

## LAB #3: FSM for a Security Code Access System

Lab writeup is due to your TA at the beginning of your next scheduled lab. Don't put this off to the last minute! There is pre-lab work to complete before the start of the next lab. **NO LATE LAB REPORTS WILL BE ACCEPTED.**

### 1 Objectives

- Use the parallel I/O functions of the CSM12C32.
- Learn about usage of FSM abstraction in C.
- Design a FSM for a security code access system.

### 2 Reading

- Read Chapter 2.3-2.11

### 3 Background

In this lab, you will use the parallel I/O ports on the CSM12C32 to design a security code access system. This device will accept four digits that it will compare with a stored access code. If the digits entered match the code stored, then an LED (LED1 on the project board) will be turned on to indicate that the lock has been opened. A second LED (LED2 on the project board) will be used to indicate that the system is ready to accept the first digit. In other words, LED2 will be illuminated initially. After the first digit is entered, it will go off. After the fourth digit has been entered, LED2 will turn on again and at the same time if the proper code has been entered LED1 will also turn on.

In lab 4, you will be using a keypad to enter the digits, but for this lab you can use the push buttons (5,6,7,8) located on the Project Board. To keep it simple, you can assume there are only four possible digits which are *one-hot encoded* (i.e., one switch or wire will be connected to high at a time). The FSM should determine that a digit is being pressed by seeing one of the four inputs going high. It should then record which digit is pressed and wait for 10ms to debounce the switch. Finally, it should wait for the digit to no longer be pressed (i.e., its input pin goes low). It should repeat this process four times checking along the way whether or not the digits being entered match the hard coded security access code. If all four digits match, then both LEDs are turned on. If any digit does not match, then only LED2 will turn on after all four digits are entered. Both LEDs will turn off when the first digit in the next code attempt is entered. The system should use push button 4 (PB4) on the project board as a reset. This enables the user to reset the system during the entry of a code. When PB4 is pressed the system should return to the initial state with LED2 illuminated.

For this lab, the user option will be disabled (by removing the USER jumpers located on the Project Board). If you wish to use any of the push buttons or LEDs on the project board, you'll need to wire them up directly. Please refer to the [Application Module Student Learning Kit Users Guide \(APS12C32SLKUG\)](#),

[MCU Project Board Student Learning Kit User Guide](#) and [MC9S12C Family Reference Manual](#) for detailed hardware information. In addition, remember to put the USER jumpers back on after you're done.

## 4 Tasks

1. Design the FSM for your security code access system.
2. Prepare a schematic for your design. The schematic can be a traditional schematic or a table like the table used on slide 41 of [Lecture 2](#). The schematic should contain information about each function connected (PB4, LED1, etc.), the User I/O signal number on the project board (if applicable), the MCU board port number, and the MCU Port (PT0, PT1, etc.). Pin numbers on the MCU for each port are not required but may be added for completeness.
3. Design the C program for your security code access system. Do not use loop delays, but instead use the TCNT timer to implement the time delays in your FSM. Be sure that it is designed using the software issues discussed in class. For example, it should consist of separate modules (ex., subroutine for waiting, another for reading which key is pressed, another for waiting for the pressed key to be released, and another for setting the LEDs, etc.). It should also be well documented.
4. Connect your circuit and debug your software. Then check-off your working security code access system with your TA.

## 5 Prelab

Complete tasks 1, 2, and 3 before coming to lab.

## 6 Writeup

Include the following items:

1. Your FSM diagram.
2. Your hardware schematic.
3. A printout of your commented C code.
4. Describe any problems that you encountered.

## 7 Extras

This addition to your project will result in 5 extra points. Note: projects that are not functional for the required tasks are NOT eligible for extra points.

1. Output a letter (a,b,c,d) or a number (1,2,3,4) for each key that is pressed to the LCD display. If the 4 digit code is correct output, "Success" for 2 seconds then return to the default message, "Please enter your code:". If the 4 digit code is incorrect output "Fail" for 2 seconds then return to the default message. You submit a second FSM diagram with the changes.