

# *Persistence of Vision Wand*

## *Assembly Project Report*

*Charles Lambert*

*CSMP*

*Pacific Union College*

*Angwin, United States of America*

*c.lambert1213@gmail.com*

### I. PROPOSAL

The assembly project I proposed was a persistence of vision wand. The wand would have the main function of slicing an image up flashing the piece a particular frequency allowing the image to be seen as the wand was waved. The sub functions were the ability to change the message and turn on and off the device.

### II. COMPONENTS

For components I used

- A Attiny85 Microcontroller [1]
- A 16-pixel long string of WS2812 LEDs [2]
- A simple resistor button

### III. PROCESS

I began my first few test spikes using a code library made by adafruit, which allowed the LED strip to be programed. The adafruit library unfortunately did not result in a useful program as the adafruit code was extremely memory hungry and not well optimized. Through some research my professor showed me a library developed by Josh Levine [3] which was optimized for leds I was using. After I managed to get the leds working in a constructive fashion, I began the task of creating a program to parse a sentence into a bitmap that could be displayed. My first few iterations were semi successful but time consuming and troublesome. After talking to my professor, he lead me to edit a piece of code from Josh Levine's code that would parse the bitmaps I created. Using this edited code and a program called Dot factory [4] to create the bit maps I was able to successfully parse and display messages to the leds. The next iteration would be the implementation of a button to change the message and turn the device on and off. This process was sped up considerably thanks to a bit of code [5] that my class mate Jeff showed me. This allowed me to use short presses for one event and long presses for another. I had to do quite a bit of research on the Attiny85's sleep modes to achieve the effect I wished for my device. It required changing several values within specific registers and setting up button interrupts. Ultimately I got it working after a few test spikes. For changing the message, I implemented simple structures in c to hold the message and its color information, I then put three of those structures in an array

and cycled through it upon a short button press. After the main functionality was complete I spent some time refactory Josh Levine's code with intent of eliminating a bit of code that repeats itself. Specifically, the on off cycle for the leds. Through a long amount of testing I found that the .rept directive made it impossible to push variables into the assembly as it requires a constant value upon assemble time, so I simply took the if statement that wrapped around the assembly and converted it into a branch statement within the assembly.

### IV. FURTHER IDEAS

There were a few ideas that I did not implement in this project to due to time constraints. One would be adding an accelerometer to the wand so that the frequency at which the wand flashes the message would be consistent no matter how fast the wand was waved. The other would have been the ability to have messages be multicolored instead of only a single color.

### V. REFERENCES

- [1] Atmel Corporation, "Attiny85," 2016. [Online]. Available: <http://www.atmel.com/devices/attiny85.aspx> . [Accessed 16 March 2016].
- [2] Worldsemi Co., "WS2812 Datasheet," [Online]. Available: <http://cdn.sparkfun.com/datasheets/Components/LED/WS2812.pdf> . [Accessed 16 March 2016].
- [3] J. Levine, "SimpleNeopixelDemo," 2015. [Online]. Available: <https://github.com/bigjosh/SimpleNeoPixelDemo/commit/f3ce28bfc4956f6e082490c7f62192b28e071be> . [Accessed 15 March 2016].
- [4] E. Duchan, "The Dot Factory," 2009. [Online]. Available: <http://www.eran.io/the-dot-factory-an-lcd-font-and-image-generator/> . [Accessed 16 March 2016].
- [5] Salsa, "Click for A, Press and Hold for B," 2009. [Online]. Available: <http://jmsarduino.blogspot.com/2009/05/click-for-press-and-hold-for-b.html> . [Accessed 15 March 2016].