Garrett Tashiro

CSE 474 Lab 2 Report

July 16, 2021

Section 1: Procedure

I approached the assigned tasks by going through the lab document and finding all the registers that would be used for the lab. Once I had them written down, I went to the data sheet and wrote down every base, offset, bit/field, and what the value in the bit/field did. With all the information for the registers written in my notebook, I started by adding the registers needed to the header files. I worked on one header file and one main at a time (essentially one part of one task). What I could copy from the pervious lab I did, then added and changed what was needed for these assigned tasks. I worked with smaller portions code at a time and ran the program to see if a light or switch would work correctly with the timers, or interrupts. I broke down the tasks in to smaller tasks for myself to assure that I could get each part working individually and built upon that. For task 2 B I created a state diagram seen in figure 1. The diagram helped show the states and transitions from state to state.

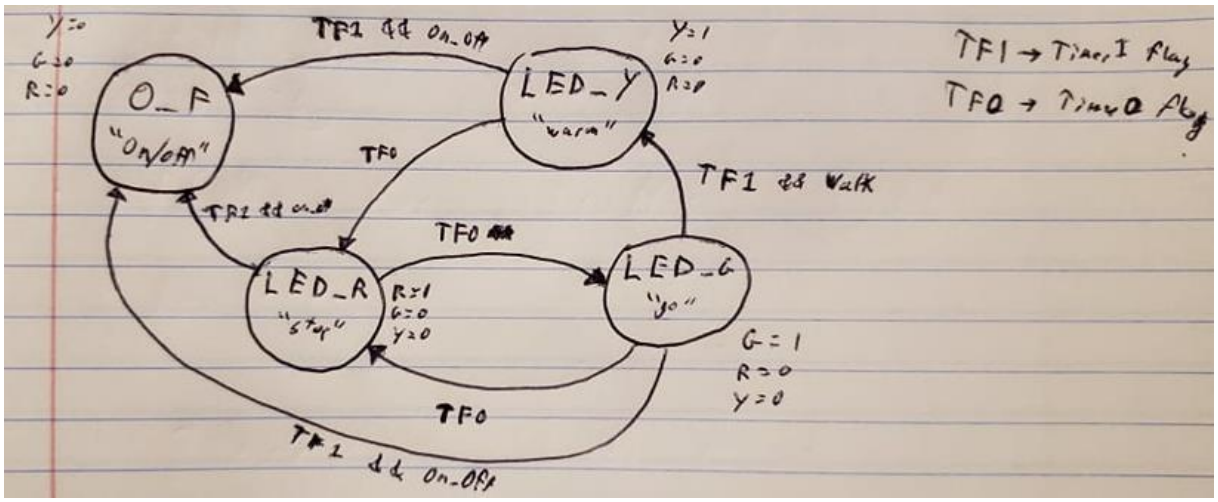

Fig. 1: State Diagram for Task 1 B

The way I organized my code for lab two was by putting all functions that initialized things at the top. I kept smaller functions that turned on/off an LED next. I put the timer reset(s) after that. The main always called a function to run the programs and had the while loop in that function along with a function to call the functions for initialization. Task 1 uses come GPTM testing done in larger functions, but for task 2 I have all values being changed, or tested in small functions by themselves, then used those functions throughout all other functions in the code. I tried to make most functions as small as possible, and then used them in other functions when needed.

Section 2: Results

Task 1 A:

The results for this task were as expected. The timer took care of when LED's turned on/off, and the LED's would turn on/off sequentially. Whenever the timer hit 0, an LED would turn on or off. The LED's lit up from left to right one at a time until all four LED's were lit up, and then turned off in the order they turned on. There was a one second delay between LED's turning on/off.

Task 1 B:

The program worked as expected, and each switch press responded correctly if held for two seconds. LED's would change after five seconds. The LED's were all off to start. When the on/off switch was held for two seconds the red LED would turn on for five seconds. After five seconds the red LED would turn off, and the green LED would turn on. If the pedestrian switch was pushed and held for two seconds with the green LED on, the green LED would instantly turn off and the yellow LED would turn on. Pushing and holding the on/off switch in any state would turn off the LED's.

Task 2 A:

The program worked like Task 1 A, but used a timer interrupt to initiate LED's turning on/off. When the interrupt is triggered, the LED's change accordingly. The timer would count down for one second, and upon hitting 0 an LED would turn on or off. The LED's lit up from left to right one at a time until all four LED's were lit up, and then turned off in the order they turned on. There was a one second delay between LED's turning on/off controlled by the interrupt.

Task 2 B:

The results from task 2 B were interesting. The pausing of the timer when SW2 was pressed would keep LED1 on or off was something new to me. Having the switch handler take care of this was interesting. LED1 blinked on/off in one second intervals. If SW2 on the board was pressed, LED1 would stay on if it was one when you pressed SW2, and off if it was off. LED2 would turn on when SW2 was pressed. The LED's would stay in the states they were at until SW1 was pressed. Once SW1 was pressed, LED2 would turn off, and LED1 would resume its blinking sequence.

Task 2 C:

The results obtained from task 2 C were almost identical to task 1 B. The only difference was that the switch presses didn't have to stay pressed for two seconds, but had to be held at the end of the two seconds. You could release the switch for a moment and then press it again to have the LED's change according to the switch pressed. The LED's were all off to start. When the on/off switch was pressed a switch interrupt would be called which would hold a value depending on which button was pressed and start the two second timer. Once the two second timer was up, the handler for the timer would call a function to turn on the red LED. If any switch was pressed, the switch handler would hold a certain value depending on the switch pressed and start the two second timer. The two second timer handler would test the value and see if the switch corresponding to that value was being pressed after two seconds and turn LED's on/off accordingly. The five second timer handler would constantly switch between the red and green LED unless any of the switches were pressed and "held" to trigger events to happen in the two second timer handler. If the green LED was on for the duration of the press/hold of the pedestrian switch, the green LED would immediately turn off, and the yellow LED would turn on for five seconds. After the yellow LED was on for five seconds it will turn off and the red LED will turn on. The normal switching between red and green will commence. If the on/off switch was pressed and held at any point, and during and LED being on, that LED would turn off. No LED's would be on, and won't turn on until you pressed and held the on/off switch for two seconds and the red LED would light up.