EE 474
Cynthia Li
7/19/2021

Lab 2 Report

Procedure:

Task 1a:

This task aims to show a periodic lighting pattern of the on-board LEDs and to familiarize us with timers. Therefore, instead of using for loop delay as in Lab1, we now set a timer to count down in each second to control the desired frequency.

Task 1b:

This task prompts us to reimplement Lab 1 task2 using a timer. To achieve desired results, I used the same method as in 1a to count for 1 second and a for loop to prolong it to 5 seconds. Inside the loop, I checked the button input length using a global variable so that when it reaches 2 seconds, it will be considered as a valid press for the FSM to assert state change. FSM as below in Fig. 1.

Task 2a:

This task aims to show a periodic lighting pattern of the on-board LEDs and to familiarize us with interrupt. I initialized the interrupt in the interruptInit and modified Timer0A_Handler to fulfill the periodic shining pattern.

Task 2b:

This task aims to familiarize us further with the use of the cstartup file, the vector table, and the interrupt. I implemented the periodic 1Hz lighting of D1 in the main function, and used an interrupt to control the response to each switches' input and D2's lighting in the PortJ_Handler. Then I added PortJ_Handler to the cstartup file, specifying it as the extern function and in the vector table.

Task 2c:

This task basically repeats task 1b, but instead of using a timer to count for seconds, we now use interrupt. FSM below as in Fig. 1.

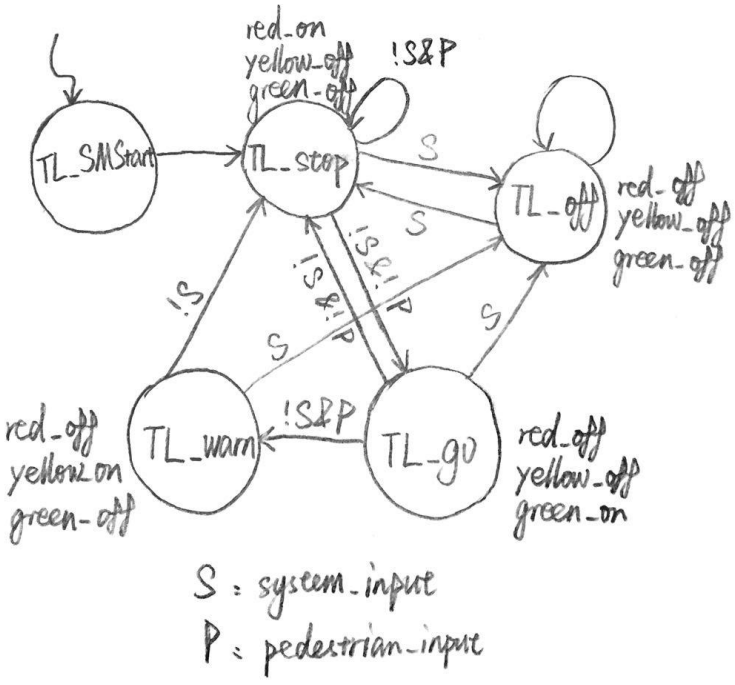

Fig. 1 FSM for Traffic Light

Results:

Task 1a:

The four LEDs turn on in the sequence of D4 to D1 one by one and turn off from D4 to D1 one by one. This pattern appears periodically. (Details see assignment 1 demo).

Task 1b:

The traffic light behavior is basically the same as Lab 1 task2 except the lighting period is longer. By default, the system is off, meaning all three LEDs are off. When the system button is pressed for longer than 2 seconds, it turns on the system to be in the stop state, thus the red LED is on. When there is no further button input, the system periodically repeats the pattern: only red LED on for 5 seconds, only green LED on for 5 seconds. When the pedestrian button is pressed for longer than 2 seconds when the system is in go state (green LED on), the system enters warn state (only yellow light on) and stays for 5 seconds, then enters stop state. When the system is on, in any state, as long as there is a longer than 2 second press on the system button, the system will be turned off (all three lights turn off). Circuit implementation is as below in Fig. 2, 3. (Details about functionality see assignment 1 demo)

Task 2a:

The on board LEDs have the same behavior as Task 1a. (Details see assignment 2 demo).

Task 2b:

By default, D1 turns on and off periodically at 1Hz rate. When SW1 is pressed, D1 stays at current status (on/off) and D2 turns on. When SW2 is then pressed, D2 turns off and D1 goes back to the periodic shining pattern. (Details see assignment 2 demo).

Task 2c:

The traffic light has the same behavior as Task 1b. Circuit implementation as below in Fig. 2, 3. (Details see assignment 2 demo).

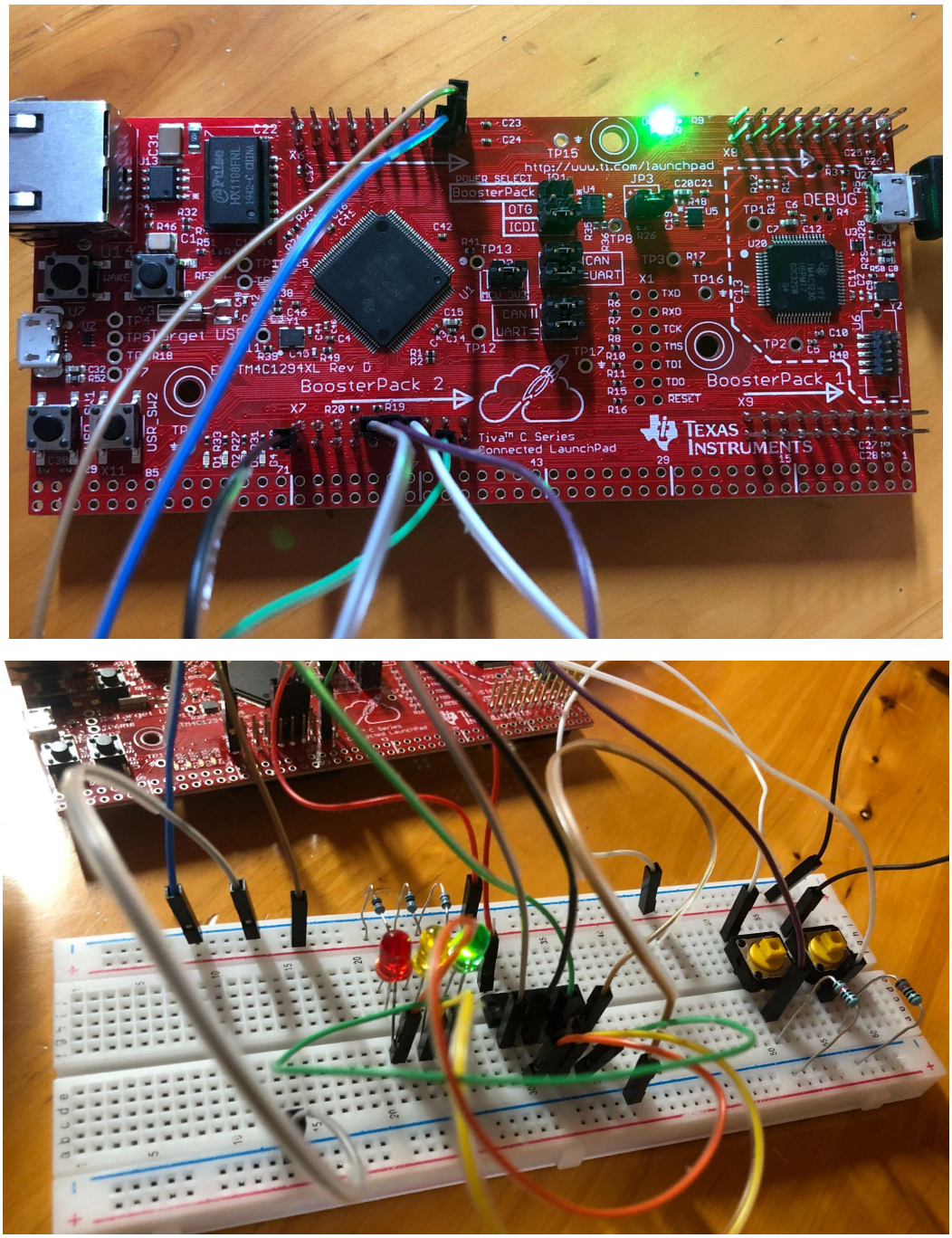

Fig. 2, 3. Traffic Light circuit built up.

Feedback:
This lab challenged me a lot during the debug and learning about the timer/ interrupt configuration so I had to reach out to TA and classmates for help. Fortunately, the functionality was able to be achieved in the end. But overall, it seems very meaningful to know more functions of the TIVA board and it was good practice for debugging.