**ECEN 3002**

Real Time Operating Systems Lab #2

GPIO, Timers, and Interrupts

Derek Wright - Spring 2020

All Answers are indicated in blue, and images may accompany those answers.

1. Using the Energy Profiler:
   1. Measure the baseline power consumption in mW when no buttons are pressed and no finger is placed on the touch slider.

A screen shot of a computer

Description automatically generatedThe baseline power consumption is 6.56mW

* 1. Control the LEDs with each pushbutton and record the power consumption in mW when each LED is lit.

A screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generated

The pushbuttons result in a 2.55 and 2.56mA consumption. 8.40mW

* 1. Control the LEDs with the touch slider and record the power consumption in mW when each LED is lit.

A screen shot of a computer

Description automatically generatedThe capsense draws less power: 8.15mW

* 1. Light both LEDs using a combination of pushbuttons and the touch slider and record the power consumption in mW.

A screenshot of a computer

Description automatically generatedWith both LEDs on, 9.68mW is drawn

* 1. Use the compile-time switch to build, run, and compare the power consumption for the polling method in Part I versus the interrupt-driven method in Part II.

The results of the interrupt implementation at 100ms timing interrupts are summarized in the below table, with the polling method shown for comparison.

|  |  |  |
| --- | --- | --- |
|  | **Polling Method** | **Interrupt Method** |
| **Baseline Power** | 6.56 mW | 2.99 mW |
| **Pushbutton Power** | 8.40 mW | 4.82 mW |
| **Capsense Power** | 8.15 mW | 4.54 mW |
| **Both LED Power** | 9.68 mW | 6.36 mW |

As we can see, the interrupt method dramatically saves power. So much so, that the interrupt method with **both LEDs on** doesn’t use as much power as does the polling method when it is **idle**!!

* 1. For the interrupt-driven method, reduce the frequency of the SysTick timer and compare the baseline power consumption.

|  |  |  |  |
| --- | --- | --- | --- |
| **200ms** | **100ms** | **10ms** | **1ms** |
| 3.00 mW | 2.99 mW | 3.54 mW | 3.80 |

I found that 100ms is right around the roll-off point for power savings. When I reduced the SysTick frequency further, I didn’t see much change in baseline power draw. But as I increased the frequency, I saw increase in power draw, cementing the principle of higher frequency 🡪 higher power.

Functional Tests: (\*\* See additions in blue (6,8) to the FTs \*\* )

The following functional tests should pass for both Part I and Part II

1. At reset, observe no LEDs are lit
2. Press Btn0 and observe only LED0 is lit
   1. Release Btn0 and observe LED0 is no longer lit
3. Press Btn1 and observe only LED1 is lit
   1. Release Btn1 and observe LED1 is no longer lit
4. Press both Btn0 and Btn1 simultaneously and observe that neither LED is lit
   1. Release only Btn0 and observe that only LED1 is lit
   2. Release both buttons and observe that no LED is lit
   3. Press both buttons again and release only Btn1 and observe that only LED0 is lit
   4. Release both buttons and observe that no LED is lit
5. Place your finger on the touch slider and slide your finger to the left
   1. Observe that LED0 is lit
6. Place your finger on the touch slide and slide your finger to the right
   1. Observe that LED1 is lit
7. With your finger on the left side of the touch slider
   1. Observe that LED0 is lit
   2. Press Btn0 and observe that only LED0 remains lit
   3. Press both Btn0 and Btn1 and observe that only LED0 remains lit
   4. Press Btn1 and observe that both LED0 and LED1 are lit
   5. Release Btn1 and observe that only LED0 is lit
   6. Place a second finger on the right side of the touch slider and observe that no LED is lit
8. With your finger on the right side of the touch slider
   1. Observe that LED1 is lit
   2. Press Btn1 and observe that only LED1 remains lit
   3. Press both Btn0 and Btn1 and observe that only LED1 remains lit
   4. Press Btn0 and observe that both LED0 and LED1 are lit
   5. Release Btn0 and observe that only LED1 is lit
   6. Place a second finger on the left side of the touch slider and observe that no LED is lit

Grading:

1. 25 points equals “100%” for this lab.
2. All project files and source code must be submitted, such that results can be duplicated by the grader. ZERO score if all project files and source code are not submitted.
3. Basic Construction:
   1. Project builds without warnings and successfully executes as your cited results indicate when compile-time switch is set for “main loop” mode: 3 pts
   2. Able to build without warnings and successfully run as your cited results indicate when compile-time switch is set for interrupt mode: 3 pts
   3. No code that is not unique to the mode is under a compile-time switch: 2 pts
4. Power Measurements:
   1. Measurements provided for each system state as specified: 3 pts
   2. Analysis and comparison of power consumption for main loop method vs. interrupt-driven method: 2 pts
5. Functional Testing:
   1. Appropriate wording of student-specified Functional Test #6: 1 pt
   2. Appropriate wording of student-specified Functional Test #9: 1 pt
   3. Each passing functional test: 0.5 pt each (4 pt maximum)
6. Good Coding:
   1. Header documentation has been provided for all functions: 2 pt
   2. Header documentation follows Doxygen guidelines: 2 pt
   3. Appropriate use of comments inside functions: 2 pt
7. Bonus points (will add to the above-listed 25 pts if the lab is turned in on time—which can push your lab#2 grade above 100%)
   1. For the interrupt-driven implementation, what is the change in baseline power consumption (no buttons pressed and slider not touched) when the SysTick period is increased to 10 ms? Next, try 100 ms and larger intervals. What is the trade-off of increasing the SysTick period? (2 pt)
   2. Is the capacitive sense circuit using Successive Approximation or Delta Modulation? (1 pt)

Point Deductions for Late Submission: -5 points per day