

# Lab4

Saturday, February 5, 2022 3:47 PM

$\frac{1}{T} = 0.25$

$X = \frac{1}{6}$  cycles

$Y = \frac{6}{12} = 0.5$

$Y \times 12 = 1$

$X = \frac{1}{6}$

$X \times 12 = 2$  cycles

(3)  $\frac{4,500,000}{6,000,000} = \text{time}$

(4) 24 bits = largest value

(5) SysTick is enabled at default  
EXTI isn't

## Pre-lab4

### Pre-lab Assignment 4

Dec 16, 11:48 2022 1/100 (approximately 1 day from the time this page was loaded)

An interrupt is a hardware-invoked subroutine. Some form of trigger causes execution of the normal program to be suspended. A subset of the state of the CPU is saved on the stack. In particular, registers R0-R3, R12, LR, PC, and PSR are saved. This stack content is called an exception frame. A special value is loaded into the LR register to indicate that the stack contains an exception frame. This tells a subsequent `LR` or `POP {r},PC` instruction to load the exception frame contents back into their respective registers. In this way, the interrupt execution can be treated just like a subroutine that can be *returned* from.

The type of trigger decides what form of *exception vector* the CPU should read from the vector table. The 32-bit value read from the vector table is put into the PC and execution continues in this new subroutine. Since the subroutine is meant to handle some kind of interrupt, it is called an Interrupt Service Routine (ISR). Since R0-R3, R12, LR, PC, and PSR have been saved, the subroutine can modify all of the registers that the ABI normally allows. If it modifies R4-R7, they should be saved on the stack in an initial `PUSH {R4-R7,LR}` instruction at the start of the subroutine.

### Academic Integrity Statement [0 ... -100 points]

By typing my name, below, I hereby certify that the work on this prelab is my own and that I have not copied the work of any other student (past or present) while completing it. I understand that if I fail to honor this agreement, I will receive a score of zero for the lab, a one letter drop in my final course grade, and be subject to possible disciplinary action.

Tzu Yu Chen

Save

### Q1 [1 point]

Look at startup/stm32.s in any Standard Peripheral project in System Workbench. The `g_pfnVectors` label indicates the beginning of the exception vector table. A comment above it describes the section information needed to make the linker place it at the proper location in memory.

What is the address range of the STM32 exception vector table? (State it in the form of two hexadecimal numbers like `0xFROM - 0xTO`) Include the reset vector and initial stack pointer value. Include the space up to, and including, the word for `USB IRQHandler`. There are a variety of ways to describe the span of the vector table, so this one is not necessarily the same value you saw in the lecture notes.

`0x00 - 0xC0`

Save

### Q2 [1 point]

Each value label used in a vector table entry in startup/stm32.s is defined further down in the file with a `weak` directive. What does this directive mean? If two different subroutines are defined in a different file with the same name, and one is given a weak designation and the other is given a global designation, what will happen?

`weak is like .global except it's weak so it will only be implemented when there are no other .global symbols of the same name. If two different subroutines are defined with the same name, the other subroutine with the .global designation will override the .weak designation.`

Save

### Q3 [1 point]

Each value label used in a vector table entry in startup/stm32.s is defined further down in the file with a `thumb_set` directive. What does this mean? What would happen if it were omitted?

`This creates a label that is the xxxx_Handler value with the LSB set to 1.`

Save

### Q4 [1 point]

In addition to declaring `it_global`, we recommend, for any label of a subroutine used as an ISR, to also declare it with the following directive:

`.type some_handler, %function`

What does this do? What would happen if it were omitted?

`This tells the assembler that this is a Thumb function.`

Save

### Q5 [1 point]

Why are things like `.thumb_set` and `.type ..., %function` needed? I.e., what happens if you omit them? Try doing so with an ISR.

`I need it so I won't get a hardfault. If I omit them then I will get a hardfault.`

Save

### Q6 [1 point]

For the next few questions, consult the documentation for the SysTick timer (STK) subsystem in section 4.4 of the STM32F0xx Cortex-M0 programming manual. The NVIC and STK subsystems are "core peripherals" and are only fully documented in the programming manual. You will not find much information about them in the Family Reference Manual.

Which bit in which register allows the SysTick down-counter to run?

`0, STK_CSR`

Save

### Q7 [1 point]

How many times per second would the LED toggle if the STK\_RVR held the value 1,499,999, and the CLKSOURCE bit of the STK\_CSR was set to 0?

If this question seems unanswerable, you might have to go digging through the Family Reference Manual to find out how the external clock source is synthesized, or you can read the entirety of the lab document. (You can guess which one will be easier.)

You should verify your answer by trying the code with the ISR you wrote above.

`0.25`

Save

### Q8 [1 point]

What is the address of the STK\_CVR register? (Indicate the absolute address, not the offset from the SysTick register base.)

`0x00`

Save

### Q9 [1 point]

If the CLKSOURCE bit in STK\_CSR is set to zero, the STK\_CVR is currently 4,499,999, and the STK\_RVR is set to 2,999,999, how many seconds will elapse from the moment the counter is started until the SysTick\_Handler ISR is invoked for the first time?

`0.75`

Save

### Q10 [1 point]

How many seconds, at most, can the duration between SysTick\_Handler interrupts be if the CLKSOURCE bit in STK\_CSR is set to 0? (State your answer with four significant digits, i.e., x.xxx)

`16777215`

Save

### Q11 [1 point]

The rate of the LED turning on will be half the toggle rate. How many times per second will the LED in question 10 make a full cycle through on-off?

`2`

Save

### Q12 [1 point]

What value would you use for the STK\_RVR if you wanted the LED in question 10 to turn on at a rate of exactly 6,000 times per second? The toggle rate should be twice that. (And remember that the SysTick counter cycle is one longer than the value in the STK\_RVR.)

`49999`

Save

### Q13 [1 point]

If the CLKSOURCE bit in STK\_CSR is set to zero, the STK\_CVR is currently 4,499,999, and the STK\_RVR is set to 2,999,999, how many seconds will elapse from the moment the counter is started until the SysTick\_Handler ISR is invoked for the first time?

`0.75`

Save

### Q14 [1 point]

How many seconds, at most, can the duration between SysTick\_Handler interrupts be if the CLKSOURCE bit in STK\_CSR is set to 0? (State your answer with four significant digits, i.e., x.xxx)

`16777215`

Save

### Q15 [1 point]

Why is it that we need to enable EXTI interrupt in the NVIC ISER register but not for the SysTick interrupt?

`sysTick is enabled at default and EXTI is not, which is why we need to manually enable the EXTI interrupt`

Save

### Q16 [1 point]

Which register is used to un-mask the EXTI interrupt for a particular pin?

`EXTI_IMR`

Save

### Q17 [1 point]

Name the register that can be used to set the EXTI interrupt to be triggered on a falling edge?

`EXTI_FTSR`

Save

### Q18 [1 point]

Which EXTI interrupt handler do pins 6 and 10 of any port invoke?

`None`

Save

### Q19 [1 point]

What register must you write to change the port used for an EXTI pin 3? (E.g., if you wanted to use PB3 instead of PA3 for an EXTI.)

`SYSCFG_EXTICR1`

Save

### Q20 [1 point]

The subregister for the register you named in the previous question must have a clock enabled for it before it can be modified. This is similar to how you must set the GPIOEN bit in the RCC\_AHBENR to enable the clock to Port C. (Almost every subsystem in the STM32 other than the NVIC and SysTick needs to have a clock enabled before it will do anything.) What bit in what register must you set to 1 to enable that clock before you update the register in the previous question? (Write the symbol name of the register and bit.)

`SYSCFG_APB2ENR`

Save

### Q21 [1 point]

What is the full name for the interrupt service routine that will be invoked for an external interrupt on pin 3 of a GPIO port?

`EXTI_4_IS-IRQn`

Save

### Q22 [1 point]

What is the full name for the interrupt service routine that will be invoked for an external interrupt on pin 5 of a GPIO port?

`EXTI_4_IS-IRQn`

Save

### Q23 [1 point]

What is the interrupt number for the interrupt in the previous question? (See Table 37 on page 217 of the Family Reference Manual.)

`13`

Save

### Q24 [1 point]

What is the default priority value in the NVIC\_IPR for the EXTI2\_3 interrupt? Check what possible values can be written to the NVIC\_IPR fields. What is the default value after reset?

`0`

Save

### Q25 [1 point]

Every interrupt has a distinct *natural secondary* priority that is used to decide which one takes precedence if they all have the same IPR value. What is that priority value for the EXTI2\_3 interrupt compared to any other interrupt with the same value in the NVIC\_IPR? In other words, if every entry in the NVIC\_IPR had the value 0, what priority number would the EXTI2\_3 interrupt have? (See Table 37 of the Family Reference Manual again.)

`13`

Save

$CLR4\_1 \Rightarrow 0x000ff000 \Rightarrow 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$

$OUT04\_9 \Rightarrow 0x00055500 \Rightarrow 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$

$OUT08\_11 \Rightarrow 0x00ff0303 \Rightarrow 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$

$CLR0\_10 \Rightarrow 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$

$OUT08\_10 \Rightarrow 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$

$CLR4\_8 \Rightarrow 0x00003ff000$

$OUT0C4\_8 \Rightarrow 0000\ 0000\ 0000\ 0001\ 0101\ 0101\ 0101\ 0101 \Rightarrow 0x00015555$

$PDO\_3 \Rightarrow 0x1010\ 1010$

$\Rightarrow 0xAA$