CALIFORNIA STATE POLYTECHNIC UNIVERSITY, POMONA
COLLEGE OF ENGINEERING

ECE 3301L Spring 2019-2 Microcontroller Lab Felix Pinai

LABs 6&7: Traffic Light Controller with the use of RBG LEDs and a LCD Panel

We will need to design a traffic light control system that will include the following items:

1) 4 sets of RGB LEDs with the following assignments:
    a. 1 RGB LED for North/South direction
    b. 1 RGB LED for North/South Left Turn direction
    c. 1 RGB LED for East/West direction
    d. 1 RGB LED for East/West Left Turn direction

2) Four DIP switches used as sensors with the following designations:
    a. NS Switch Pedestrian (NSPED) switch acting to indicate pedestrian present in the North/South direction
    b. NS Left Turn (NSLT) switch acting as a sensor for cars making left turn on the North/South direction
    c. EW Switch Pedestrian (EWPED) switch acting to indicate pedestrian present in the East/West direction
    d. EW Left Turn (EWLT) switch acting as a sensor for cars making left turn on the East/West direction
3) Light Sensor to define the mode of operation. There will be two modes: Day Mode and Night Mode.

4) A LCD TFT Display used to show the operation of the controller. A typical screen shot is shown below:

#

#

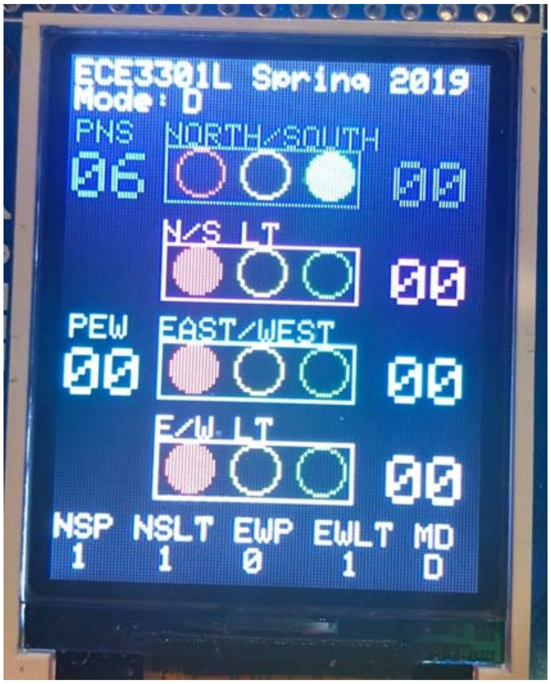
#

#

#

#

#



Here are some descriptions on the information above:

1) First Line: Name of this class and its semester

2) Mode: Actual mode the controller is running. A ‘D’ is to indicate ‘Day’ mode while ‘N’ is for night operation

3) ‘NORTH/SOUTH’: The rectangular box below this line shows the three different colors in three circles for the traffic light in that direction: RED, YELLOW, GREEN. When the color is active, the circle is filled with that color. When not active, only the outline of the circle is shown

4) The two ‘00’ digits at the right of the ‘NORTH/SOUTH’ are used to display the number of seconds left on this direction when the lights are turned into the green and yellow colors.

5) ‘N/S LT’, ‘EAST/WEST’, ‘E/W LT’: the same descriptions on the item above apply to these lines

6) ‘PNS’ and the number right below show the actual number of seconds left on the Pedestrian Counter in the North/South direction.

7) ‘PEW’ and the number right below show the actual number of seconds left on the Pedestrian Counter in the East/West direction.

8) The line at the bottom shows the following: ’NSP NSLT EWP EWLT MD’. The numbers below that line show the status of the switches for the North/South PED, NSLT sensor, East/West PED, EWLT sensor and the logic state of the light sensor for MODE.

The operation of the traffic light will be described on Lab #7. The purpose of this lab is to prepare the hardware and some of the supporting software routines needed to make the implementation easy and fast.

The schematic of the design is shown on the attached file.

The TFT Panel uses the hardware interface called SPI Bus in order for the microcontroller to display data on the TFT’s screen. The basic understanding of the SPI bus will be visited on a later lab and the hardware explanation will be handled at that time.

In addition, since it would take a good amount of time to implement all the software routines in order to allow the user to display texts or to generate different types of shapes, a library software has been compiled and put together into the file "ST7735_TFT.inc". This file should be included at the beginning of the ‘.c’ program to provide all library functions used in the program.

To develop the basic screen shown on the above screenshot, it will take a good amount of time to do so. We will skip this step and instead a basic routine is provided to the student that will create the basic framework of the screen. The student should look at the provide routine ‘Initialize_TFT()’ to get the basic ideas to implement the rest of the requirements of this lab.

Here are the definitions for the various values that the variable ‘direction’ can have:

```
#define NS          0           // Number definition of North/South
#define NSLT        1           // Number definition of North/South Left Turn
#define EW          2           // Number definition of East/West
#define EWLT         3           // Number definition of East/West Left Turn
```

Here are the definitions for the various values that the variable ‘color’ can have:

```
#define Color_Off       0      // Number definition of Off Color
#define Color_Red       1      // Number definition of Red Color
#define Color_Green    2      // Number definition of Green Color
#define Color_Yellow  3      // Number definition of Yellow Color
```

We will need to implement the codes for the following functions:

A) void update_color (char direction, char Color)

This routine will change the color of the traffic light on the TFT panel for the specified ‘direction’ with the specified ‘Color’. Fox example, if this function is called with direction=NS and Color=Color_Red, then the RED circle in the North/South field should be filled. The other two colors (Yellow and Green) should only have its outline drawn but the inside of the circle should not be filled.

B) void update_RGB (char direction, char Color)

This routine will change the color of the RGB traffic LED for specified ‘direction’ with the specified ‘Color’. Fox example, if this function is called with direction=EW and Color=Color_Green, then the Green light in the East/West RGB LED field should be turned on.

C) void update_Count(char direction, char count)

This routine will change the value of the Sec Counter (right side of the traffic light) for the specified ’direction’ with the value indicated by ‘count’.

D) void update_PED_Count(char direction, char count)

This routine will change the value of the PED Counter for the specified ’direction’ with the value indicated by ‘count’. There are only two PED counters, one for NS and one for EW

E) void update_misc()

There are four DIP switches and one light sensor uses as general inputs:

```
#define NS_PED_SW   PORTAbits.RA3      // Location of Switch for North/South PED
#define NS_LT_SW    PORTAbits.RA4      // Location of Switch for North/South Left Turn
#define EW_PED_SW   PORTBbits.RB0      // Location of Switch for East/West PED
#define EW_LT_SW    PORTBbits.RB1      // Location of Switch for East/West Left Turn
```

Once these switches are read, the following variables must be updated accordingly:

```
char SW_NSPED;                // RAM variable for NS Pedestrian Switch
char SW_NSLT;                 // RAM variable for NS Left Turn Switch
char SW_EWPED;                // RAM variable for EW Pedestrian Switch
char SW_EWLT;                 // RAM variable for EW Left Turn Switch
```

Next, the information of those variables must be updated on the screen under the header NSP, NSLT, EWP and EWLT respectively

In addition, the voltage of the photo sensor PR1 must be read and from its value the variable SW_MODE must be set to 1 if the sensor is in daylight versus a 0 for dark mode.

```
char SW_MODE;                 // RAM variable for Mode Light Sensor
```

Then under 'MD' on the TFT panel, a 'D' should be displayed for day more or 'N' for night mode.

F) void wait_one_second()

This routine will need to implement the following steps:

1) Turn on the 'SEC_LED' LED
2) Put the character '*' on the top right corner of the TFT screen
3) Do a delay for 500 msec by calling the provided routine delay_ms(int ms)
4) Turn off the 'SEC_LED' LED
5) Erase the character '*' on the top right corner of the TFT screen.
6) Do a delay for 500 msec
7) Call the 'update_misc()' routine developed on E)

G) void wait_one_second_beep(char direction)

This routine is almost identical to F) with the additional task that it will generate a beep sound:

1) Turn on the 'SEC_LED' LED
2) Put the character '*' on the top right corner of the TFT screen

3) Generate a 2Khz sound if the direction is NS. If the direction is EW, then generate a 1Khz sound
4) Do a delay for 500 msec
5) Turn off the ‘SEC_LED’ LED
6) Erase the character ‘*’ on the top right corner of the TFT screen.
7) Turn off the buzzer
8) Do a delay for 500 msec
9) Call the ‘update_misc()’ routine developed on E)

The file ‘traffic.c’ is provided to the student with some codes on the main() routine that will call directly or indirectly the above functions. If the implementations are done correctly, then information on the TFT panel and the RGB LEDs will be updated properly. See the demo from the instructor to confirm that the student’s implementation is done correctly.

Once the supporting routines are done, we will use them on the next lab to implement the traffic light controller.