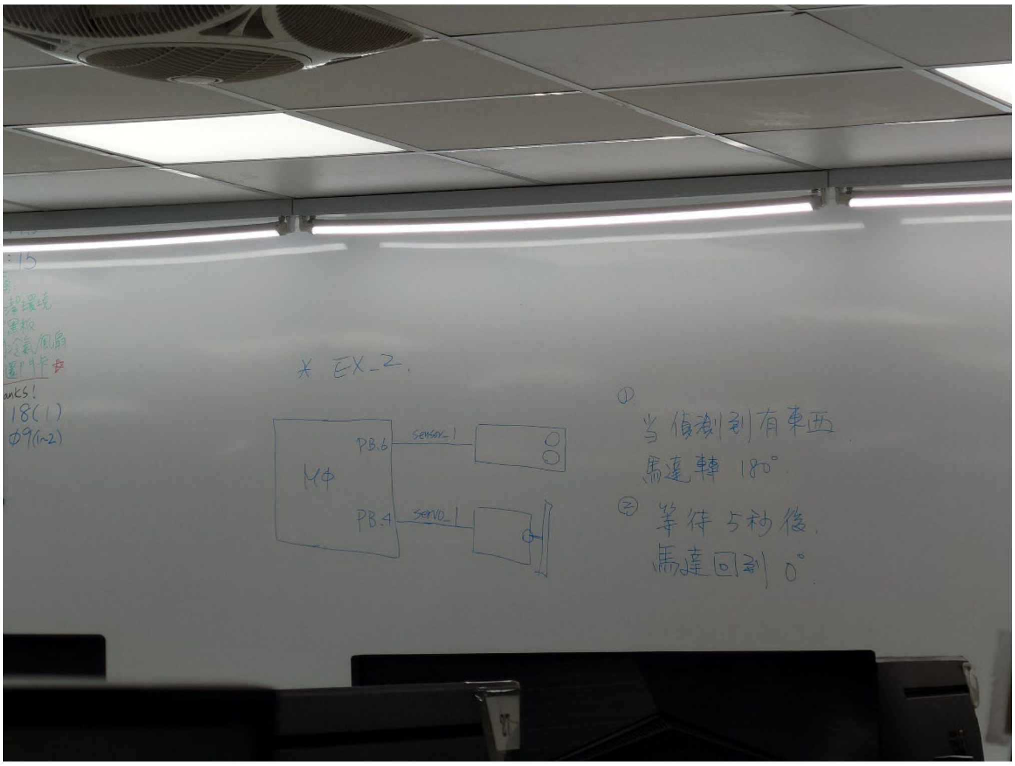
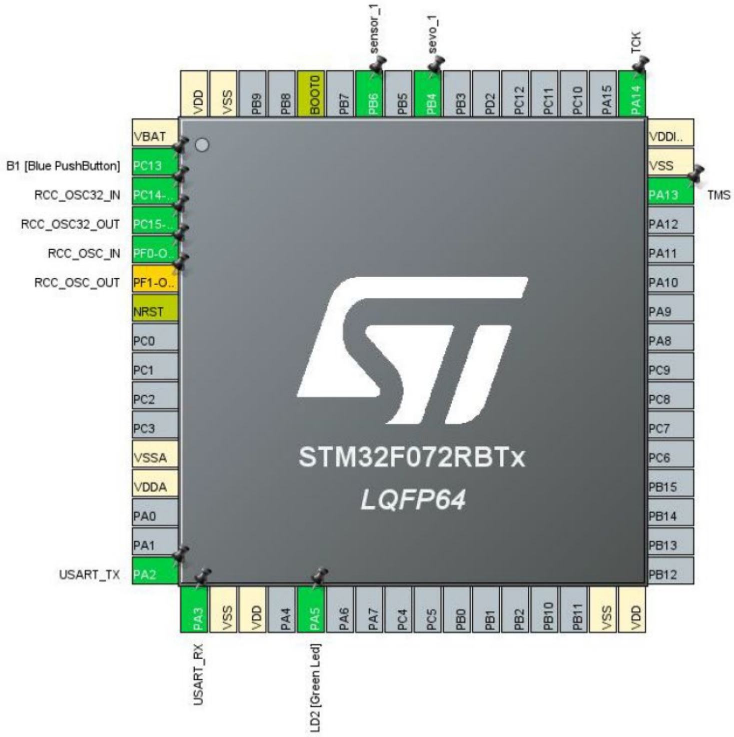
作業 2


* EX_2.
MCU
PB.6
sensor_1
PB.4
servo_1
①
当偵測到有東西
馬達轉 180°.
②
等待5秒後,
馬達回到 0°.


sensor_1
sevo_1
TCK
VDD
VSS
PB9
PB8
BOOT0
PB7
PB6
PB5
PB4
PB3
PD2
PC12
PC11
PC10
PA15
PA14
VBAT
B1 [Blue PushButton]
PC13
RCC_OSC32_IN
PC14-..
RCC_OSC32_OUT
PC15-..
RCC_OSC_IN
PF0-O..
RCC_OSC_OUT
PF1-O..
NRST
PC0
PC1
PC2
PC3
VSSA
VDDA
PA0
PA1
USART_TX
PA2
VDDI..
VSS
PA13
TMS
PA12
PA11
PA10
PA9
PA8
PC9
PC8
PC7
PC6
PB15
PB14
PB13
PB12
PA3
VSS
VDD
PA4
PA5
PA6
PA7
PC4
PC5
PB0
PB1
PB2
PB10
PB11
VSS
VDD
USART_RX
LD2 [Green Led]
STM32F072RBTx
LQFP64

3. Pins Configuration

Pin Number LQFP64	Pin Name (function after reset)	Pin Type	Alternate Function(s)	Label
1	VBAT	Power		
2	PC13	I/O	GPIO_EXTI13	B1 [Blue PushButton]
3	PC14-OSC32_IN	I/O	RCC_OSC32_IN	
4	PC15-OSC32_OUT	I/O	RCC_OSC32_OUT	
5	PF0-OSC_IN	I/O	RCC_OSC_IN	
6	PF1-OSC_OUT *	I/O	RCC_OSC_OUT	
7	NRST	Reset		
12	VSSA	Power		
13	VDDA	Power		
16	PA2	I/O	USART2_TX	USART_TX
17	PA3	I/O	USART2_RX	USART_RX
18	VSS	Power		
19	VDD	Power		
21	PA5 **	I/O	GPIO_Output	LD2 [Green Led]
31	VSS	Power		
32	VDD	Power		
46	PA13	I/O	SYS_SWDIO	TMS
47	VSS	Power		
48	VDDIO2	Power		
49	PA14	I/O	SYS_SWCLK	TCK
56	PB4	I/O	TIM3_CH1	sevo_1
58	PB6	I/O	GPIO_EXTI6	sensor_1
60	BOOT0	Boot		
63	VSS	Power		
64	VDD	Power		

2.3. TIM3

Channel1: PWM Generation CH1

2.3.1. Parameter Settings:

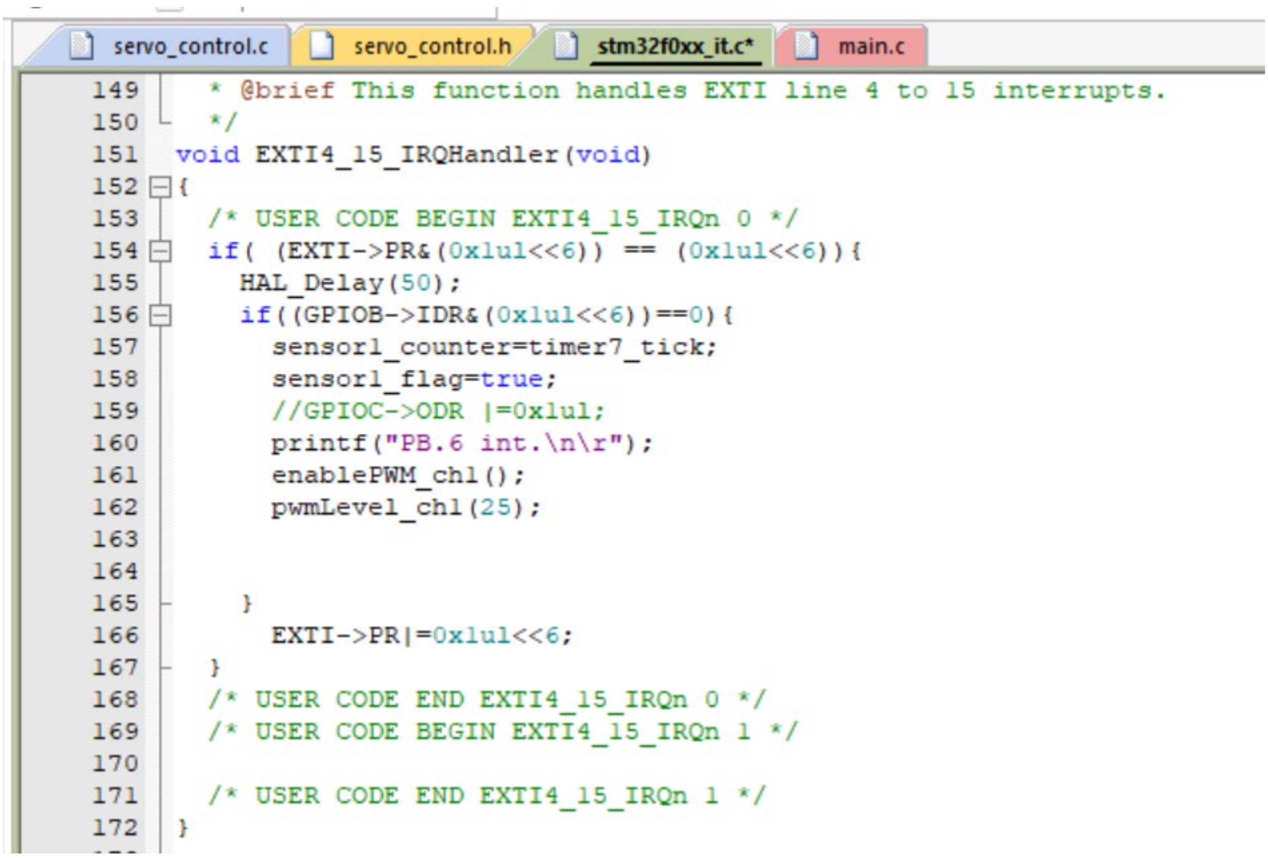
Counter Settings:	
Prescaler (PSC - 16 bits value)	**4800-1** *
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	**400-1** *
Internal Clock Division (CKD)	No Division
auto-reload preload	**Enable** *
Trigger Output (TRGO) Parameters:	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
Clear Input:	
Clear Input Source	Disable
PWM Generation Channel 1:	
Mode	PWM mode 1
Pulse (16 bits value)	0
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High

2.4. TIM7

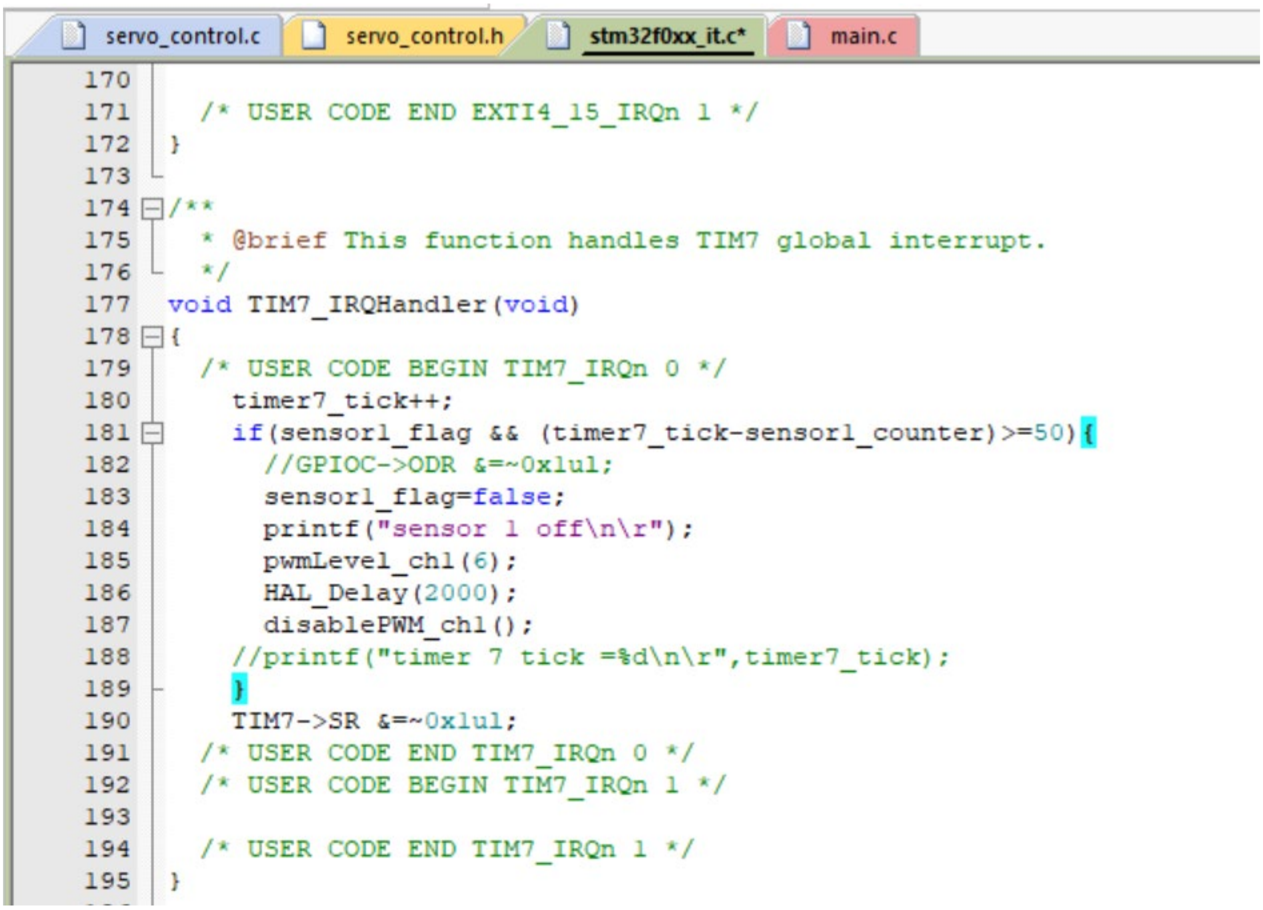
mode: Activated

2.4.1. Parameter Settings:

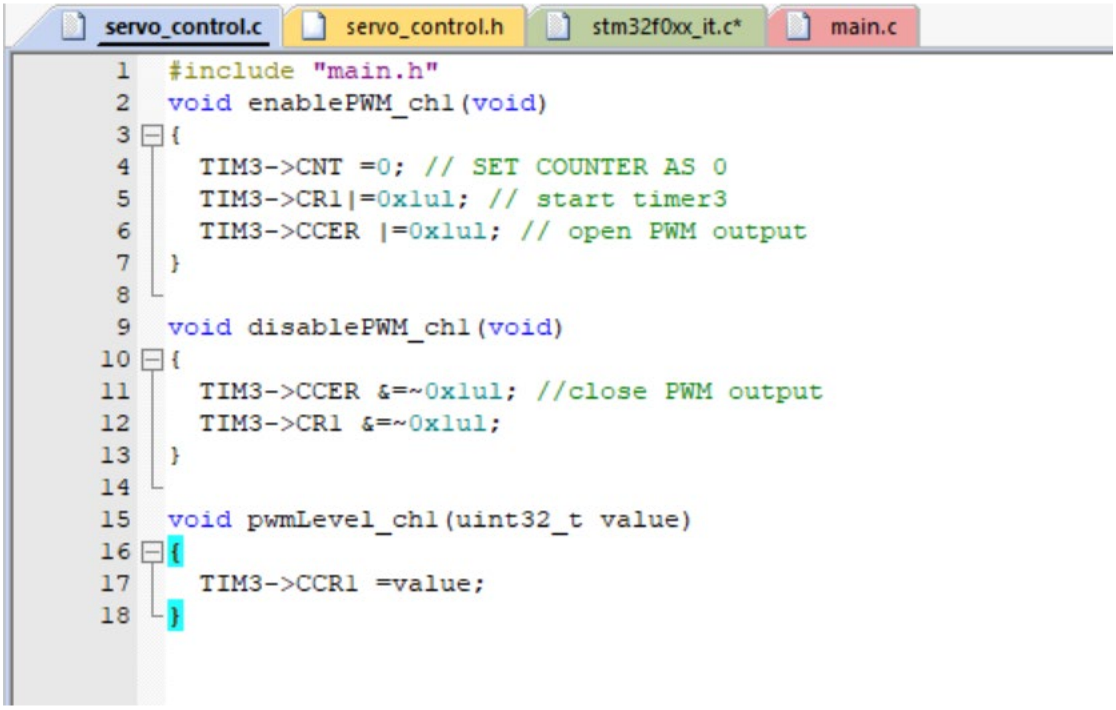
Counter Settings:	
Prescaler (PSC - 16 bits value)	**4800-1** *
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	**1000-1** *
auto-reload preload	**Enable** *
Trigger Output (TRGO) Parameters:	
Trigger Event Selection	Reset (UG bit from TIMx_EGR)



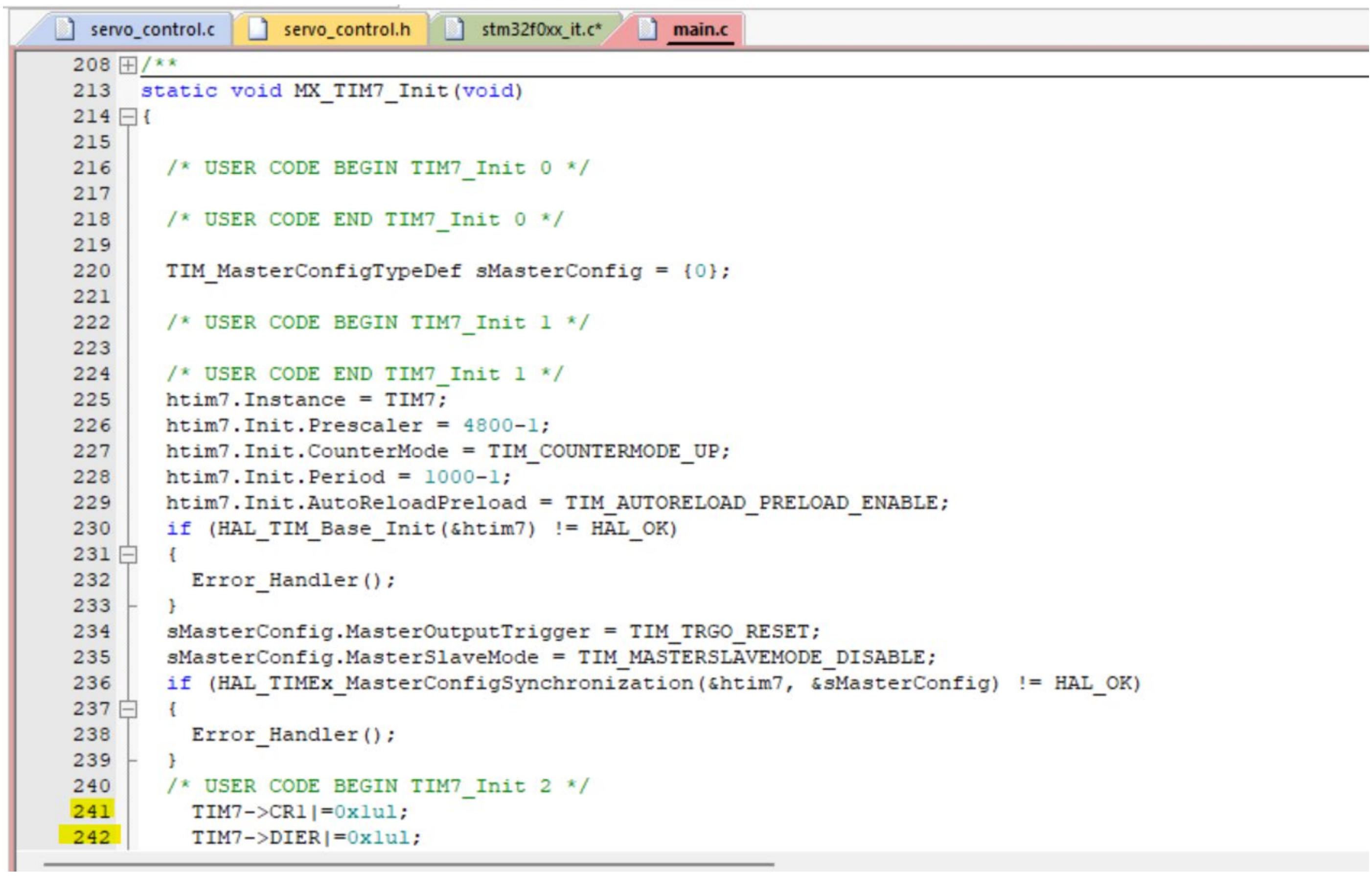
```c
  * @brief This function handles EXTI line 4 to 15 interrupts.
  */
void EXTI4_15_IRQHandler(void)
{
  /* USER CODE BEGIN EXTI4_15_IRQn 0 */
  if( (EXTI->PR&(0x1ul<<6)) == (0x1ul<<6)){
    HAL_Delay(50);
    if((GPIOB->IDR&(0x1ul<<6))==0){
      sensor1_counter=timer7_tick;
      sensor1_flag=true;
      //GPIOC->ODR |=0x1ul;
      printf("PB.6 int.\n\r");
      enablePWM_ch1();
      pwmLevel_ch1(25);


    }
      EXTI->PR|=0x1ul<<6;
  }
  /* USER CODE END EXTI4_15_IRQn 0 */
  /* USER CODE BEGIN EXTI4_15_IRQn 1 */

  /* USER CODE END EXTI4_15_IRQn 1 */
}
```



```c

  /* USER CODE END EXTI4_15_IRQn 1 */
}

/**
  * @brief This function handles TIM7 global interrupt.
  */
void TIM7_IRQHandler(void)
{
  /* USER CODE BEGIN TIM7_IRQn 0 */
    timer7_tick++;
    if(sensor1_flag && (timer7_tick-sensor1_counter)>=50){
      //GPIOC->ODR &=~0x1ul;
      sensor1_flag=false;
      printf("sensor 1 off\n\r");
      pwmLevel_ch1(6);
      HAL_Delay(2000);
      disablePWM_ch1();
    //printf("timer 7 tick =%d\n\r",timer7_tick);
    }
    TIM7->SR &=~0x1ul;
  /* USER CODE END TIM7_IRQn 0 */
  /* USER CODE BEGIN TIM7_IRQn 1 */

  /* USER CODE END TIM7_IRQn 1 */
}
```



```c
#include "main.h"
void enablePWM_ch1(void)
{
  TIM3->CNT =0; // SET COUNTER AS 0
  TIM3->CR1|=0x1ul; // start timer3
  TIM3->CCER |=0x1ul; // open PWM output
}

void disablePWM_ch1(void)
{
  TIM3->CCER &=~0x1ul; //close PWM output
  TIM3->CR1 &=~0x1ul;
}

void pwmLevel_ch1(uint32_t value)
{
  TIM3->CCR1 =value;
}
```



```c
/**
static void MX_TIM7_Init(void)
{

  /* USER CODE BEGIN TIM7_Init 0 */

  /* USER CODE END TIM7_Init 0 */

  TIM_MasterConfigTypeDef sMasterConfig = {0};

  /* USER CODE BEGIN TIM7_Init 1 */

  /* USER CODE END TIM7_Init 1 */
  htim7.Instance = TIM7;
  htim7.Init.Prescaler = 4800-1;
  htim7.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim7.Init.Period = 1000-1;
  htim7.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
  if (HAL_TIM_Base_Init(&htim7) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim7, &sMasterConfig) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN TIM7_Init 2 */
    TIM7->CR1|=0x1ul;
    TIM7->DIER|=0x1ul;
```