

# ECE353 In-Class Exercise

## GPIO Pins – Interrupts/Button Debounce

### Problem Objectives

- Configure the SysTick interrupt to generate a periodic interrupt
- Debounce a push button using SysTick Timer

### 1. Modify interrupt.h

- A. Add the TM4C123.h header file

```
#include "TM4C123.h"
```

- B. Create a function declaration for the SysTick interrupt. The name of the handler should match what is found in the vector table.

### C. Create interrupts.c

- A. Include interrupts.h  
B. Add an extern for a Boolean variable called AlertSysTick  
C. In the ISR, set the AlertSysTick timer to be true  
D. Clear the SysTick Interrupt

### D. Modify main.c

- A. Create a Boolean global variable called AlertSysTick. This variable will be accessed by both the main routine and the interrupt service handler, so declare it as volatile. Initialize its value to false.
- B. Initialize the SysTick timer so that an interrupt is generated once every 5 milliseconds. Use **SysTick\_Config(uint32\_t ticks)** to configure the time.
- C. Read GPIOF once every 5 milliseconds.
- D. Detect when the LEFT or RIGHT buttons have been pressed. For this exercise, a push button is considered to be pressed if the GPIO pin indicates the signal is logic 0 for 20 consecutive milliseconds. This process is called debouncing a push button. **The push buttons have already been configured for you.** You only need to read the PORTF->DATA register and examine the correct bits.
- E. A button pressed for a length  $\geq$  to 20 milliseconds should only report a single button event.

- F. For each button event, print out an indication message to the serial debug port for the given button.

## **E. What to Turn In**

Turn in `main.c` to the dropbox on the course website.