EE 474
Cynthia Li
7/3/2021

Lab 1 Report

Procedure:

Task 1a:

This task aims to let us familiarize ourselves with the datasheet, header fields, and the onboard LED by asking us to turn on and off the 4 internal LEDs in a periodic pattern. First, I looked through the user manual to find the ports for the LEDs, and the datasheet for the address corresponding to each port. Then I used this collected information to define these variables in the header file and initialize them in the main. Last, based on the given code for delay and to light LED4, I changed GPIODATA for each LED to alter their state and added delays in between to create the pattern.

Task 1b:

This task aims to let us familiarize ourselves with the use of the internal switch of the launchpad and bitwise control by asking us to use 2 switches to control 2 LEDs. I checked the user manual for user switches' ports and found the corresponding address of ports on the datasheet. Then, I used this information for the header file and initialization in the main function. To set conditions in the while{1} loop, I used the if statement with GPIODATA of switches after bit operation as conditions, in each statement, GPIODATA of the LEDs are changed based on the case .

Task 2:

This task aims to let us familiarize with the implementation of SM in C language by asking us to develop a switched controlled traffic light model. I chose the M port on GPIO for switch and LED connection, and updated the address, etc. for the header files and functions provided to initialize the switches and LED. Then, I drew the SM on paper to list the transitions and actions in each state. Next, I followed this chart to implement the code on IAR. Lastly, I built up the circuit on the breadboard and debugged the code.

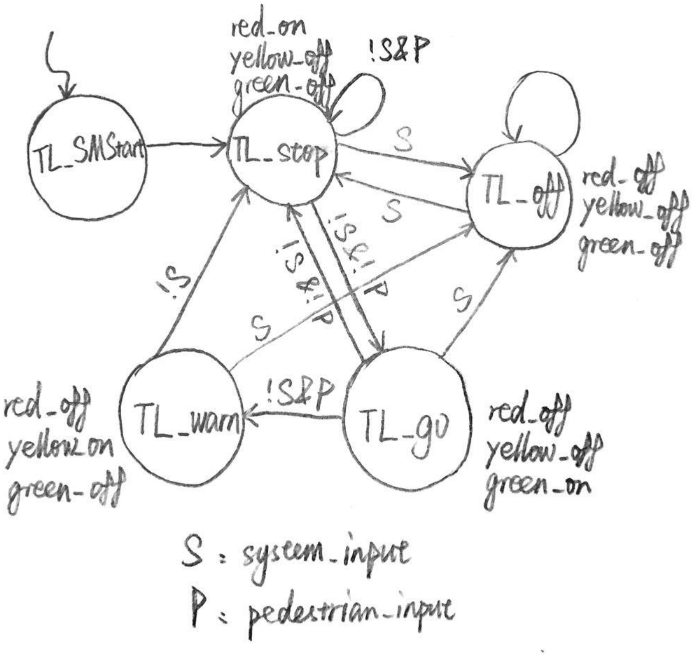

Fig. 1. SM of trafficLight

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 four LEDs are defaultly turned off. If user switch 1 is pressed, D1 turns on; If user switch 2 is pressed, D2 turns on. If both are pressed, D1, D2 both turn on. (Detail see assignment 1 demo).

Task 2:

There are two switches, each representing the system on/off button and the pedestrian button, and three LED lights including red, yellow, and green, one for each color. The system has the following functionality. By default, no button is pressed, thus the system is on and periodically repeats the pattern that the red turns on and off and then the green LED turns on and off; when the system button is pressed, all LED lights are turned off; when the system is off and the system button is pressed, the system is turned on again, the red LED lights up and would repeat the red-green pattern if no further button is pressed; when the system is on, and the pedestrian button is pressed when the red LED is on, the red LED stays as lit up and other lights are off; when the system is on, and the pedestrian button is pressed when the green LED is on, the green light turns off, the yellow LED turns on, last for a short time and turns off, and the red LED turns on. The circuit setup is as shown in Fig. 2-3. (For specific details about functionality, please refer to the demo)

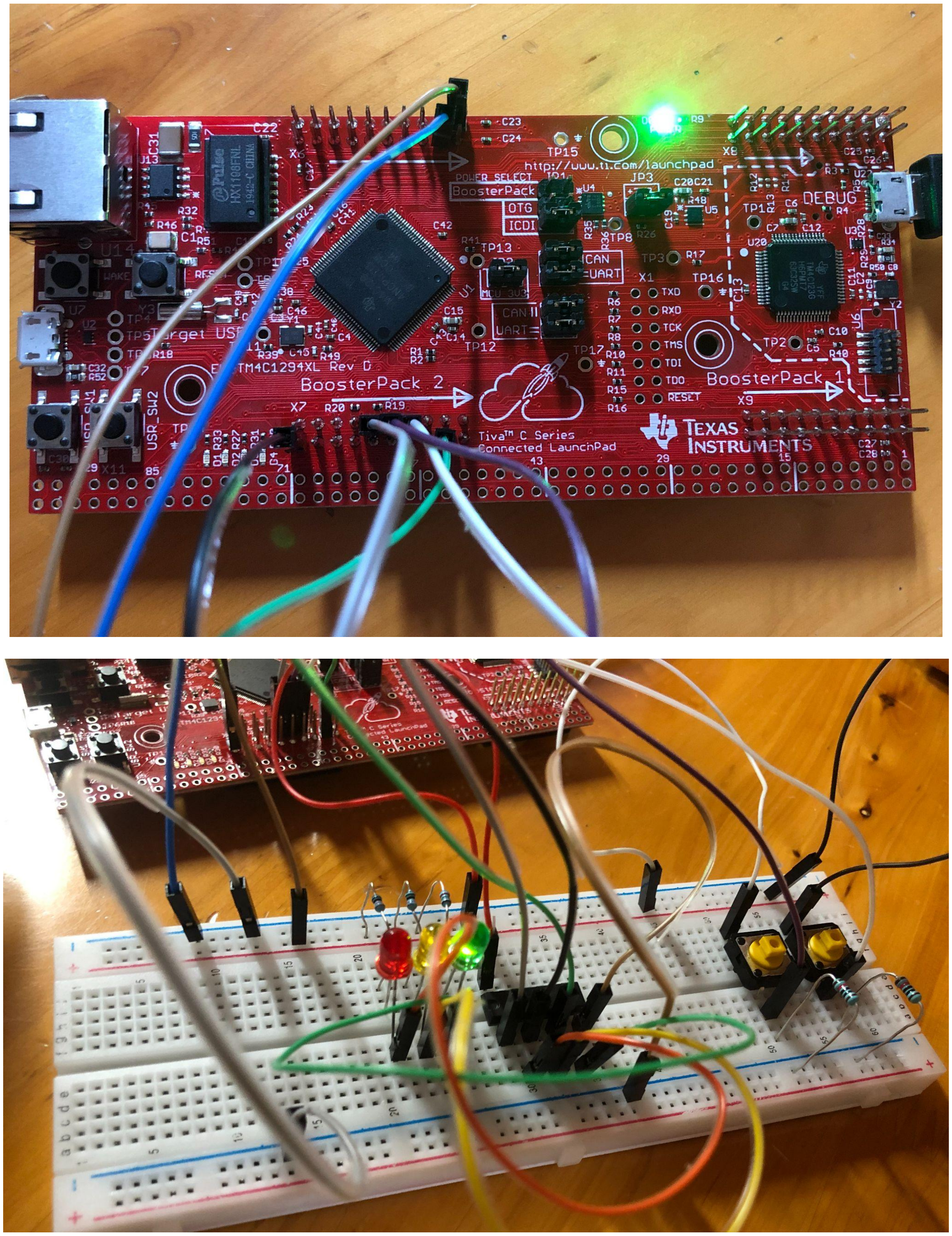

Fig. 2, 3. Traffic Light circuit built up.

Feedback:

The lab is a good design in a way to familiarize us with the basic use of the softwares, the buttons and LED, and how to debug step by step, so it wasn't hard. However, it did appear to us (at least to my group) that there are some parts not very clear, such as the behavior of the system during the stop state in task 2.