

CSCE 236 Embedded Systems, Spring 2015

Lab 2 Pre-Assignment

Started: Monday, February 9, 2015

Due: Beginning of class Thursday, February 12, 2015

Instructions: This pre-lab assignment is an individual assignment, collaboration is not allowed. If you discuss any problems with others, please note this on the assignment as described in the syllabus. Also note any materials outside of lecture notes, course textbooks, and datasheets that you used. Show your work and describe your reasoning to get partial credit if your solution is incorrect. This pre-lab is due on the date listed above before the start of class.

Name:

Problem 1. How much time did the total assignment take?

1 Detecting Button Bounces

In class, you saw how to use `Timer1`, configured with pin `T1/PD5` as the button input to count the number of times the button was pressed. Download this code from the course website (in the assignments section next to pre-lab2). Examine this code and determine which pins the LEDs and button should be connected to. Connect everything and verify the functionality of the code by pressing the button and observing the counter output on the serial console.

As you saw in class, occasionally the counter incremented more than once for a single press due to the button “bouncing.” Bouncing occurs when the button is being pressed and it rapidly oscillates between logical zero and one. In this section, you will look at two different ways to detect this.

1.1 Manual Delay

First, modify the code so that it only prints the value of the register `TCNT1` when it changes. Be aware that `TCNT1` may change at any point, so you should only read it once in each iteration of your main `loop()` to ensure that the value you are using is consistent.

One way to detect that the button bounced is if `TCNT1` has increased by more than one since the last iteration. However, since the main loop runs very quick, you should add a delay in the main loop so that you only check it periodically, perhaps every 100 milliseconds. Implement this approach and print a warning every time the button bounces.

Bring a printout of your code and the output of your program showing an example where it bounced.

1.2 Checking Time

Using a manual delay is somewhat wasteful in that nothing else can occur during the delay. There is a preexisting Arduino sketch `Examples->Digital->Debounce` which implements debouncing by using the command `millis()` to record when the button was pressed and then to only check the state again after a fixed delay period. Look at this code to see how they implemented it and make sure you understands the code. Ask the instructors if you have questions on how it functions.

1.3 Datasheet on Timers

Read in detail section 16.6 of the datasheet and browse all of section 16 before lab.