EECE 344 Digital Systems Design

Lab 4: Finite State Machine (FSM)

1 Pre-Lab

1. Review the lecture notes about the finite state machine (FSM).
2. Read the textbook Section 6.5 and review basic C programming in Sections 6.1 through 6.4.
3. Read the textbook Section 6.4 to review the C `struct` data structure.
4. You can reuse the code you have written for the past labs or any code samples from Canvas as the starting point for this lab.

2 Objective

The objective of this lab is to practice a design process:

1. analyze a given task;
2. describe your design ideas in a finite state machine;
3. interface the circuit of your design ideas;
4. implement your finite state machine in embedded C code;
5. practice bitwise logic operations and `struct` in C.

3 Requirements

Required parts: the Tiva LaunchPad for TM4C123, jumpers, ADALM 2000 (as the +5V power source and oscilloscope), LEDs, and a breadboard.

Work in pairs for this lab assignment.

4 System Requirements

In this lab, you will design a controller for a visitor booth. The Toyota Motor Manufacturing Kentucky in Georgetown, Kentucky assembles the Toyota Camry, Toyota RAV4, and Lexus ES along with producing engines. It welcomes visitors to tour their plants. However, to protect the plants to be clean, visitors are required to go through a booth for hygiene purposes. The booth is shown in the following Figure 1.

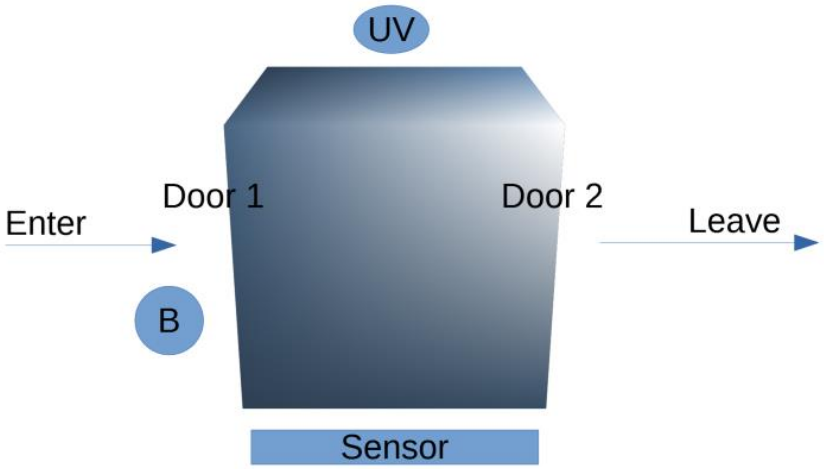

Figure 1: A Visitor Booth at Toyota

You are expected to design a controller for the booth to work as follows.

1. The booth has two doors, Door 1 and Door 2. It has a UV light on the top of the booth. It has a weight sensor on the floor of the booth.
2. To open Door 1, there is a push-button "B". Whenever a visitor presses "B", Door 1 opens for 2 seconds if there is nobody inside the booth. Door 1 remains closed if there is a person in the booth.
3. When there is a person inside the booth, the sensor should detect the person and turns on the UV light for 4 seconds.
4. After the 4 seconds, Door 2 opens. The person inside the booth is to leave the booth within 3 seconds. Door 2 should close after the 3 seconds have passed.

4.1 Build the interfacing circuit

Interface your Launchpad using 2 switches (push-buttons) as inputs by **active-high** on PA7 and PA6, and 3 LEDs as outputs on PA5 (LED-1), PA4 (LED-2), and PA3 (LED-3).

1. The Switch 1 on **PA7** is expected to simulate "B" button of the booth. Pressing Switch 1 means there is a visitor pressing the "B" button.
2. The Switch 2 on **PA6** is expected to simulate the weight sensor. Pressing Switch 2 means the visitor is inside the booth.
3. LED-1 on PA5 is to indicate Door 1 being open or not. When Door 1 is open, then LED-1 is on. Following the system behavior descriptions in Section 4 above, Door 1 is open for 2 seconds. Therefore, LED-1 should be off after 2 seconds.
4. LED-2 on PA4 is to simulate the UV light. If the UV light is turned on by the weight sensor (Switch 2), then LED-2 is on. Following the system behavior descriptions in Section 4 above, the UV light is on for 4 seconds. Therefore, LED-2 should be on for 4 seconds if Switch 2 is pressed.
5. LED-3 on PA3 is to indicate Door 2 being open or not. If Door 2 is open, then LED-3 is on. Following the system behavior descriptions in Section 4 above, Door 2 is open for 3 seconds. Therefore, LED-3 should be off after 3 seconds.

4.2 Design and Implement the Embedded Software

Read the problem description carefully and create your finite state machine for the booth controller. The state transition diagram should be drawn using the free online flowchart tool: https://www.draw.io/
You should make clear notation on the graph about the meaning of your states. Export your diagram into a PDF file. This PDF file is your diagram to be submitted.

1. The timing should be achieved by making the **PLL** to run at **50 MHz** and use the SysTick to get the desired wait time. Feel free to reuse and modify your code from last lab.
2. **You must design a controller using Finite State Machine using the C struct data structure and cycle through the states in the controller program.**

5 Grading

The grading of your lab comprises the following components. Refer to the rubrics about the ranking of each component.

5.1 Performance of the Lab

Show the performance of your built system. Explain the interfacing circuit to your instructor.

5.2 Demonstration

When demonstrating the program, you are expected to explain each line of code if asked. Each student will be asked a different question to demonstrate his/her understanding of the project at hand.

5.3 Deliverables

Please indicate who your lab partner is in your deliverables. Submit your deliverables on Canvas to “Lab 4: Finite State Machine” under “Assignments” by the due time and date. Refer to the schedule shown on Canvas. For this lab, your deliverables are:

1. Provide state transition graph you designed. Use software to create the graph, as any hand-drawn graph will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by a professional software, such as PCB Artist (available at https://www.4pcb.com/free-pcb-design-software.html). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.

5.4 Code

Submit your code on Canvas to “Lab 4: Finite State Machine” under “Assignments” by the due time and date. Your project folder should contain all C code, compressed into a zip file. Include comments and indentation in your code. Points will be deducted for sloppy code lacking documentation.

5.5 Lab Report

You (as an individual student, not a group) may choose to write a lab report for this lab if you have not fulfilled the requirement of writing three lab reports. If you choose to write a report for Lab 4, submit your report on Canvas “Assignments” by the due time and date. Follow the instructions in the section on "Lab Report Writing Guidelines" in “EECE 344 Digital Systems Design Lab Policy.” In the body of your report, make sure you include the following:

1. A state transition graph of the program you designed. You may use the graph you prepared for Deliverables (5.3 above). Use software to create the graph, as any hand-drawn graph will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by professional software. You may use the circuit you prepared for Deliverables (5.3 above). Any hand-drawn circuit will be classified as at best second ranking or lower according to the rubrics.
3. A photo of the circuit on the breadboard for the system.
4. Technical writing explaining your design and the graph of your code. Describe any challenges in your experiment and how you solved the problems. Draw a conclusion about designing and implementing FSM on the LaunchPad.