EE 474
Cynthia Li
8/5/2021

Lab 4 Report

Procedure:

Task 1a:

This task aims to familiarize us with the LCD's basic functions. Following the lab specifications and the comment, I checked the tm4c1294ncpdt.h file to look for the defined names and enable the port to finish LCD_GPIOInit(). Then I created the main file to select the color for the LCD screen.

Task 1b:

This task aims to require us to implement the LCD_printFloat() function. Since most of the functionality is the same as Lab3 task 1b and the only difference changed the print to I/O to LCD display, I copied all the code from Lab3, added LCD initialization, changed the printf() functions in ADC0SS3_Handler() to LCD_printf() function. The temperature is printed using LCD_printFloat().

Task 1c:

Task 1c requires us to read through the provided code and learn the functions of LCD to utilize its display and touch function. I copied the code from 1b and deleted PortJ_Handler since we no longer use on-board buttons. Since we have many LCD features to create, I used a new function LCD_Pattern() to define the regions. In this function, besides the original printed text from 1b, I created two buttons using LCD_DrawFilledRect(), one filled with bright blue color for SW1, the other filled with bright red color for SW2. I also added two labels to make the switch clearer ***(This feature was added after the demo, I thought the color distinction is clear enough, but using labels is better, so I hope it's not a big deal).*** Also, since we now use virtual buttons, we need to determine the effective region of the buttons. I used printf() function to print out the x, y coordinates read from touch screen to I/O, observe the edge scale for x and y, and used it to define SW1(), SW2() which represents whether the button is pressed or not. SW1(), SW2() are then used to determine the system clock we use. Other part of the code is basically the same as 1b.

Task 2a:

This is the main function file for task 2a, the majority of the code is from Lab2. I added an LCD_Pattern() function to set the LCD_display and revised sys_switch() and ped_switch() using printf() to find the appropriate value range of x and y and double check the press at the start and the end of the 2 seconds to ensure a long press. The go(), warn(), stop(), and off() are functions to change virtual LED patterns. TrafficLight is the FSM that defines the state transitions and actions of the program. It has the same logic as lab2. The FSM is shown below in the next page, Fig. 1.

Task 2b:

This task aims to familiarize us with free RTOs. I first set up the project as specified in instructions and read through the main files, etc. I added the sub files such as PLL_Header, init files, etc. Then following the comment guide, I first completed the initialization in the main and assigned the priority of the tasks. When implementing StartStop and Pedestrian, I realized that these are the functions to update the global variables of whether the button is pressed for longer than 2s so I implemented sys_switch and ped_switch that determine a press in the button region and let the StartStop and Pedestrian function to determine the press length using tick_time as condition. Then Control and FSM is basically the same as in Task 2a, except that for state transitions we can now easily use global variables as conditions. I also added four functions to include state actions. The FSM is shown below in Fig. 1.

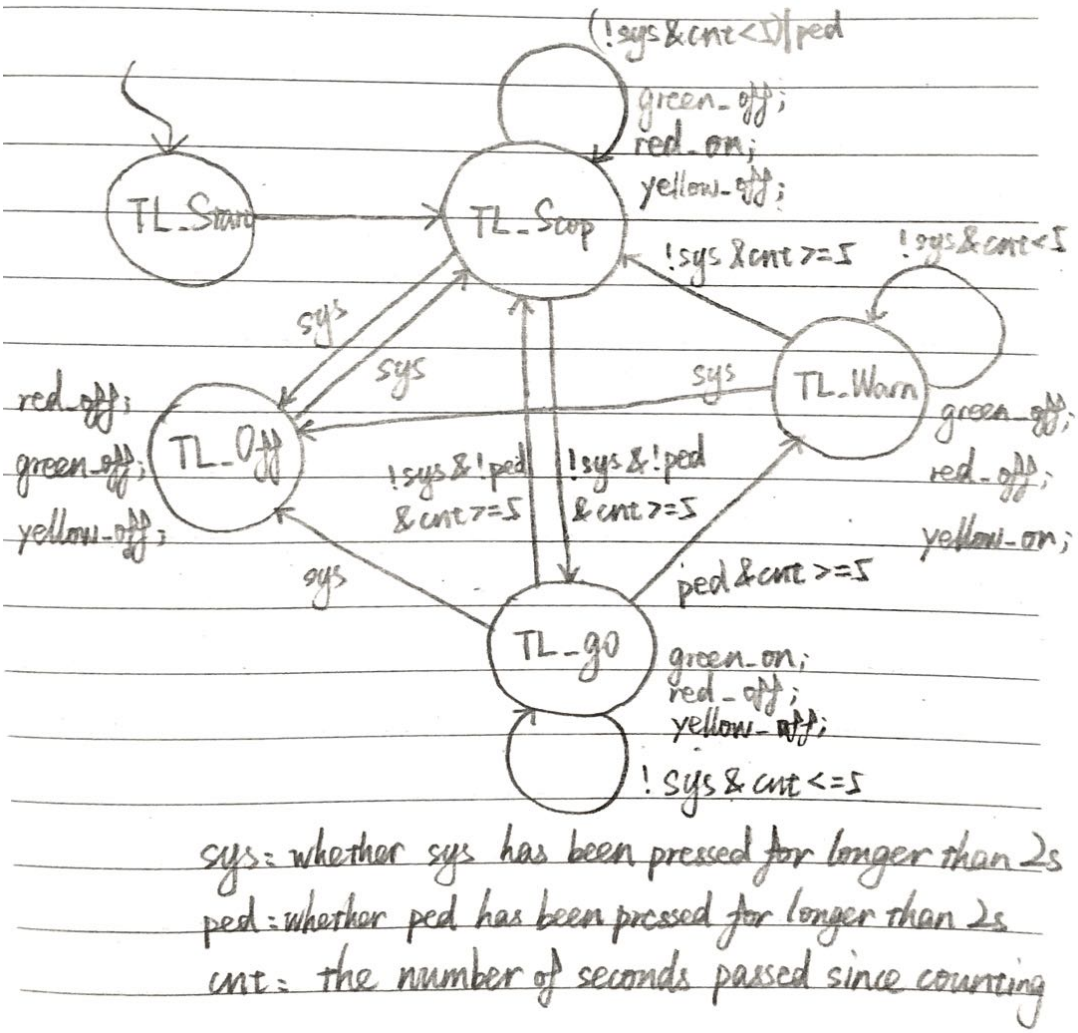

Fig.1. FSM of TrafficLight

Results:

Task 1a:

The screen is filled with bright blue but a different color can be chosen by other Color4 indexes.

Task 1b:

Two on-board switches control the system clock frequency (by default it's 60 MHz). SW1 changes it to 12 MHz, SW2 changes it to 120 MHz. The text describing the read board temperature in Celsius and Fahrenheit and the current system clock frequency is displayed on the LCD screen as required by the lab specifications. When switching from SW1 to SW2, we observe a temperature increase for about 1 celsius degree.

Task 1c:

The functionality is the same as 1b but the buttons are now virtual buttons on the LCD screen: the blue button labeled "SW1" on the left is SW1, the red button labeled "SW2" on the right is SW2.

Task 2a:

The LCD screen is filled with white background, and it has 3 black LEDs (off) and two blue circled buttons (labeled, not filled), "sys" and "ped". The button is considered pressed if the press is longer than 2s. The system will turn on to stop state if "sys" is pressed. It will repeat the stop-go state transition if no button is pressed. Whichever state the system is in, if the "sys" button is pressed, it will turn off the system. If the"ped" button is pressed, when it is in stop state, it will stay in current state while when it is in go state, it will proceed to warn state.

Task 2b:

The functionality is the same as 2a, it just used a different implementation method.

Conclusion:

This lab is relatively simple. It let me gain knowledge of LCD's function and implementation and use it to achieve more complex purposes. It is also interesting to learn free RTOs. One important skill I've been practiced is to learn from the open source code and use it for our own purposes.