

ES LAB 2: SysTick.

Student: Johnson Domacasse
Date: 20 sept 2023
Student#: 4471709
Teacher: Suzana Andova

Introduction:

In this lab We were tasked with working with the an internal timer known as SysTick. The SysTick is an internal timer that ticks and allows the system to perform various tasks, like initiating an action on a periodic base or interrupting the system periodically to execute an action. In the procedure section I will talk about what I did in my implementation.

Procedure:

In this lab we focus on two aspects regarding the SysTick, making it work with interrupts and without interrupts. You may notice that the principle stays the same for both of them, but there are different aspects within the code base that are different.

It is important to note that before beginning to code, I had set up my Cortex System Timer to 16 MHz . I also had to calculate the reload value for a my SysTick. I wanted a SysTick interval every 1ms(0.001s), so I multiplied this with my clock frequency. The value of the outcome is: 16000 (-1).

SysTick without interrupt:

I start by setting up the SysTick_Init function. All the function does is simply set to set the reload speed value, setting the interrupt priority to least urgency, resetting the counter value, then enabling both the processor clock and the SysTick itself.

I then enable my LED function. This is by simply setting up the internal LED by using the steps from the last Lab assignment.

Finally I set up my delay function performs a delay using the input value of n.

Within my main function I call the first two functions. In my while loop I then use the Delay function created to set the delay to toggle the LED. In this case, the value is 1000 because I want the LED to blink every second (1000 milliseconds).

SysTick with interrupt:

Same as previously I set up the SysTick_Init function. The only difference compared with the other one is that I set the TICKINT bit as well. I reuse the same function for setting up the LED.

Regarding the delay function, I had to do some things differently. Using the slides as my source, I navigated to the generated "it" file, I took the SysTick handler function and used it in my main file. I then set up the delay function to have a similar working as the last part. I then went into my main loop, where I ran the same code as in the previous part.

Anomaly found:

My initial approach to the calculation had to do with configuring the clock depending on the delay I would want. So I wouldn't use 1ms interval but 1000ms interval. The only difference this makes is that when you set this as your load value, everything will be based on 1 second intervals. I later understood that I want to work with milliseconds. If I wanted to use 10ms delay, I would use multiply my clock frequency by 0.01s to get my load value.

Another thing I found was that I could code the LED like this without needing to use the main function. I didn't know if this was good practice, or if my implementation is either. What I would do is simply code everything in the SysTick handler function. Meaning I wouldn't toggle the LED in the main loop but my SysTick handler function. Just to be safe, I had this method commented out below my original SysTick handler function.