started early and were able to deal with the unexpected delay, but the process still served as an important reminder to plan even further ahead for future projects.

Link to video:

https://youtu.be/CGl3-Q0e-mg

Link to your code

https://github.com/IDD-su18/hw2-jasonxzhou/blob/master/hw2/hw2.ino