

Practice Timers

The program below is supposed to toggle bit PA5 at 8 KHz using timer 3 in an interrupt mode. Fill in the missing pieces.

```
//TimerInt.c
/* This program uses Timer 3 in an interrupt mode to toggle bit PA5
   about every 0.125 msec (8KHz). The timer is uses autoreloading.
*/
#include "stm32f407vg.h"
int flag;
int main()
{int tmp;
 //Clock bits
 RCC_AHB1ENR |= _____ //GPIOA clock enable bit
 RCC_APB1ENR |= _____ //Enable peripheral timer for timer 3
 //Interrupt bits
 NVICISER0 |= (1 << 29); //Bit 29 in ISER0 corresponds to int 29 (TIM 3)
 TIM3_DIER |= _____ //Enable Timer 3 update interrupt enable
 TIM3_DIER |= _____ //Enable Timer 3 trigger interrupt enable
 //I/O bits
 //Bits 10-11 = 01 for digital output on PA5 in GPIOA_MODER
 GPIOA_MODER |= _____ //Bits 10-11 to 01 for output on PA5
 //OTYPER register resets to 0 so it is push/pull by default
 GPIOA_OSPEEDER |= _____ //Bits 10-11 = 11 for high speed on PA5
 //PUPDR defaults to no pull up no pull down
 //Timer 2 bits
 TIM3_CR1 |= _____ //Set Auto reload so that it is buffered
 TIM3_PSC = _____ //Don't use prescaling
 TIM3_ARR = _____ //((168 MHz/2)/10500 = 8000 Hz
 TIM3_CR1 |= _____ //Enable Timer 3
 //Main program loop
 tmp = 0;
 while(1)
 {GPIOA_ODR = tmp; //Only PA7 is set up for output so other bits
 tmp = ~tmp; // have no effect.
 flag = 1;
 while(flag == 1);
 TIM3_CR1 |= _____ //Restart timer
 }
 }
void TIM3_IRQHandler()
{flag = 0;
 TIM3_SR &= _____ //Turn off interrupt
}
```