



# TIM

## 目录

使用 STM32CubeMX 配置 TIM 定时器 .....	2
初始化代码.....	3
常用 TIM 相关操作函数 stm32f4xx_hal_tim.c.....	4
1.1 开始定时器中断 HAL_TIM_Base_Start_IT.....	4
1.2 停止定时器中断 HAL_TIM_Base_Stop_IT .....	5
1.4 定时器中断回调函数 HAL_TIM_PeriodElapsedCallback.....	5
1.5 定时器设置计数值函数 __HAL_TIM_SET_COUNTER .....	6





## 初始化代码

在 tim.c 源文件中，有如下配置代码

```
1. void MX_TIM2_Init(void)
2. {
3.
4. /* USER CODE BEGIN TIM2_Init 0 */
5.
6. /* USER CODE END TIM2_Init 0 */
7.
8. TIM_ClockConfigTypeDef sClockSourceConfig = {0};
9. TIM_MasterConfigTypeDef sMasterConfig = {0};
10.
11. /* USER CODE BEGIN TIM2_Init 1 */
12.
13. /* USER CODE END TIM2_Init 1 */
14. htim2.Instance = TIM2;
15. htim2.Init.Prescaler = 8400-1;
16. htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
17. htim2.Init.Period = 10000-1;
18. htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
19. htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
20. if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
21. {
22.     Error_Handler();
23. }
24. sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
25. if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
26. {
27.     Error_Handler();
28. }
29. sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
30. sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
31. if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
32. {
33.     Error_Handler();
34. }
35. /* USER CODE BEGIN TIM2_Init 2 */
36.
37. /* USER CODE END TIM2_Init 2 */
38. }
```



## 常用 TIM 相关操作函数 stm32f4xx\_hal\_tim.c

### 1.1 开始定时器中断 HAL\_TIM\_Base\_Start\_IT

```
1. HAL_StatusTypeDef HAL_TIM_Base_Start_IT(TIM_HandleTypeDef *htim)
2. {
3.     uint32_t tmpsmcr;
4.
5.     /* Check the parameters */
6.     assert_param(IS_TIM_INSTANCE(htim->Instance));
7.
8.     /* Check the TIM state */
9.     if (htim->State != HAL_TIM_STATE_READY)
10.    {
11.        return HAL_ERROR;
12.    }
13.
14.    /* Set the TIM state */
15.    htim->State = HAL_TIM_STATE_BUSY;
16.
17.    /* Enable the TIM Update interrupt */
18.    __HAL_TIM_ENABLE_IT(htim, TIM_IT_UPDATE);
19.
20.    /* Enable the Peripheral, except in trigger mode where enable is automatical
        ly done with trigger */
21.    if (IS_TIM_SLAVE_INSTANCE(htim->Instance))
22.    {
23.        tmpsmcr = htim->Instance->SMCR & TIM_SMCR_SMS;
24.        if (!IS_TIM_SLAVEMODE_TRIGGER_ENABLED(tmpsmcr))
25.        {
26.            __HAL_TIM_ENABLE(htim);
27.        }
28.    }
29.    else
30.    {
31.        __HAL_TIM_ENABLE(htim);
32.    }
33.
34.    /* Return function status */
35.    return HAL_OK;
36. }
```

入口参数：TIM 句柄



返回值: HAL\_OK(成功) or HAL\_ERROR(失败)

使用示例:

```
1. HAL_TIM_Base_Start_IT(&htim2);
```

## 1.2 停止定时器中断 HAL\_TIM\_Base\_Stop\_IT

```
1. HAL_StatusTypeDef HAL_TIM_Base_Stop_IT(TIM_HandleTypeDef *htim)
2. {
3.     /* Check the parameters */
4.     assert_param(IS_TIM_INSTANCE(htim->Instance));
5.
6.     /* Disable the TIM Update interrupt */
7.     __HAL_TIM_DISABLE_IT(htim, TIM_IT_UPDATE);
8.
9.     /* Disable the Peripheral */
10.    __HAL_TIM_DISABLE(htim);
11.
12.    /* Set the TIM state */
13.    htim->State = HAL_TIM_STATE_READY;
14.
15.    /* Return function status */
16.    return HAL_OK;
17. }
```

入口参数: TIM 句柄

返回值: HAL\_OK(成功)

使用示例:

```
1. HAL_TIM_Base_Stop_IT(&htim2);
```

## 1.4 定时器中断回调函数 HAL\_TIM\_PeriodElapsedCallback

在配置工程时, 因使用了 TIM14 作为系统时钟源, TIM 的回调函数已被重写至 main.c 源文件中

```
1. void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
2. {
3.     /* USER CODE BEGIN Callback 0 */
4.
5.     /* USER CODE END Callback 0 */
6.     if (htim->Instance == TIM14) {
7.         HAL_IncTick();
8.     }
```



```
9.  /* USER CODE BEGIN Callback 1 */
10. if(htim->Instance == TIM2)
11. {
12.     if(__HAL_TIM_GET_FLAG(&htim2,TIM_FLAG_UPDATE) != RESET)
13.     {
14.         //用户代码
15.     }
16. }
17. /* USER CODE END Callback 1 */
18. }
```

>判断是否是 TIM2 发生了中断

>判断是否是 TIM2 的溢出中断

>用户代码

## 1.5 定时器设置计数值函数\_\_HAL\_TIM\_SET\_COUNTER

在 HAL 库中，用户可操作的外设配置函数，都以\_\_双下划线开始命名  
此函数直接对 TIM 的 CNT 计数寄存器进行配置

```
1. #define __HAL_TIM_SET_COUNTER(__HANDLE__, __COUNTER__) ((__HANDLE__)->Instance->CNT = (__COUNTER__))
```

入口参数：TIM 句柄、计数值

返回值：无

使用示例：

```
1. __HAL_TIM_SET_COUNTER(&htim2,1000);
```

21-MAR-2021

厦大嘉庚 TCR 嵌入式