

## CECS 346 Lab 3 – Simple Traffic Light Controller

### Preparation:

You will need a LaunchPad, two push buttons or switches, two 10k $\Omega$  resistors, two sets of the following color LEDs: red, yellow, and green, and six resistors for the LEDs (between 330 $\Omega$  to 1k $\Omega$ ).

**Book Reading:** Textbook 2.7, 4.2, 6.1 – 6.6,

**Reference Project:** SimpleTrafficLight

**Starter project:** Lab 2

### Purpose:

The purpose of this lab is to learn how to implement a Moore FSM to control a simple traffic light system, learn how to use SysTick timer to generate delay, and review GPIO.

### System Requirements:

In this lab, you will build a simple traffic light control system using two switches to simulate two sensors and six LEDs represent one set of traffic lights in north/south and east/west directions. You are required to implement **positive logic** for both switch interface and LED interface. You are required to use a Moore FSM for your software design. No conditional statement is allowed in your FSM engine. **SysTick timer** is required to provide the time delay needed for this embedded system.

### Configuration:

- You will use two sets of color LEDs: one set for north/south and another set for east/west, and two switches to simulate two sensors: one for north/south and one for east/west.
  - Port assignment:
    - Traffic lights: you can use Port E: PE5-0 or Port A: PA7-2.
    - Switches: you can use port A: PA3-2, or Port B: PB1-0
- You can click the drop down menu at the TExaS edX Lab 10 traffic light GUI to choose your input and output ports. See two examples in Figure 1.

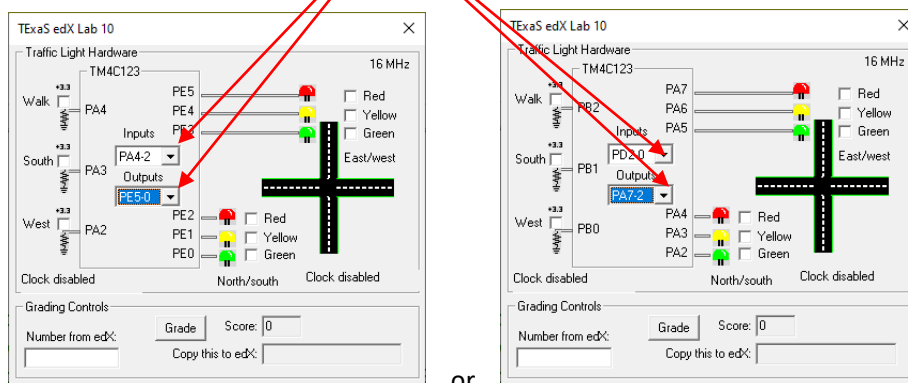


Figure 1. Traffic Light Simulation Interface.

Functionality:

The system implements a traffic light for traffic flowing at an intersection.

- 1) The system starts with the green light on east/west, red light for north/south.
- 2) The timing requirement is: 2s for green, 1s for yellow.
- 3) If cars are detected in only one direction, move to and stay green in that direction
- 4) If cars are detected in both directions, cycle traffic lights to allow cars to pass in both directions (e.g. North-South, East- West, North-South, ...)

### Extra challenge (5 points):

Add one more feature to this simple traffic light control system: add two more switches, one for each direction, to simulate special vehicles, such as ambulance or police cars. When cars are detected in both direction and special vehicles are detected in one direction but not the other direction, move to and stay green in that direction until no more special vehicles in that direction, then cycle traffic lights to allow cars to pass in both directions. You cannot simulate the two extra switches with edXLab10.dll. You will need to debug this feature on board.

### Procedure:

1. Design and implement your software, simulate it with Keil uVision. Simulate your program on Keil using Logic analyzer to show the output traffic lights wave form. To bring up the logic analyzer window: click the drop down window for “view”->Analysis Windows->Logic Analyzer. Then Click the **setup** button at Logic Analyzer to set up all output signals (6 LED lights). See Figure 2 below for an example.

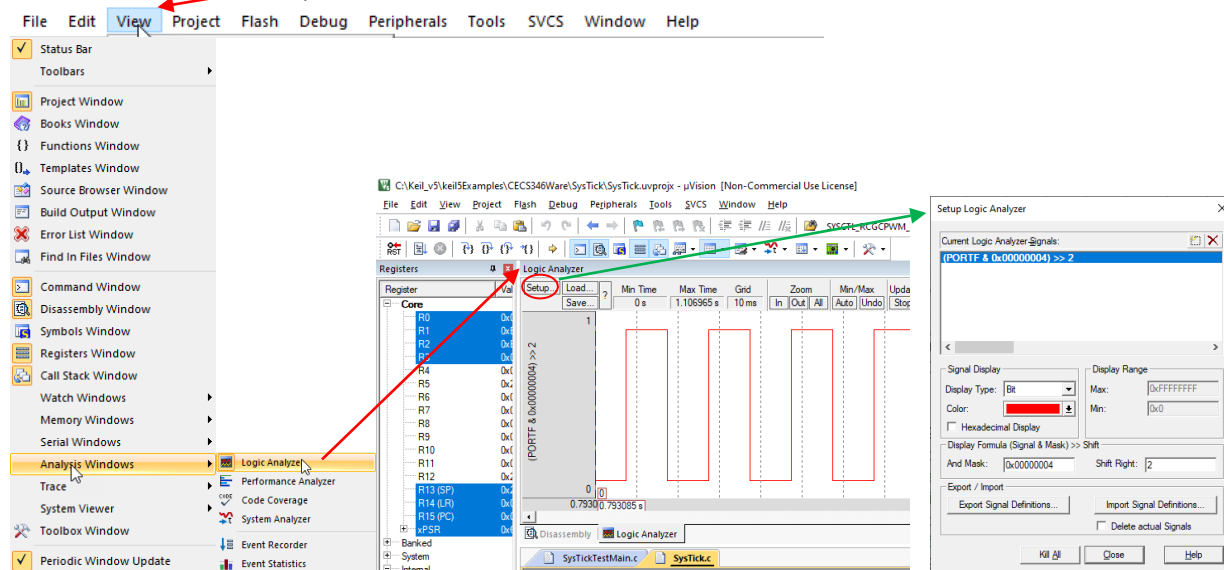


Figure 2. How to open Logic Analyzer window.

**Make sure to move the display bit to the least significant bit position and set the display type in the setting window to “Bit”.**

2. Download your program to Launchpad and test it on your embedded system.

**Deliverable:**

- 1) Demonstrate your lab
  - a. in simulator
  - b. on board
- 2) Submit a lab report (e.g. Word Document) to the Beachboard Dropbox containing:
  - a. Class name, lab number and name, your name
  - b. Schematic of your hardware
  - c. A picture of your embedded system
  - d. Software source code: The .c file
  - e. Video or video link for your onboard and simulation demonstration.

You can attach items a, b, c, e to end of this file. Submit the source code separately.