EE 2361 - Homework 3

Sunday, March 8, 2020 8:05 PM

Andrea Smith
5423096
Smit9523

Homework Assignment #3

Due: **March 17, 2020, before midnight**

1. Maximum delay using Timer 1
   a. What is the maximum delay that Timer 1 (16-bit) can generate? Assume that the PIC24 microcontroller is running at 16MHz.

$$PR1 = \frac{delay}{Tcy * PRE} - 1$$

delay = (PRx + 1) * Tcy * PRE

= (65535+1)(62.5 nsec)(256)
= 1.04857 seconds

   b. Write a program in C to implement the maximum delay using Timer 1. Reuse the example code `int_first_t1.c` which is available in course Canvas. Implement both polling and interrupt versions. Submit your c file.

$$PR1 = \frac{1.04857}{(Tcy * 256)} - 1$$

= 65535

   c. Measure the time using Stopwatch and report the results. Set a breakpoint at the line `IFS0bits.T1IF = 0;` (or `_T1IF = 0;`) in the infinite loop (polling) and the `_T1Interrupt` ISR (interrupt) and measure the time between the line using Stopwatch. Report the results of the Stopwatch.

Interrupt: 1.04876 seconds

Polling: 1.04876 seconds

2. Assume that INT0 pin receives the following electric signal.

Note that the unit of time is the clock cycle. The goal is to measure the time (in clock cycles) of the high period and the low period and save them to `high` and `low` variables. Use Timer 1 and its register `TMR1` to measure the time. To simulate the signal to INT0, use Stimulus (Pin/Register Actions)[1]. Reuse the example code `int_int0PinT1_determineDutyCycle.c` which is available in course Canvas.

   a. Write a C program using **polling** and run the simulation. Report the value of `high` and `low` variables (i.e. 4 numbers).

high = 998, 497
low = 498, 994

   b. Write a C program using an **interrupt** and run the simulation. Report the value of `high` and `low` variables (i.e. 4 numbers).

high: 1005, 499
low: 503, 1003

   c. Are these `high` and `low` variables close to 1000, 500, 500, 1000? Briefly discuss why the results you obtained are not exactly the same as the ideal values.

Yes, they are close but not exact because when using an ISR, there are a few of code that must be executed before the high and low values are stored.

[1] **Stimulus**: You can find a video tutorial of using Stimulus in MPLAB X IDE at https://youtu.be/4gzeR4YnMFY.

3. Now let's assume that IC1 pin receives the same electric signal that we considered in the previous problem. Reuse the example code `ic_example_with_int.c` which is available in course Canvas.

   a. Write a C program using **polling** and run the simulation. Report the value of `high` and `low` variables (i.e. 4 numbers).

high: 1000, 500
low: 500, 1000

   b. Write a C program using an **interrupt** and run the simulation. Report the value of `high` and `low` variables (i.e. 4 numbers).

high: 1000, 500
low: 500, 1000

   c. Are these `high` and `low` variables close to 1000, 500, 500, 1000? Briefly discuss the differences in the obtained values of IC1 and INT0 and explain why.

The values in the IC version are exactly 1000, 500, 500, and 1000 because this version instantly stores the timer data when it is triggered instead of having to wait for a few lines to execute like the ISR version does.