CSE 4342 Embedded Systems II
Fall 2023, Lab 1

This lab is due by September 10, 2023.

In this lab, we will interface the controller with a SPI bus expander. The SPI bus expander is used to add additional GPIO connections to the controller if the controller is pin-limited or can also minimize routing of many signal paths across a system board. The lab will also interface an interrupt request from the bus expander to allow devices to alert the controller of changes in status.

The following steps will guide you through this process:

1. Create a file lab1_*your_name*.c, where *your_name* is replaced with your name as it appears in MyMav, which implements steps 2-6 and 9-11. You should use the CSE 4342/4352-specific device libraries for this lab.
2. Using the SSI0 interface on the TM4C123GXL board, connect an MCP23S08 bus expander to the appropriate clock, TX, RX, and chip select lines for the SSI0 interface. Connect the interrupt output of the bus expander to a GPI. Pull the A1 and A0 pins on the bus expander low.
3. Interface two LEDs (red and green) and a pushbutton to the bus expander on GPIO 4, 5, and 7 respectively.
4. Write code to configure the SSI interface for operation at a 2 MHz rate. Note that the chip select needs to be low for the sequence of device opcode, register address, and one or more data bytes.
5. Write code to initialize the bus expander over the SPI bus so that the LED pins are outputs and the pushbutton pin in an input.
6. Modify code in the stop go main function to work with the bus expander components. It should behave identically as the examples from class.
7. Configure the oscilloscope in a single trace mode, with normal trigger on the falling edge of the chip select, and capture the clock and TX lines during a write to the device.
8. Configure the oscilloscope in a single trace mode, with normal trigger on the falling edge of the chip select, and capture the clock and TX lines during a read to the device.
9. Enable interrupts on the bus expander and the GPIO pin being used for the interrupt connection. Add an ISR to handle this interrupt.
10. Modify the stop go code in step 6 to wait 1s after the green LED turns on. At that time, clear any existing interrupts, enable the NVIC interrupt control, and then turn on the red LED again and turn off the green LED.
11. Write an ISR that is called, read the bus expander device. If the pushbutton is pressed, turn off the red LED and turn on the green LED, effectively moving the check for a pushbutton to an interrupt event and the LED color change to the ISR.
12. Demonstrate your code and e-mail the file to the grader.