

EE 361L Digital Systems and Computer Design Laboratory

University of Hawai'i

Department of Electrical Engineering

Lab 2.2: Traffic Light Controller

Version 3.1

Date: September 8, 2011. Revised by: Roxanne Yee. Further revision by G. Sasaki

Design a controller for the traffic light on Dole Street near the Parking Structure (in the Quarry) using the PIC16F648A. That traffic light controls the automobile traffic on Dole Street and the pedestrian traffic crossing the street. The outputs of the controller are Red (R), Yellow (Y), Green (G), Don't-Walk (D), and Walk (W). There are two inputs, B1 and B2, which are each connected to a Pedestrian-Walk button. The outputs and inputs correspond to the pins of the PIC as follows $R = RB0$, $Y = RB1$, $G = RB2$, $D = RB3$, $W = RB4$, $B1 = RB5$, and $B2 = RB6$. Connect LEDs to $RB0$, $RB1$, $RB2$, and $RB3$ (you can choose appropriate colors).

There are four states of the traffic light as shown in Figure 1:

- ❑ **R/W1**: Traffic light is Red and pedestrian light is Walk. The controller is in this state for 3 seconds.
- ❑ **R/W2**: Traffic light is Red and pedestrian light is a blinking Walk. The Walk is blinking to caution pedestrian that it will soon turn to Don't Walk. The Walk goes on 1/2 sec, off 1/2 sec, on 1/2 sec and so on. The controller is in this state for 3 seconds.
- ❑ **G/D**: Traffic light is Green and pedestrian light is Don't Walk. The controller is in this state for at least 6 seconds. The controller will get out of this state when
 - 6 seconds have elapsed *and*
 - one of the pedestrian buttons has been pushed (goes on) since the end of the last R/W1 state, i.e., the beginning of the last R/W2 state.
- ❑ **Y/D**: Traffic light is Yellow and pedestrian light is Don't Walk. The controller is in this state for 3 seconds.

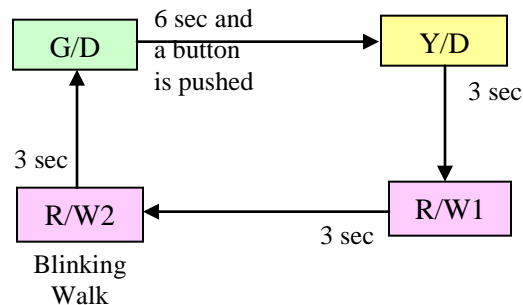


Figure 1. Transitions of traffic light controller.

You will need a half-second delay. Implement the half-second delay using TIMER0. You should have no assembly language instructions in your code.

Hints:

Use the MPLAB simulator as much as possible to design your code.

First, make sure that your traffic light controller is logically correct. Don't worry about the half-second delay just yet. In fact, in place of the half-second delay, use a much shorter delay so you can run the simulation, maybe just a simple stub that doesn't do anything. You can also test your implementations using breakpoints. Then replace the stub with a short delay for few tens of instruction cycles. Again you can debug using break points.

Recall that you should be able to simulate input values (for the push buttons of the traffic light controller) using Stimulus.

When designing your code, make sure that you can easily upgrade to a half-second delay.

You should be able to get most or even all your design done before lab if you have access to MPLAB software. This should be available in the HP Lab in POST second floor (POST 208).

When using the protoboard and the actual circuit, to facilitate debugging, have an additional output (LED) to indicate if a pedestrian button (B1 and B2) has been pressed.)