Caitlin Gruis
9/11/16
HW#2

Text Entry Device Writeup

URL to GitHub Repository: https://github.com/cgruis/IDD_HW_2_gruis
YouTube Video of Working Device: https://youtu.be/vMu0i-hHeS8

For my text entry device, I created a Morse Code Controller, complete with a key, as shown in Figure 1 below. It utilizes three switches, with one switch for the "dot" button, one switch for the "dash" button, and a third switch for the submit button.

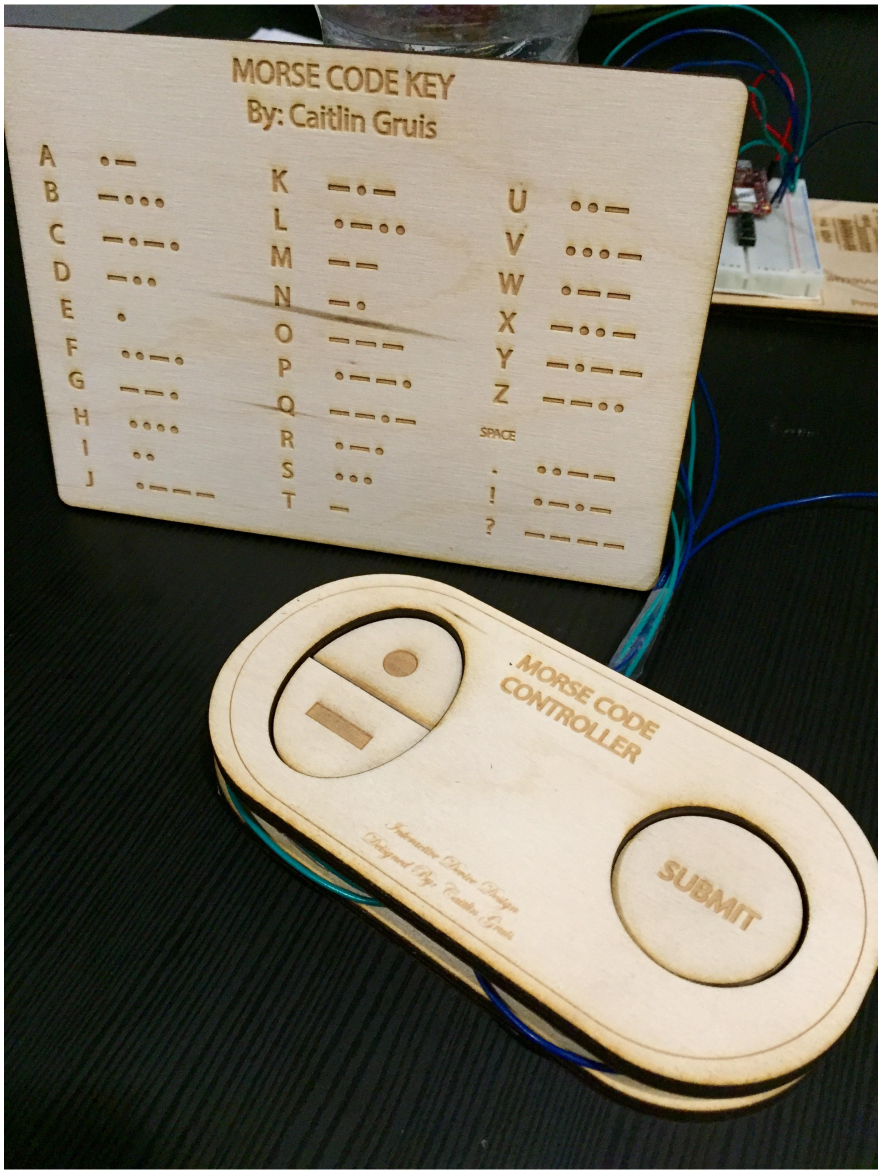

Figure 1. Morse Code Controller and Key

I chose this text entry technique because I thought that the “decoding” aspect of Morse code was interesting. For techniques such as multitap or direction pads, you are directly entering alphabetic letters onto your user interface. With Morse code, however, there is an added level of complexity in that you have to decode the dots and dashes and translate them into their corresponding alphabetic letters and symbols.

I was able to fully implement my character recognition using only the Arduino IDE. The code works by first creating an array called morseCodeArray. The array has four values, corresponding to the number of dots and dashes an alphabetic letter or symbol can potentially have. When loop() runs, it checks the value of each of the three buttons. If the “dot” button is pressed, it fills the nth value of the array with a 1, and increments n. The same goes for if the “dash” button is pressed, except it fills the nth value of the array with a 2. If at any point the submit button is pressed, it sends morseCodeArray to the morseDecoder function. This function looks at the array and determines what character the array corresponds to based on the order of the 1s and 2s in it. It then returns this character to the main function, where the character is then printed on the serial monitor, and the morseCodeArray and n are both cleared. The hardware wiring diagram for my device is shown in Figure 2 below. Note also that the pulldown resistors on pins D2, D3, and D4 were activated to prevent shorting power to ground.

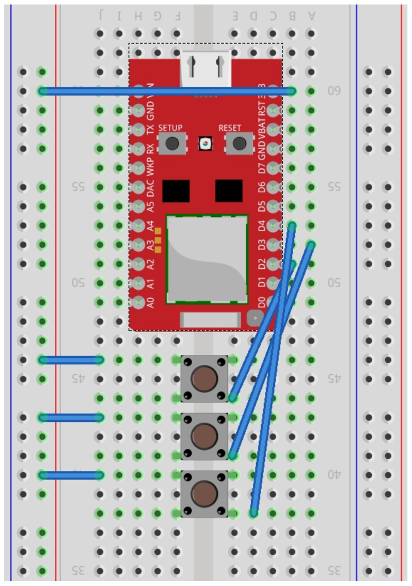

Figure 2. Morse Code Device Wiring Diagram

The physical device was constructed using a laser cutter machine to create a controller. I decided on a controller because I thought that it would be an intuitive, familiar way to enter text, and that this would make entering the less-intuitive Morse code much easier. The controller features two laser cut oval controller shapes, with the actual momentary switches and wires sandwiched between the two pieces so that the hardware is hidden from the user. Wires are routed out of the controller and into the RedBear Duo microcontroller.

From this assignment, I learned that something that seems very simple to the user (entering text on a device) can actually become quite complicated on the software end of things. Also, hardware can be unreliable and not always give you the results that you expect. With that being said, I thought that this assignment was at just the right level of difficulty. There were some things that I was more comfortable with (like the coding, because I've previously programmed a lot in the Arduino IDE), and other areas of the assignment that were more challenging for me (like doing my first big laser cutting project). I also really enjoyed having the ability to be a lot more creative than I usually get to be in an engineering class.