Lab #2

1. What is the address of the `SysTick_Handler()` function? Verify it (i.e. take a screenshot) in the debug environment.

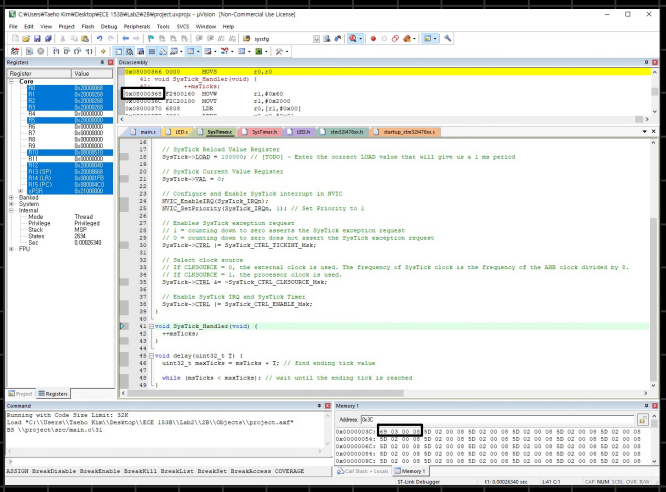
1. Address of pointer = 64 + 4 x n

   Systick interrupt number = -1

   Address of pointer to Systick ISR = 64 + 4 x -1

   = 60

   = 0x3C



   At 0x3C, there's '69 03 00 08' stored.

   Using little-endian, it's 0x80000369 - 1

   = 0x80000368 as shown above.

2. Set up a breakpoint within the `SysTick_Handler()` function. In the debug environment, find out the exception number in the program status register when the program runs to the breakpoint. Explain what this number means.

2. When setting a breakpoint within the SysTickHandler function, the program status register has the exception number of ISR = 15.

   It means that the program is interrupted by a system tick interrupt which is one of the peripheral interrupts.

3. Cortex-M series supports up to 256 interrupts. What is the interrupt number of SysTick that is defined in CMSIS?

3. SysTick_IRQn = -1

4. Does a higher priority value represent a higher urgency?

4. No, priority and urgency have inverse relationship, so a higher priority value means a lower urgency.

5. Suppose a clock of 16 MHz is used to drive the system timer. What is the maximum period between two consecutive SysTick interrupts that we can possibly obtain?

5. $\text{Interval} = \frac{\text{MaxReload} + 1}{\text{SysTick Freq}}$

$= \frac{0xFFFFFF + 1}{16\text{Mhz}}$

$= \frac{16777216}{16\text{Mhz}}$

$\approx 1.049\text{s}$