

**Pre-lab Assignment 5**

Due: Fri, 18 Feb 2022 11:30:00 (approximately 3 days from the time this page was loaded)  
 [15 points possible] [100 penalties possible] [0 penalties graded so far]

For this lab experiment, you will use assembly language to configure the GPIO ports, build interrupt service routines (ISRs), initialize timers, and handle debouncing of keys in a different way than in lab experiments 3 and 4.

**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

**(1) [1 point]**

What value should be written to a timer's PSC register to configure the prescaler to divide the input clock by 200? Write your answer as a positive decimal integer. (**Note Well:** "200" is not the correct answer.) Understanding this one nuance of timers is a very important detail, so we'll spend a lot of time on it with this pre-lab assignment.

$200-1 = 199$

$$\text{Counter clock frequency} = \frac{f_{\text{clock}}}{\text{PSC}[15:0]+1}$$

**(2) [1 point]**

What is the largest divisor that the timer 6 prescaler can divide the clock by? (And this is also not the largest value you can write to the PSC register.) **Note:** The timer lecture, on slide 7 says one thing, page 539 of the FRM says another. FRM page 540 contradicts page 539. There is only one right answer. That is the only one you will receive credit for. Your instructor reminds you that [this is a course about learning to read \(bad\) documentation](#). Express your answer as a positive decimal integer.

$65536$

$$\text{If you set every 10 sec } (10000) \left( \frac{1}{10} \right) = \frac{48MHz}{48000}$$

$$\text{if & times} \\ \frac{48MHz}{48000} = (ARR+1)(4) \\ ARR+1 = 500 \\ \frac{48MHz}{48000} = (ARR+1)(4)$$

**(3) [1 point]**

Assume that a timer has an input clock of 48MHz, and its prescaler is configured to divide by 4800. What value should be written to the auto-reload register (ARR) cause an update event exactly once every second? (Remember: A value that is evenly divisible by 100 will be an incorrect value.) Express your answer as a positive decimal integer.

$9999$   
 $\text{count freq} = \frac{48MHz}{4800} = 10000$   
 $ARR+1 = 10000$   
 $ARR = 9999$

**(4) [1 point]**

Assume that a timer has an input clock of 48MHz, and its prescaler is configured to divide by 2000. What is the longest update event period, in seconds, that can be set by writing a value into the ARR? State your answer with at least eight decimal places so we can determine if you knew whether you should add or subtract one in the right place.

Let's assume the timer is Timer 6. It has a 16-bit ARR.  
  $23988.00599700$   
 $\text{count freq} = \frac{48MHz}{2000+1} = 23988.00599700$   
 $\text{timer freq} = \frac{\text{count freq}}{ARR+1}$   
 $\text{if } ARR=0 = 23988.00599700$

$$(ARR+1) \cdot 2 = \frac{48MHz}{48000}$$

$$ARR+1 = 500$$

**(5) [1 point]**

The PSC, ARR, and CNT registers for timer 6 are each 16 bits wide. A common mistake is to write a value into one of these registers that is too large. Let's see what happens.

If the value 788572 is written to the PSC, what will it be configured to divide the clock by? Express your answer as a positive decimal number. Remember that the value written to the PSC is not the value that the prescaler divides by. It's still off by one, and we want you to understand -- very well -- which way it's off by one. **Hint:** The easiest way to check the answer to this question is to use the I/O Register debugger to write the value into the PSC, read the value back, and decide if you need to add or subtract one. Note that the value read from the PSC is not the same as the value written to the PSC.

$788573$

**(6) [1 point]**

Which bit of which control register should a '1' be written to enable the (48MHz) system clock for timer 1?

bit:  11 of:  RCC\_APB2ENR

**(7) [1 point]**

Which bit of which control register should a '1' be written to enable the (48MHz) system clock for timer 2?

bit:  0 of:  APB1ENR

**(8) [1 point]**

Suppose you want a timer to produce an update event exactly 113.55705650646448 times per second. If the system clock rate is 48 MHz, the value 928 is written to the PSC register (and the prescalar division will differ from this by one), what value should be written to the ARR register to produce the correct update event frequency?

$454$   
 $\text{count freq} = \frac{48MHz}{928+1} \quad \text{timer freq} = 113.55705646448$   
 $(ARR+1)(\text{timer freq}) = 51668.46071044133$

$454$   
 $\text{timer freq} = \frac{\text{count freq}}{ARR+1}$   
 $ARR = 455 - 1 = 454$

Which bit of the NVIC\_ISER configuration register must a '1' be written to enable the interrupt raised by the update event of timer 15?

$20$

**(10) [1 point]**

Type the exact name of the interrupt service routine for timer 6 when using the STM32 Standard Peripheral firmware. Remember that you should copy and paste the name of the label for the subroutine from startup/startup\_stm32.s.

$TIME6_DAC_IRQHandler$

**(11) [1 point]**

Certain values written to ARR will cause the timer to produce different behavior than may be expected. If a timer is configured so that the value written to the PSC register is 1017 and its ARR register is 0, how many times per second will it generate an update event if the system clock rate is 48 MHz?

$0$   
 $\text{when ARR register is 0, the counter won't run}$   
 $\text{so it would be 0 times/sec}$

**(12) [1 point]**

Which two pins of the keypad are electrically connected when the '4' button is pressed?

Row2  
 Col1

**(13) [1 point]**

For the TDCR1050M 4-digit common-anode 7-segment display, the digits are named, from left-to-right D1, D2, D3, and D4. Which pin numbers are used to illuminate the 'D' segment of digit D1?

Anode:  1  
 Cathode:  3

Save

**(14) [1 point]**

For the 74HC138 3-to-8 decoder with active-low outputs, if 3V is connected to pin 16, 0V is connected to pin 8, and pins 1-6 are wired as follows:

- Pin 1: high
- Pin 2: low
- Pin 3: high
- Pin 4: low
- Pin 5: low
- Pin 6: high

which output pins, if any, are driven low? Chose "none" if none of the output pins are driven low.

$none$

**(15) [1 point]**

For the 74HC138 3-to-8 decoder with active-low outputs, if 3V is connected to pin 16, 0V is connected to pin 8, and pins 1-6 are wired as follows:

- Pin 1: high
- Pin 2: low
- Pin 3: low
- Pin 4: low
- Pin 5: low
- Pin 6: low

which output pins, if any, are driven low? (List them with their "Yx" name or "none" if there are no pins driven low.)

$none$