Demo 3: Systick Timer Interrupts

ECE3056 Fall 2016

September 26, 2016

1 Overview

This program flashes the LEDs of the STM32 Value Line Discovery board in response to button presses, using an interrupt handler. The number of flashes of the LEDs indicates the amount of elapsed time since the board was booted. This value is held in a counter which is incremented by an interrupt handler triggered by the `systick` timer. [1] This is the primary system interval timer on the STM32F100 microcontrollers.

2 Operation

2.1 The `systick` Timer

Interval timers provide a way to schedule an interrupt at a precise time in the future. These can be used to perform time-critical tasks, allowing them to temporarily preempt longer-running lower-priority tasks. The STM32F100 microcontrollers have a large set of timers for various applications. The simplest is the `systick` timer, which provides a way to set an interrupt after a precise number of CPU clock pulses.

Using the `systick` timer as an interrupt source is very simple:

- The appropriate interrupt vector table entry is set to the `systick` handler. [2]
- The `load` register [3] is set to the number of clock cycles until the interrupt, minus 1. This number must be 24 or fewer bits wide.
- The value 3 is stored in the `ctrl` register. [4] This enables the interrupt and starts the timer.

[1] PM0056 p. 150

[2] PM0056 p. 36

[3] PM0056 p. 152

[4] PM0056 p. 151

This sets up a periodic timer at the highest-possible interrupt priority. In our demo, this is used to periodically increment a counter. The interrupt mechanism from Demo 2 is present, only slightly modified. Instead of flashing a number of times equal to the number of button presses, the LEDs will flash a number of times equal to the number of timer interrupts that have ocurred.

3 Further Reading

The source is thoroughly annotated with comments, pointing to the pages in the ST Microelectronics manuals where further information can be found on each system used.

4 Exercises

The following exercises can be completed by modifying the demo code.

1. Instead of silently incrementing a counter, display its lower 2 bits on the LEDs.
2. Set the timer interval so that the counter is updated once every second. What does this say about the CPU clock frequency?
3. Use the timer and a manual count of the number of instructions in a single iteration to measure, for a loop of your choice, the number of instructions executed per clock cycle.