



链滴

STM32 WWDG 窗口看门狗

作者: [Today](#)

原文链接: <https://ld246.com/article/1528525636564>

来源网站: [链滴](#)

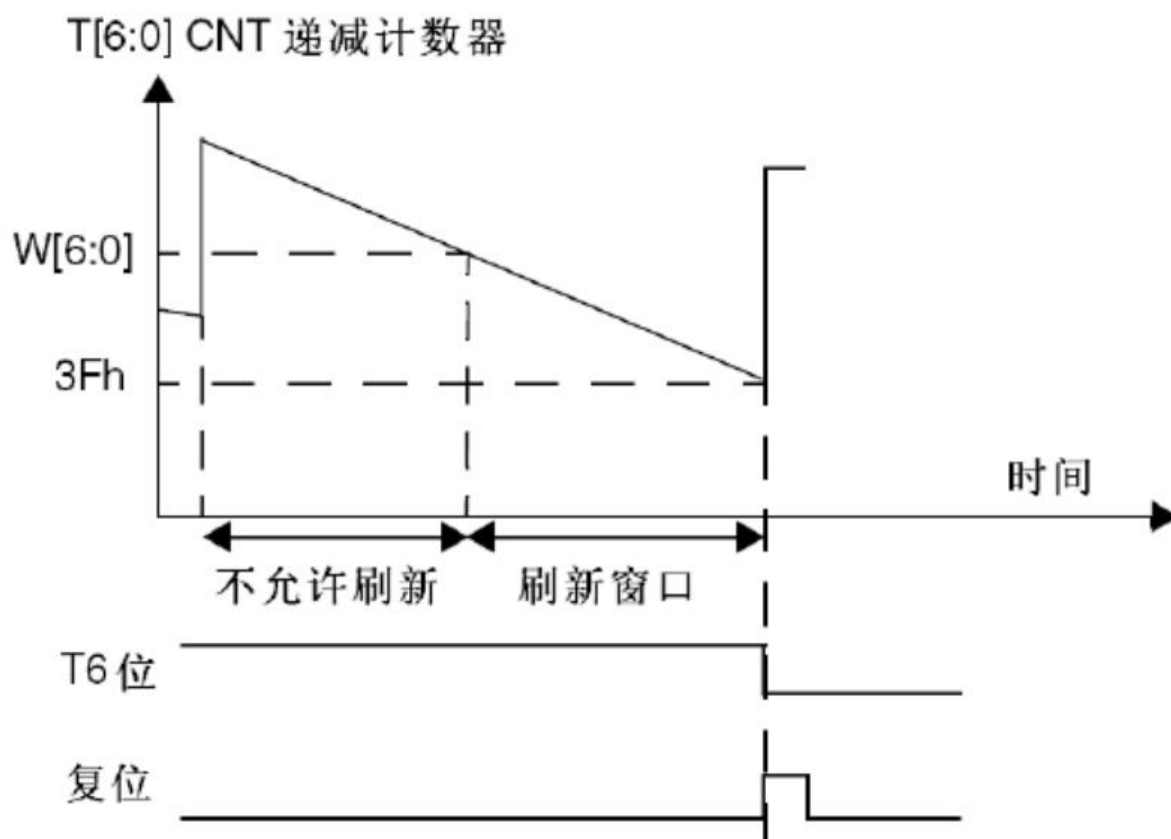
许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



STM32 窗口看门狗简介

窗口看门狗 (WWDG) 通常被用来监测由外部干扰或不可预见的逻辑条件造成的应用程序正常的运序列而产生的软件故障。除非递减计数器的值在WWDG->CR 的第六位变成 0 前被刷新, 看门狗电在达到预置的时间周期时, 会产生一个 MCU 复位。在递减计数达到窗口配置寄存器(WWDG->CFR) 值之前, 如果 7 位的递减计数器数值(在控制寄存器中)被刷新, 那么也将产生一个 MCU 复位。这表递减计数器需要在一个有限的时间窗口中被刷新。

STM32有两个看门狗, 一个是独立看门狗 (IWDG) 之前写过《[STM32 IWDG独立看门狗](#)》, 另一是窗口看门狗 (WWDG)。我们知道独立看门狗的工作原理就是一个键值寄存器 (IWDG_KR) 低16 寄存器。软件必须每隔一定时间写入0xAAAA,否则, 当计数器减到0时, 程序会自动复位。窗口看门跟独立看门狗一样, 也是一个递减计数器不断的往下递减计数, 当减到一个固定值 0X40 时还不喂狗话, 就会产生复位, 这个值叫窗口的下限, 是固定的值, 不能改变。这个是跟独立看门狗类似的地方不同的地方是窗口看门狗的计数器的值在减到某一个数之前喂狗的话也会产生复位, 这个值叫窗口的限, 上限值由用户设置。窗口看门狗计数器的值必须在上窗口和下窗口之间才可以喂狗, 这就是窗口门狗中窗口两个字的含义。



上窗口值 (W[6:0]) 是由用户自己设定的, 根据实际要求来设计窗口值, 但是一定要确保窗口值大于 X40, 否则窗口就不存在了。

窗口看门狗的超时公式

$$TWWDG = (4096 \times 2^{WDGTB} \times (T[5:0] + 1)) / FPCLK1;$$

TWWDG: WWDG 超时时间 (单位为 ms)

FPCLK1: APB1 的时钟频率 (单位为 Khz)

WDGTB: WWDG 的预分频系数

T[5:0]: 窗口看门狗的计数器低 6 位

现在假设 FPCLK1 = 36 MHZ, 那么就可以得到最小到最大超时时间表如下:

| WDGTB 时间 | 最小超时时间 | 最大超 |
|-------------|--------|---------|
| 0 | 113微秒 | 7.28毫秒 |
| 1 | 227微秒 | 14.56毫秒 |
| 2 | 455微秒 | 29.12毫秒 |
| 3 | 910微秒 | 58.25毫秒 |

窗口看门狗3个重要的寄存器

WWDG的第一个寄存器是控制寄存器 (WWDG_CR)



它只有低八位有效，T[6: 0]用来存储看门狗的计数器值，随时更新的，每个窗口看门狗计数周期（ $496 \times 2^{\text{WDGTB}}$ ）减1。当该计数器的值从0X40变为0X3F的时候，将产生看门狗复位。WDGA则是看门狗的激活位，该位由软件置1，以启动看门狗，并且一定要注意的是该位一旦设置，就只能硬件复位后才能清零了。

WWDG的第二个寄存器是配置寄存器（WWDG_CFR）



| | |
|-------|---|
| 位31:8 | 保留。 |
| 位9 | EWI : 提前唤醒中断 此位若置1，则当计数器值达到40h，即产生中断。 此中断只能由硬件在复位后清除。 |
| 位8:7 | WDGTB[1:0] : 时基 预分频器的时基可根据如下修改： 00: CK计时器时钟(PCLK1除以4096)除以1 01: CK计时器时钟(PCLK1除以4096)除以2 10: CK计时器时钟(PCLK1除以4096)除以4 11: CK计时器时钟(PCLK1除以4096)除以8 |
| 位6:0 | W[6:0] : 7位窗口值 这些位包含了用来与递减计数器进行比较用的窗口值。 |

WWDG_CFR中的EWI是提前唤醒中断，也就是在快要产生复位的前一段时间（T[6:0]=0X40）来提醒我们，需要进行喂狗了，否则将复位！因此，我们一般用该位来设置中断，当窗口看门狗的计数器值减到0X40的时候，如果该位设置，并开启了中断，则会产生中断，我们可以在中断里面向WWDG_CR重新写入计数器的值，来达到喂狗的目的。注意这里在进入中断后，必须在小于1个窗口看门狗计数周期的时间（在PCLK1频率为36M且WDGTB为0的条件下，该时间为113μs）内重新写WWDG_CR，否则，看门狗将产生复位。

WWDG的最后一个寄存器是状态寄存器（WWDG_SR），该寄存器是用来记录当前是否有唤醒的标志。该寄存器仅有第0位（从0开始计算）有效，其他都是保留位，当计数器值达到0x40时，此位由硬件置1。它必须通过软件写0来清除。对此位写1无效。即使中断未被使能，在计数器的值达到0X40的时候，此位也会被置1。

在介绍完了窗口看门狗的寄存器之后，我们介绍要如何启用STM32的窗口看门狗。这里我们介绍库函数中用中断的方式来喂狗的方法，WWDG库函数相关源码和定义分布在源文件stm32f10x_wwdg.c和头文件stm32f10x_wwdg.h

一般步骤

使能 WWDG 时钟

设置窗口值和分频数

开启 WWDG 中断并分组

设置计数器初始值并使能看门狗

编写中断服务函数

硬件设计

指示灯 DS0和DS1，通过 DS0 和 DS1 来指示 STM32 的复位情况和窗口看门狗的喂狗情况。

软件设计



wwdg.h



```
#ifndef __WDG_H
#define __WDG_H
```

```
#include <stm32f10x.h>
```

```
void WWDG_Init(u8 tr,u8 wr,u32 fprer); //初始化WWDG
void WWDG_Set_Counter(u8 cnt);        //设置WWDG的计数器
```



```
void WWDG_NVIC_Init(void);
```

```
#endif
```

```
wwdg.c
```

```
@wwdg.c
10
11 void WWDG_Init(u8 tr, u8 wr, u32 fprer) {
12
13     RCC_APB1PeriphClockCmd(RCC_APB1Periph_WWDG, ENABLE); //WWDG时钟使能
14
15     WWDG_SetPrescaler(fprer); //设置IWDG预分频值
16     WWDG_SetWindowValue(wr); //设置窗口值
17     WWDG_Enable(tr); //使能看门狗, 设置 counter
18     WWDG_ClearFlag();
19     WWDG_NVIC_Init(); //初始化窗口看门狗 NVIC
20     WWDG_EnableIT(); //开启窗口看门狗中断
21 }
22
23 //窗口看门狗中断服务程序
24 void WWDG_NVIC_Init() {
25
26     NVIC_InitTypeDef NVIC_InitTypeDef;
27
28     NVIC_InitTypeDef.NVIC_IRQChannel = WWDG_IRQn; //WWDG中断
29     NVIC_InitTypeDef.NVIC_IRQChannelPreemptionPriority = 2; //抢占2, 子优先级3, 组2
30     NVIC_InitTypeDef.NVIC_IRQChannelSubPriority = 3; //抢占2, 子优先级3, 组2
31     NVIC_InitTypeDef.NVIC_IRQChannelCmd = ENABLE;
32
33     NVIC_Init(&NVIC_InitTypeDef); //NVIC初始化
34
35 }
36
```

```
#include "wwdg.h"
#include <misc.h>
#include <stm32f10x_rcc.h>
#include <stm32f10x_wwdg.h>
#include "../led/led.h"
```

```
u8 WWDG_CNT = 0x7f;
```

```
void WWDG_Init(u8 tr, u8 wr, u32 fprer) {
```

```
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_WWDG, ENABLE); //WWDG时钟使能
```

```
    WWDG_SetPrescaler(fprer); //设置IWDG预分频值
    WWDG_SetWindowValue(wr); //设置窗口值
    WWDG_Enable(tr); //使能看门狗,设置 counter.
    WWDG_ClearFlag();
    WWDG_NVIC_Init(); //初始化窗口看门狗 NVIC
    WWDG_EnableIT(); //开启窗口看门狗中断
}
```

```
//窗口看门狗中断服务程序
```

```
void WWDG_NVIC_Init() {
```

```
    NVIC_InitTypeDef NVIC_InitTypeDef;
    NVIC_InitTypeDef.NVIC_IRQChannel = WWDG_IRQn; //WWDG中断
    NVIC_InitTypeDef.NVIC_IRQChannelPreemptionPriority = 2; //抢占2, 子优先级3, 组2
    NVIC_InitTypeDef.NVIC_IRQChannelSubPriority = 3; //抢占2, 子优先级3, 组2
    NVIC_InitTypeDef.NVIC_IRQChannelCmd = ENABLE;
```

```

    NVIC_Init(&NVIC_InitTypeDef);           //NVIC初始化
}
void WWDG_IRQHandler(void) {
    WWDG_SetCounter(0x7F); //当禁掉此句后,窗口看门狗将产生复位
    WWDG_ClearFlag(); //清除提前唤醒中断标志位
    LED1=!LED1;
}

```

main.c

```

17 #include "wwdg/wwdg.h"
18
19 int main(int argc, char **argv) {
20
21     delay_init();
22     NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2); // 设置中断优先级分组2
23     uart_init(9600);
24     LED_Init();
25     LED0 = 0;
26     delay_ms(300);
27     WWDG_Init(0X7F, 0X5F, WWDG_Prescaler_8);
28
29     while (1) {
30         LED0 = !LED0;
31         delay_ms(1000);
32     }
33 }
34
35 //int main(int argc, char **argv) {
36 //
37 //     delay_init();
38 //     LED_Init();

```

#include <misc.h>

#include <stm32f10x_wwdg.h>

#include "delay/delay.h"

#include "led/led.h"

#include "wwdg/wwdg.h"

int main(int argc, char **argv) {

```

    delay_init();
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2); // 设置中断优先级分组2
    LED_Init();
    LED0 = 0;
    WWDG_Init(0X7F, 0X5F, WWDG_Prescaler_8);
    while (1) {
        LED0 = !LED0;
        delay_ms(1000);
    }
}

```

效果:系统上电后LED0每隔1000ms闪烁一次，LED1会在喂狗时闪烁。