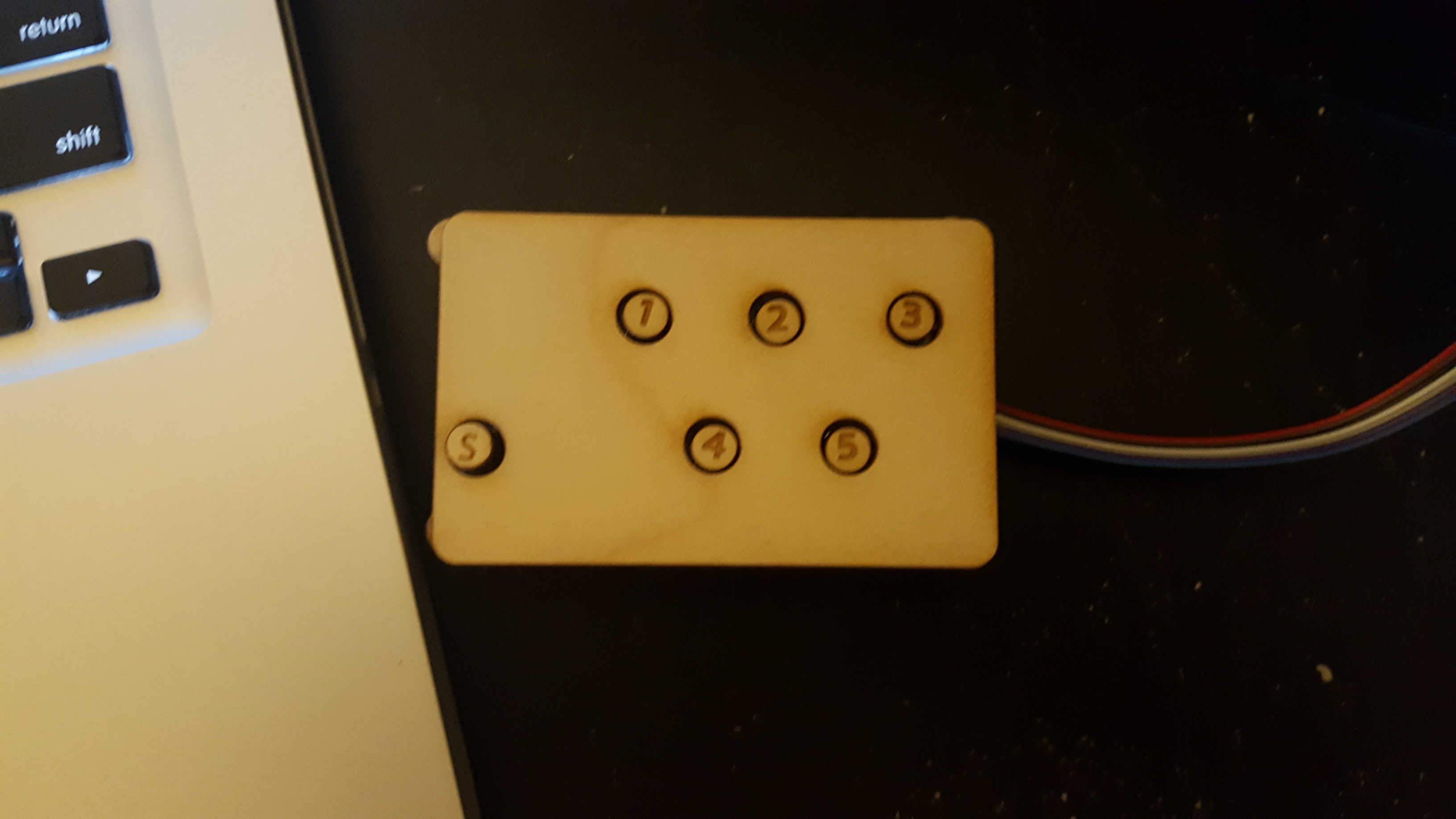
HW2: Text Entry Device

Tyler Chesebro

IDD Fall 2016

UCBerkeley

I created a 6-key text entry device. The motivation was to have 5 main keys and a “space” key. Letters are chosen through a combination of one or more of the main keys, numbered 1-5 and a “space” is a single tap of the “S” key. The idea behind the design of the key-board was to allow a user to type using only one hand. This fabrication lends itself to be used in the right hand, with the 5 main buttons near the fingers and the “space” key near the thumb.



**IMPLEMENTATION**

The unusual part of the device is how the characters are entered. The buttons act as binary digits, where the 1 corresponds to 1, the 2 to 2, 3 to 4, 4 to 8, 5 to 16, and “S” to 32. Through different combinations this allows for a total of 62 different characters to be entered by the board using 5 buttons at a time. 63 characters is technically possible, but in sticking with the singlehanded use of the board I did not want to include any combinations that would take 6 simultaneous button presses. Also the [000000] state does not contain a character as that is the rest state.

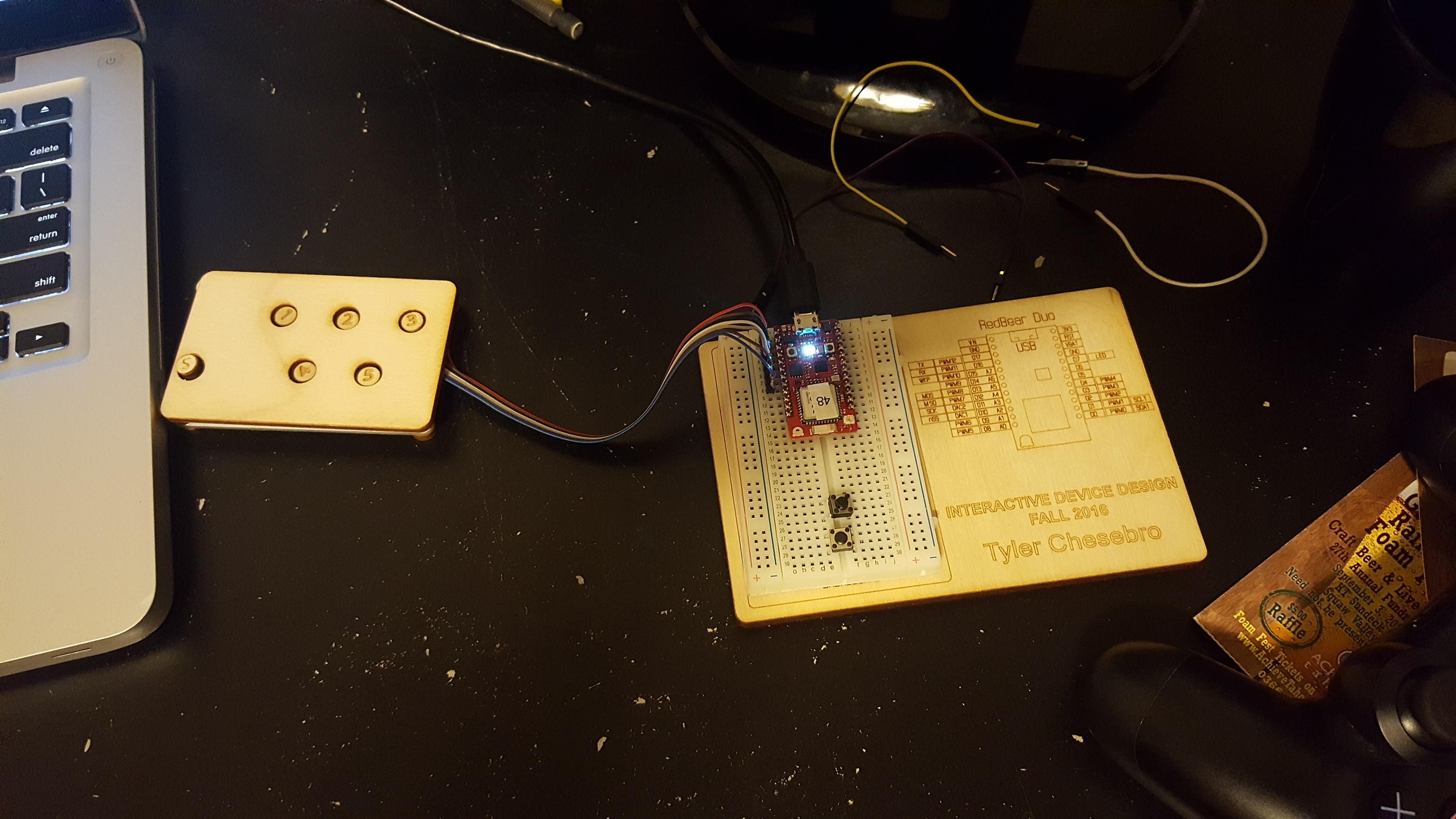
The order of the letters has also been altered. Rather than type alphabetically, letters are assigned to key combinations based on their descending frequency in the English language. The key combinations are chosen in increasing complexity/number of buttons needed to be pressed. The ordered alphabet I used is shown below with number of buttons needed to press color coded: yellow is one button, green two, blue three, and red 4:

**e t a o i n s r h l d c u m f p g w y b v k x j q z**

Buttons are pressed in the as if they were binary digits, with a press as a “1” and a not press as a “0”. Meaning pressing 2, 3 and 4 simultaneously would equate to “2+4+8” giving the users the 14th character as the output. Below is a table of all outputs on the keyboard. (for this coding, I only went up to 32 characters, but all of the capitol letters could easily be added in code later.)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **e** | **t** | **n** | **a** | **s** | **l** | **p** | **o** | **r** | **u** |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| **g** | **d** | **w** | **y** | **q** | **i** | **h** | **m** | **b** | **c** |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| **v** | **k** | **z** | **f** | **x** | **j** | **.** | **,** | **?** | **;** |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| **!** | **space** |

**CONSTRUCTION**

The board is soldered on an Adafruit circuit board and mounted with laser-cut plywood on both sides for protection. The wiring is well contained and I used soldered all wires to a male-header to more easily plug into the breadboard of the RedBear Duo. 





The circuit diagram is below:



**REACTION:**

Overall I found the project very enjoyable, I felt it was a good introduction to the laser cutter (the part I am least experienced with) and the programming for the duo. I learned the most about timing and debouncing switches on the duo as my project involved sensing concurrent button pressing. In future applications of the project I would have taken more care in making sure the buttons were large enough to protrude above the cover layer.