Eugene Ngo | 1965514

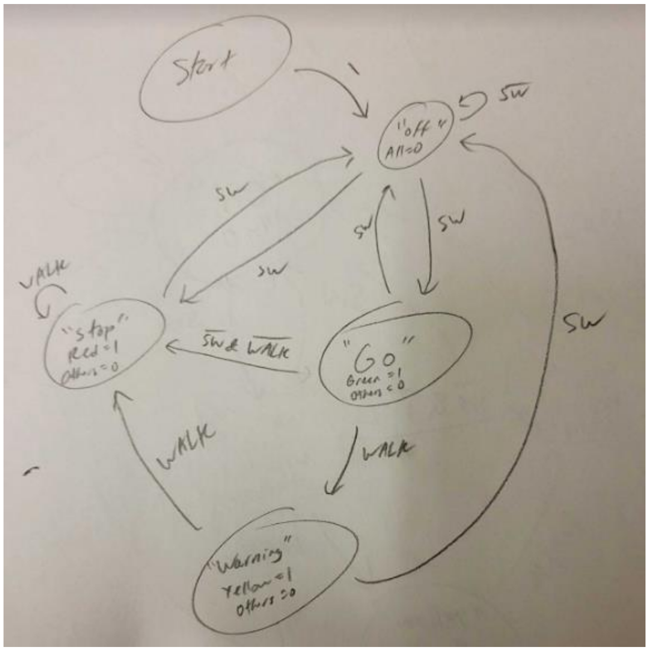
Lab 2 Report

Task 1a Procedure:

I used the same GPIO pins as before but the implementation of this task was different since we have begun using the built in general purpose timer modules. I followed the lab procedure to enable and configure the timers and set it to achieve a 1 Hz blink rate. Once I did this, I applied it to the program that I had for lab 1, but instead using the timer modules. I would go through turning on each LED sequentially based on the timer, clearing the bit, then turning off the LED, turning the next one on, and so forth.

Task 1b Procedure:

The state diagram will be the same as the first lab and is attached below.



For task 1b I built the timers into the program so that inputs would only register as true and be read as a user input, when the user pressed and held the button for two seconds. I used the timer for this task. Once the switch is pressed then the timer would begin and if held for two seconds then the input would return true and repeat the same process as the traffic light in lab 1. Each LED was held in its state for 5 seconds unless the system was turned off or the pedestrian switch was held for two seconds in which case the state would switch from 'go' to 'warn' immediately.

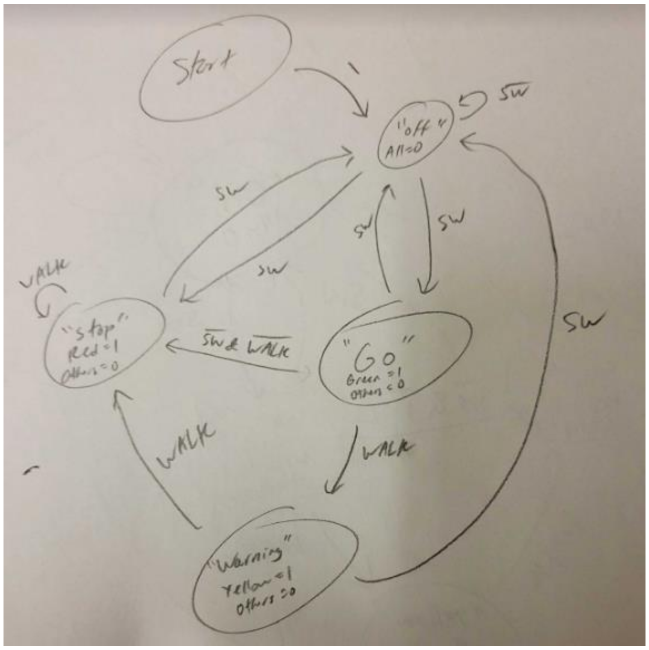
Task 2a Procedure:

The set up for this task was the same as Lab 1 and task 1a from this lab. Using the same GPIO pins, we could make the same LEDs blink. Then initialize the timer by looking through the data sheet to find the timer pins, enable them and configure them to have a frequency of 1 Hz. Then in the main function, I combined the two: enable the pins for the GPIO pins to point to the onboard LEDs and initialize and run the timer alongside it to achieve the continuous sequence of lights at a rate of 1 Hz.

Task 2b Procedure:

Using the same GPIO pins as before I enabled the onboard LEDs 1 and 2. Then using the data sheet to find the correct pins for the timer again, I enabled and configured them to have a frequency of 1 Hz and interrupt once the desired frequency is reached. Then once again, in the main function I enable and initialize the GPIO pins to point to the onboard LEDs and initialize and run the timer.

Task 2c Procedure:



Same state diagram again. Then, using the same GPIO pins in task 1, I enabled the traffic LED lights. Once again, I refer to the data sheet to find the timer pins and enabled them. Then configure them to have a frequency of 32 MHz and to have a 2 second delay on checking the button state. Then there is another timer that I set up to run at a frequency of 80 MHz to have a 5 second delay on transitioning states. Then in the main function, we enable and initialize the GPIO pins to point to all the traffic LEDs and initialize and run the different timers, with transitions between states being dependent on the 5 second timer and the button presses depending on the 2 second timer.