

晟矽微电 应用笔记

MS32F031

Timer1 CCxIF

AN22007

V1.1





目 录

1	适用范围	1
2	Timer1 输出比较 CCxIF	1
2.1	TIM1_RCR	1
2.1.1	理论分析	1
2.1.2	例程测试	1
2.1.3	分析及运行	1
2.2	CCxIF 置位条件	2
2.2.1	CCxIF (X:1, 2, 3)	2
2.2.2	CC4IF	3
2.2.3	中断服务函数建议	4
2.3	Timer1_CH4 PWM 输出	5
3	修订记录	6
4	免责声明	7

Sinomcu.com



1 适用范围

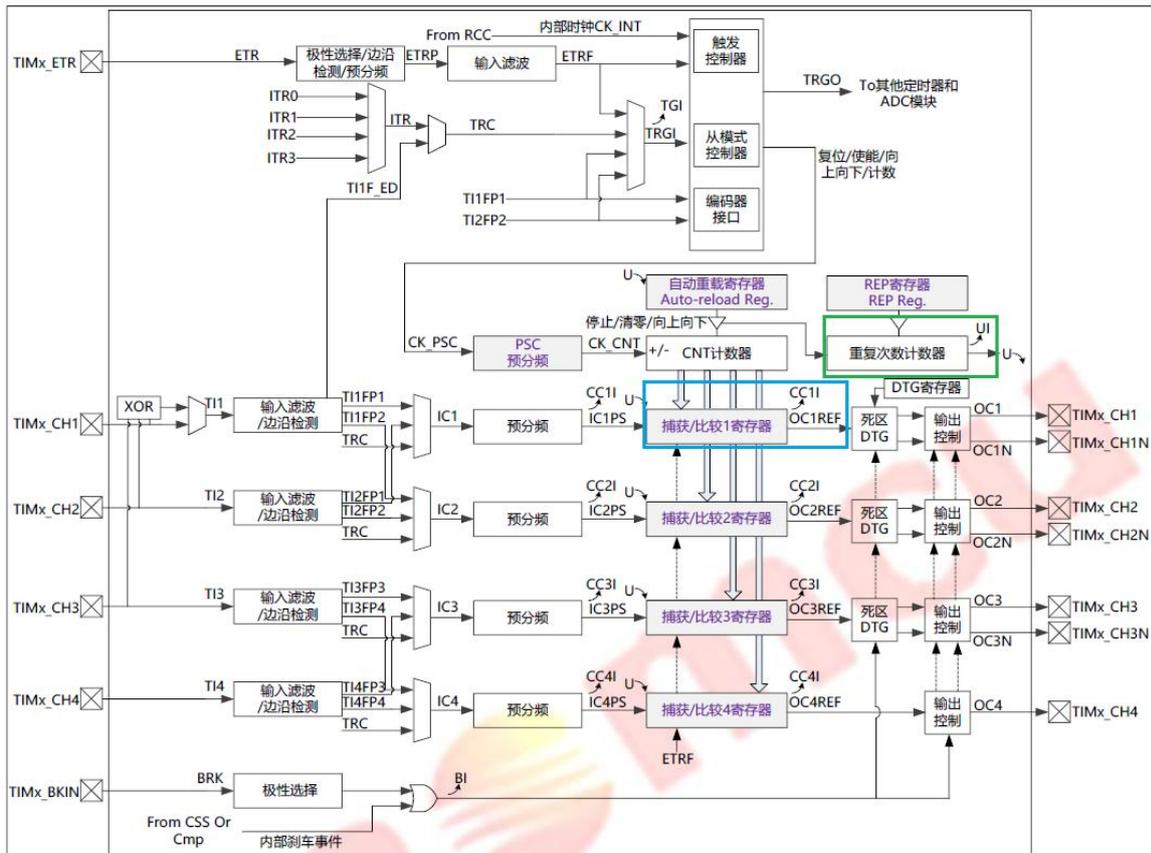
本文档适用于 MS32F031A6, Timer1 输出比较 CCxIF 使用参考。
基于 MS32F031A6 EV Board V1.1(2021-11-25)测试。

2 Timer1 输出比较 CCxIF

2.1 TIM1_RCR

2.1.1 理论分析

CCxIF (x: 1、2、3、4) 与 TIM1_RCR (重复计数寄存器) 值无关, TIM1_RCR 值影响 UIF;
MS32F031A6 用户手册 “13.3 功能描述章节” 中框图如下:



2.1.2 例程测试

附件例程解压后放在 MS32F0x1_Periph_Lib_Example\proj\MS32F031_EV\TIMER 目录下。

proj > MS32F031_EV > TIMER > TIM1_CCxIF >

名称	修改日期
Keil_Project	2022/4/6 17:30
startup	2022/1/7 19:55
system	2022/1/7 19:55
user	2022/1/7 19:55
readme.txt	2022/4/6 17:30

2.1.3 分析及运行

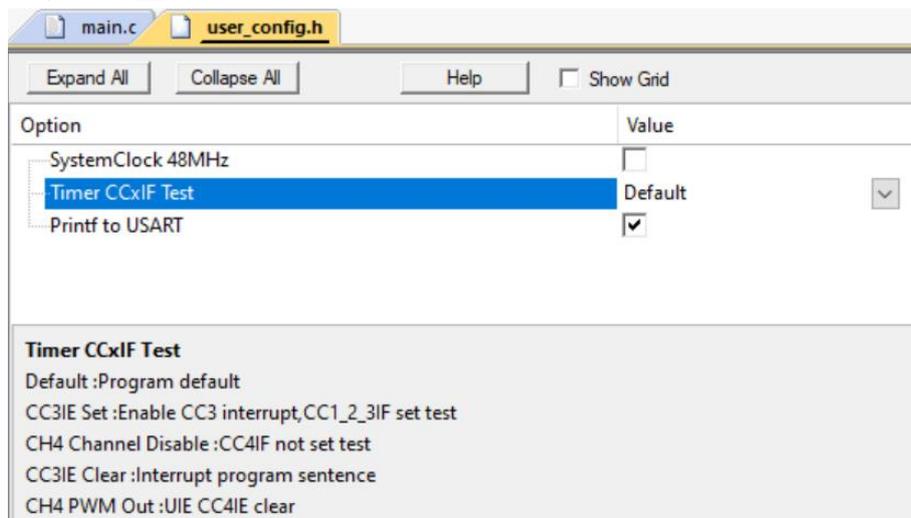
例程 Timer 向上计数 100ms, TIM1_RCR=4, 向上计数溢出 5 次 UIF 置位; CHANNEL_CH4 占空比设置为 25%; CC4IF 置位 5 次 UIF 置位 1 次; 程序中允许 UI 及 CC4I 中断。



```
59 // count up,PWM period 10Hz
60 MS32_TIM_StructInit(&TimerInitStruct);
61 TimerInitStruct.CounterMode = MS32_TIM_COUNTERMODE_UP;
62 TimerInitStruct.Prescaler = __MS32_TIM_CALC_PSC(SystemCoreClock,1000); // unit Hz,div sysclk
63 TimerInitStruct.AutoReload = __MS32_TIM_CALC_ARR(SystemCoreClock,TimerInitStruct.Prescaler,10); //
64 TimerInitStruct.RepetitionCounter = 4; // 500ms/100ms=5,5-1

69 MS32_TIM_OC_StructInit(&TimerOCInitStruct);
70 TimerOCInitStruct.OCMode = MS32_TIM_OCMODE_PWM1; // PWM1 mode
71 TimerOCInitStruct.CompareValue = (( MS32_TIM_GetAutoReload(TIM1) + 1 ) * 25 ) / 100);
72 MS32_TIM_OC_Init(TIM1, MS32_TIM_CHANNEL_CH4, &TimerOCInitStruct);
```

使用 Keil MDK 打开工程，打开文件 user_config.h 文件，借助 Configuration Wizard ，参考下图确认程序默认配置：



编译运行，EV 串口输出信息如下：

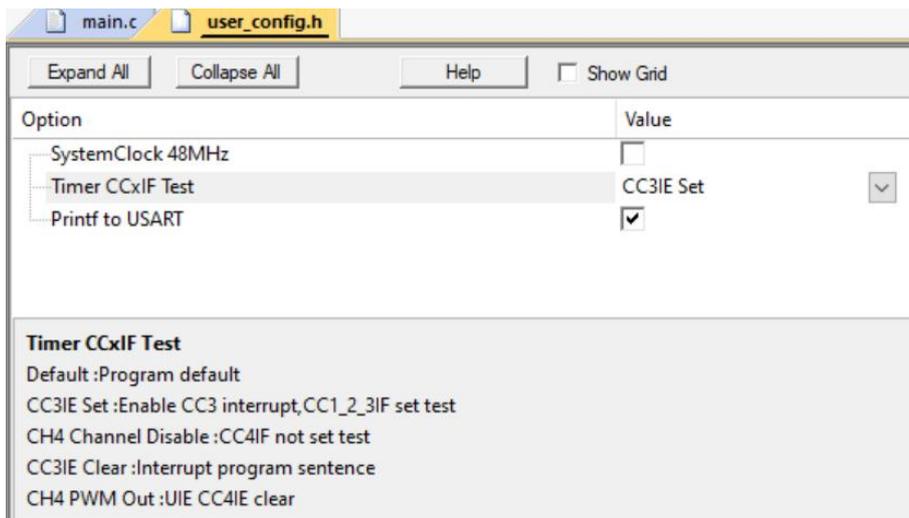
```
*****Timer1 CCxIF Example*****
--Inf: System Core Clock 8000000 Hz.
--Inf: Timer1 CC4IF set.
--Inf: Timer1 UIF set.
--Inf: running count:1
--Inf: Timer1 CC4IF set.
--Inf: Timer1 UIF set.
--Inf: running count:2
```

2.2 CCxIF 置位条件

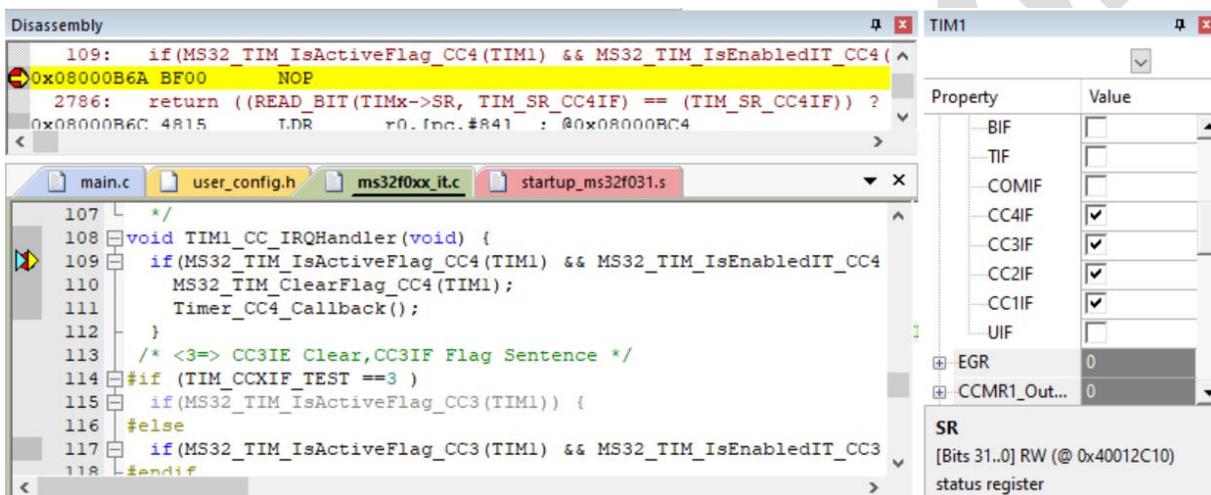
2.2.1 CCxIF (X:1, 2, 3)

CCxIF(X:1, 2, 3)每次计数过程中满足比较条件即置位；使能对应 CCxIE 后产生中断。

打开文件 user_config.h 文件，借助 Configuration Wizard，参考下图修改程序配置：



编译后，调试，参考下图设置断点，运行结果如下：

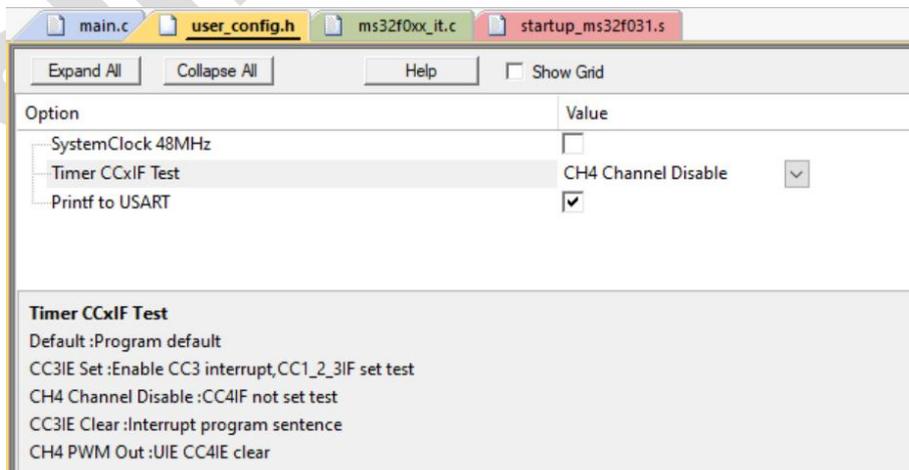


2.2.2 CC4IF

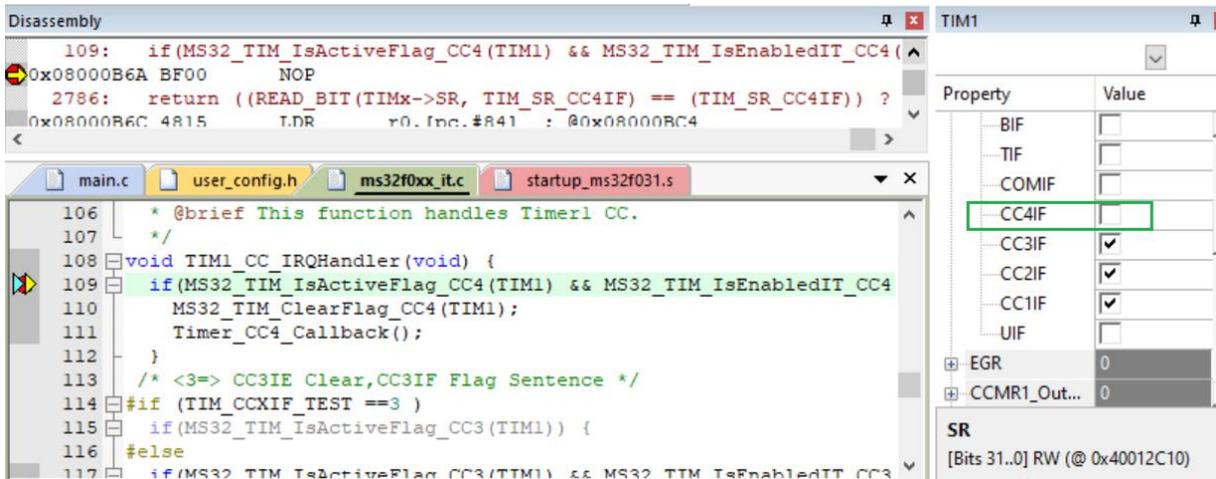
CC4IF 置位需要将通道配置为输出(例程默认)；

MS32_TIM_CC_EnableChannel(TIM1, MS32_TIM_CHANNEL_CH4)；

反例验证，借助 Configuration Wizard，参考下图修改程序配置：



编译后，调试参考下图设置断点，运行如下图：



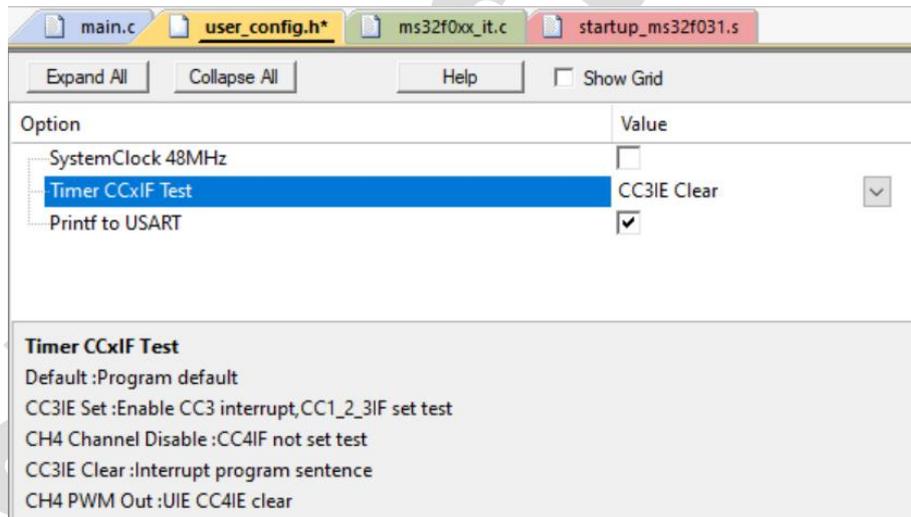
2.2.3 中断服务函数建议

中断服务函数建议使用如下语句，x=1、2、3、4。

`MS32_TIM_IsActiveFlag_CCx(TIM1) && MS32_TIM_IsEnabledIT_CCx(TIM1)`

```
108 void TIM1_CC_IRQHandler(void) {
109     if (MS32_TIM_IsActiveFlag_CC4(TIM1) && MS32_TIM_IsEnabledIT_CC4(TIM1)) {
110         MS32_TIM_ClearFlag_CC4(TIM1);
111         Timer_CC4_Callback();
112     }
}
```

不使能 CC3IE，使用 `if (MS32_TIM_IsActiveFlag_CC3(TIM1))` 语句判断：借助 Configuration Wizard，参考下图修改程序配置：



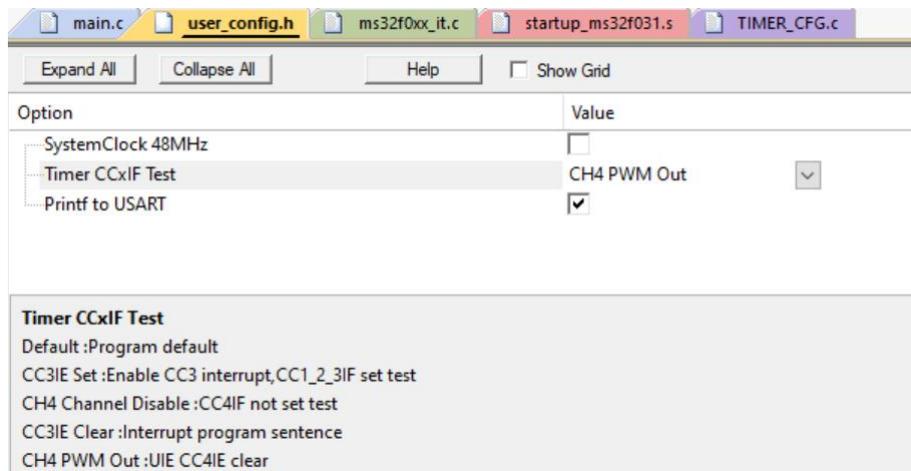
编译下载运行，串口输出信息：

```
*****Timer1 CCxIF Example*****
-Inf: System Core Clock 8000000 Hz.
-Inf: Timer1 CC4IF set.
-Inf: Timer1 CC4IF set., CC3IF set.
-Inf: Timer1 UIF set.
-Inf: running count:1, CC3IF set.
-Inf: Timer1 CC4IF set.
-Inf: Timer1 CC4IF set., CC3IF set.
-Inf: Timer1 UIF set.
```



2.3 Timer1_CH4 PWM 输出

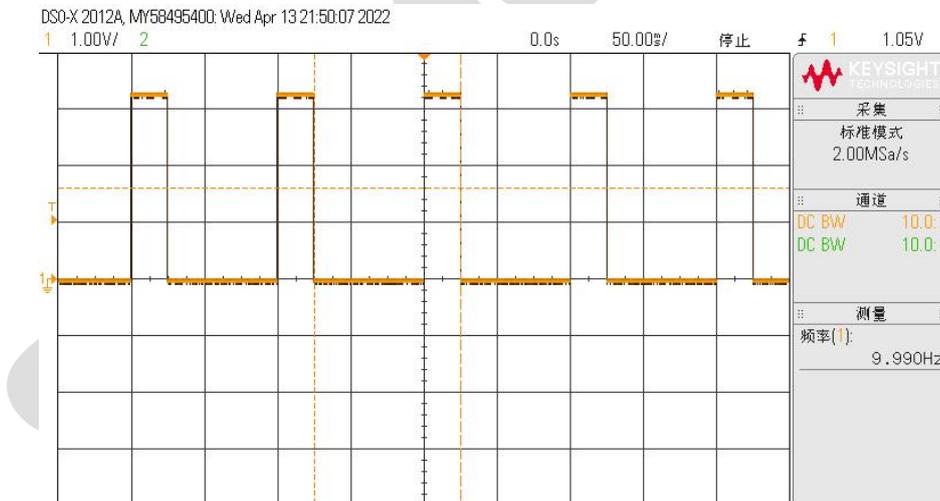
输出 CH4 PWM 信号, 可不使能 UI 及 CCxI 中断; 借助 Configuration Wizard, 参考下图修改程序配置:



编译、下载运行, 串口输出信息:

```
*****Timer1 CCxIF Example*****  
-Inf: System Core Clock 8000000 Hz.  
-Inf: running count:1  
-Inf: running count:2  
-Inf: running count:3  
-Inf: running count:4  
-Inf: running count:5  
-Inf: running count:6
```

示波器测量 PA11 (TIM1_CH4) 信号应为 10Hz, 占空比为 25% 的 PWM 波, 实测波形如下:





3 修订记录

版本	修订日期	修订内容
V1.0	2022-08-26	1359, 初版。
V1.1	2022-11-02	1359, 修订: 1) 附件例程改为使用 user_config.h 文件进行配置; 2) 依据例程修改 2.2、2.3 章节程序配置方法; 3) 增加第 4 章节。

sinomcu.com



4 免责声明

本资料为晟矽微电子（以下简称“我司”）版权所有。

我司将力求资料内容准确无误，同时保留在不通知用户的情况下，对本资料内容的修改权。如您需要获得最新的资料，请及时联系我司。

Sinomcu.com