

## **PAN1080 HAL WWDT Sample Application Note**

PAN-CLT-VER-B0, Rev 0.1

PANCHIP

PanchipMicroelectronics

[www.panchip.com](http://www.panchip.com)

## 修订历史

版本	修订日期	描述
V0.1	2023-9-27	初始版本创建

PANCHIP

## 目录

第 1 章 例程演示内容 .....	4
1.1 测试内容 .....	4
1.2 环境准备 .....	4
1.2.1 软件环境 .....	4
1.2.1.1 待测代码 .....	4
1.2.1.2 软件工具 .....	4
1.2.2 硬件环境 .....	4
第 2 章 例程演示流程 .....	5
2.1 环境配置 .....	5
2.1.1 测试程序编译烧录 .....	5
2.1.2 硬件接线 .....	5
2.2 WWDT 工作流程 .....	5
2.3 测试程序初始化 .....	5
2.4 基本功能验证 .....	5
2.4.1 WWDT 所有寄存器默认状态 .....	5
2.4.2 超时窗口设置 .....	6
2.4.2.1 设置超时窗口时间, 且 WWDT_CLK 时钟源为 APB .....	6
2.4.2.2 设置超时窗口时间, 且 WWDT_CLK 时钟源为 32K RCL .....	7
2.4.3 重载计数器与复位测试 .....	8
2.4.3.1 在 CNTDAT 减至 CMPDAT 之前重新加载计数器 .....	9
2.4.3.2 在 CNTDAT 减至与 CMPDAT 相等的时候重新加载计数器 .....	10
2.4.3.3 在 CNTDAT 小于 CMPDAT 的时候重新加载计数器 .....	11
2.4.4 中断测试 .....	12
第 3 章 注意事项 .....	14

# 第1章 例程演示内容

## 1.1 测试内容

1. 寄存器默认值 (Register default value)
2. 超时窗口设置 (Timeout Window)
  - a) 设置超时窗口时间, 且 WWDT\_CLK 时钟源为 APB
  - b) 设置超时窗口时间, 且 WWDT\_CLK 时钟源为 32K RCL
3. 重载计数器与复位测试 (Reload and Reset)
  - a) 在 CNTDAT 减至 CMPDAT 之前重新加载计数器
  - b) 在 CNTDAT 减至与 CMPDAT 相等的时候重新加载计数器
  - c) 在 CNTDAT 小于 CMPDAT 的时候重新加载计数器
4. 中断测试 (Interrupt Test)

## 1.2 环境准备

### 1.2.1 软件环境

#### 1.2.1.1 待测代码

测试工程文件:

<PAN1080-DK>\03\_MCU\mcu\_samples\_hal\WWDT\keil\WWDT.uvprojx

测试源文件目录:

<PAN1080-DK>\03\_MCU\mcu\_samples\_hal\WWDT\src

#### 1.2.1.2 软件工具

- 1、SecureCRT (用于显示 PC 与 EVB 的交互过程, 打印 log 等)
- 2、KingstVIS (逻辑分析仪 LA1010 配套软件)

### 1.2.2 硬件环境

- 1、PAN1080 EVB 1 块
  - a) UART0 (测试交互接口, TX: P00, RX: P01, 波特率: 921600)
  - b) WWDT (待测模块)
  - c) GPIO (P30, 使用 LA 观察引脚变化, 可以知道某些事件到来的时间点, 如中断发生, 超时发生等)
  - d) SWD (用来调试和烧录程序, SWDCLK: P46, SWDIO: P47)
- 2、逻辑分析仪 (波形抓取工具)
- 3、JLink (SWD 调试与烧录工具)

## 第2章 例程演示流程

### 2.1 环境配置

#### 2.1.1 测试程序编译烧录

打开测试工程，确保可以编译通过。

#### 2.1.2 硬件接线

接线方面，需要：

1.将 EVB 板的 RX0 和 TX0 进行跳线，然后连接 USB->UART 到 PC。

2.WWDT 本身没有外部输出或输入，不过为准确获知一些事件到来的时间（如中断发生、超时发生等），程序中使用 GPIO P30 的电平变化来指示，因此需将 P30 接入逻辑分析仪。

### 2.2 WWDT 工作流程

参考 User Manual 文档。

### 2.3 测试程序初始化

硬件连线完成并烧录测试程序后，EVB 上电，观察串口是否正常打印测主菜单。

```
CPU @ 64000000Hz
+-----+
|                                     PAN1080 wwDT Sample Code.                                     |
+-----+
|                                     Press key to start specific testcase:                                     |
|                                     |                                     |                                     |
|                                     Input '0'   Testcase 0: Register Default value Check. |
|                                     Input '1'   Testcase 1: Timeout window.             |
|                                     Input '2'   Testcase 2: Reload and Reset.           |
|                                     Input '3'   Testcase 3: Interrupt.                 |
+-----+
```

### 2.4 基本功能验证

#### 2.4.1 WWDT 所有寄存器默认状态

在主菜单下，输入 ‘0’ 命令 打印所有寄存器默认值：

测试目的：

检查所有 WWDT 相关寄存器复位 Default 值状态。

测试预期：

寄存器默认值应和 Datasheet 上 WWDT 模块默认值一致。

测试现象:

```
0
WWDT Register Default Values:
-----
RLDCNT      = 0x00000000
CTL         = 0x003f0800
STATUS      = 0x00000000
CNT         = 0x0000003f

WWDT Flags: WWDTF=0, WWDTRF=0, WWDTIF=0
-----
WWDT Test OK, Success case: 0
```

测试分析:

参考芯片手册对比寄存器信息，发现是完全一致的，符合预期。

## 2.4.2 超时窗口设置

在主菜单下，输入 ‘1’ 命令 进入 Subcase 菜单:

```
-----
Press key to test specific function:
Input 'A'   Test Timeout window and set APB as WWDT_CLK.
Input 'B'   Test Timeout window and set LIRC as WWDT_CLK.
Press ESC key to back to the top level case list.
-----
```

### 2.4.2.1 设置超时窗口时间，且 WWDT\_CLK 时钟源为 APB

测试目的:

WWDT 时钟源为 APB 的情况下，设置预分频系数 (Prescaler) 和比较计数值 (Compare Value)，观察 WWDT 的比较匹配事件是否准确到来，观察 WWDT 能否准确在超时时 Reset 系统。

测试预期:

随意设置一个预分频系数和比较计数值，WDT 计数开始后，可以准确产生比较匹配事件，可以准确在超时时 Reset 系统。

测试现象:

先正确连接 Test Board 与逻辑分析仪，然后输入 ‘A’ 命令，这时候提示输入预分频值，手动输入一个 0~15 之间的数字 (如 4) 并按回车键，会继续提示输入比较计数值，再手动输入一个 0~63 之间的数字 (如 35) 并按回车键，可以看到 Log 打印设置完成的提示，Prescaler 最终为 16，Compare Value 为 35。接着提示计数开始，并打印出 APB Clock 频率。稍等片刻出现 Timeout Window reached 字样，系统 Reset。

```

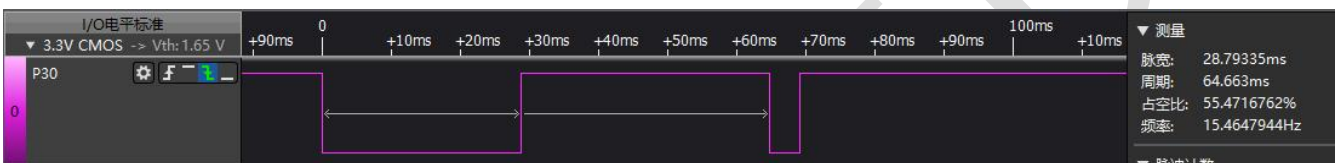
Please input PSCSEL (Range 0 ~ 15, Press <Enter> to confirm):
Please input Compare Value (Range 0 ~ 63, Press <Enter> to confirm):
Timeout window setting done, Prescaler: 16 (PSCSEL = 4), Compare Value: 35

Start WWDT Counting (Clock Source: APB1, Freq = 32000000Hz)...

Timeout window reached, wait for reset...

CPU @ 64000000Hz
    
```

再看 LA 的 P30 波形，发现开始的时候波形被拉低，接着在 28.79ms 后被拉高，接着波形一直保持高电平。



**测试分析:**

从 Log 可知，APB Clock 为 32MHz，由此可知 WWDT Clock:

$$WWDT\_CLK = \frac{APB1\_CLK}{2048} = \frac{32MHz}{2048} = 15625Hz$$

又由 Prescaler 为 16，CompareValue 为 35，于是比较匹配事件的产生时间（即 Counter 从最大值 0x3F 减至与 CompareValue 相同所需时间）为:

$$t_{event} = \frac{63 - CompareValue}{WWDT\_CLK/Prescaler} = \frac{63 - 35}{15625/16} s = 28.67ms$$

系统复位超时时间

$$t_{reset} = \frac{63}{WWDT\_CLK/Prescaler} = \frac{63}{15625/16} s = 64.51ms$$

再看 LA 波形，第一个低电平持续时间即为开始计数到比较匹配事件产生的时间，为 28.79ms，可见与计算的 28.67ms 基本完全相同；接着电平被拉高，然后 WWDT 计数超时，在 64.66ms 时再次被拉低，可见系统成功 Reset，与计算的 64.51ms 基本完全相同。

**2.4.2.2 设置超时窗口时间，且 WWDT\_CLK 时钟源为 32K RCL**

**测试目的:**

WWDT 时钟源为 32K RCL 的情况下，设置预分频系数 (Prescaler) 和比较计数值 (Compare Value)，观察 WWDT 的比较匹配事件是否准确到来，观察 WWDT 能否准确在超时后 Reset 系统。

**测试预期:**

随意设置一个预分频系数和比较计数值，WDT 计数开始后，可以准确产生比较匹配事件，可以准确在超时后 Reset 系统。



### 测试现象:

先正确连接 Test Board 与逻辑分析仪，然后输入 ‘B’ 命令，这时候提示输入预分频值，手动输入一个 0~15 之间的数字（如 8）并按回车键，会继续提示输入比较计数值，再手动输入一个 0~63 之间的数字（如 54）并按回车键，可以看到 Log 打印设置完成的提示，Prescaler 最终为 192，Compare Value 为 54。接着提示计数开始，并打印出 APB Clock 频率。稍等片刻出现 Timeout Window reached 字样，再稍等片刻，系统 Reset

```

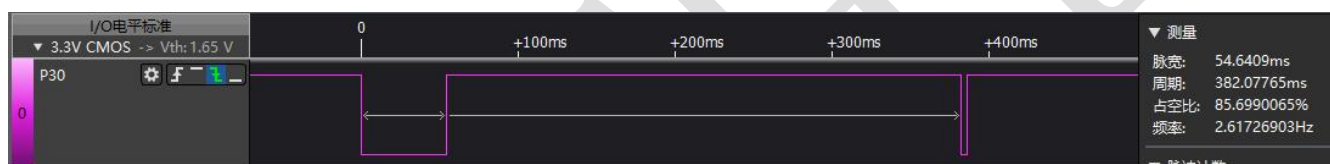
Please input PSCSEL (Range 0 ~ 15, Press <Enter> to confirm):
Please input Compare Value (Range 0 ~ 63, Press <Enter> to confirm):
Timeout window setting done, Prescaler: 192 (PSCSEL = 8), Compare Value: 54

Start WWDT Counting (Clock Source: 32KHZ LIRC)...

Timeout window reached, wait for reset...

CPU @ 64000000HZ
    
```

再看 LA 的 P30 波形，发现开始的时候波形被拉低，接着在 54.64ms 后被拉高，持续了 327.7ms 后又被拉低，持续一段时间被拉高，接着波形一直保持高电平。



### 测试分析:

从 Log 可知，由于时钟源选择为 32K RCL (RCL 未校准，根据之前测试实际值约为 31.76K)，可知 WWDT Clock:

$$WWDT\_CLK = RCL = 31.76KHz$$

又由 Prescaler 为 192，CompareValue 为 53，于是比较匹配事件的产生时间（即 Counter 从最大值 0x3F 减至与 CompareValue 相同所需时间）为：

$$t_{event} = \frac{63 - CompareValue}{WWDT\_CLK/Prescaler} = \frac{63 - 54}{31760/192} s = 54.4ms$$

系统复位超时时间

$$t_{reset} = \frac{63}{WWDT\_CLK/Prescaler} = \frac{63}{31760/192} s = 380.8ms$$

再看 LA 波形，第一个低电平持续时间即为开始计数到比较匹配事件产生的时间，为 54.64ms，可见与计算的 54.4ms 基本符合；接着电平被拉高，然后 WWDT 计数超时，在 327.4ms 时再次被拉低，与计算的 380.8ms 基本符合，系统可见成功 Reset。

### 2.4.3 重载计数器与复位测试

在主菜单下，输入 ‘2’ 命令 进入 Subcase 菜单:



```

Press key to test specific function:

Input 'A'   Reload Counter before CNTDAT is reduced to CMPDAT.
Input 'B'   Reload Counter when   CNTDAT is equal to   CMPDAT.
Input 'C'   Reload Counter while  CNTDAT is less than  CMPDAT.
Press ESC key to back to the top level case list.
    
```

### 2.4.3.1 在 CNTDAT 减至 CMPDAT 之前重新加载计数器

#### 测试目的:

验证递减计数值 CNTDAT 在减至与 Compare Value (CMPDAT) 相等之前的时候, 重新加载计数器, 观察是否触发系统复位。

#### 测试预期:

递减计数值 CNTDAT 在减至与 Compare Value (CMPDAT) 相等之前的时候, 试图重新加载计数器将会触发系统复位。

#### 测试现象:

先正确连接 Test Board 与逻辑分析仪, 然后输入 ‘A’ 命令, 可以看到 Log 打印 APB Clock 频率, 设定的预分频系数, 比较计数值等参数。稍等片刻打印当前计数值(51)并提示其比 Compare Value (25) 大, 接着试图重新加载计数器, Log 显示等待重启, 系统重启

```

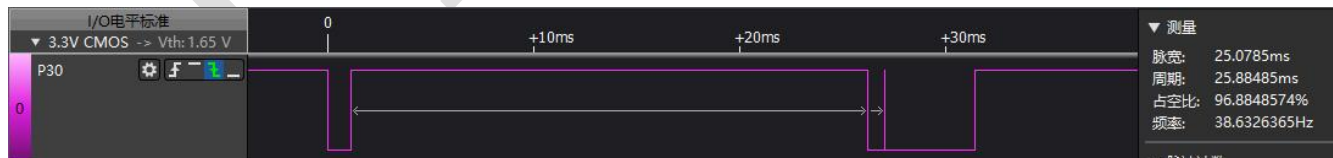
Press key to test specific function:

Input 'A'   Reload Counter before CNTDAT is reduced to CMPDAT.
Input 'B'   Reload Counter when   CNTDAT is equal to   CMPDAT.
Input 'C'   Reload Counter while  CNTDAT is less than  CMPDAT.
Press ESC key to back to the top level case list.

Start WDT Counting (APB1_CLK = 32000000Hz)...
Prescaler: 32 (PSCSEL = 5), Compare Value: 25

wwdt Counter Value = 51, and is larger than Compare value(25)
About to RCPu @ 64000000Hz
    
```

再看 LA 的 P30 波形, 发现开始的时候波形被拉低一段时间后拉高, 持续 25ms 后被拉低, 然后在 806.36us 后被拉高并在随后后拉低, 复位成功。



#### 测试分析:

从 Log 可知, 程序在试图 Reload Counter 的时候, Counter 值 (51) 是比 Compare 值 (25) 大的, 于是此时的 Reload Counter 操作将会触发系统复位。

又由 APB Clock 为 32MHz, Prescaler 为 32, CompareValue 为 25, 于是比较匹配事件的产生

时间（即 Counter 从最大值 0x3F 减至与 CompareValue 相同所需时间）为：

$$t_{event} = \frac{63 - CompareValue}{WWDT\_CLK/Prescaler} = \frac{63 - 25}{15625/32} s = 77.8ms$$

再看 LA 波形，第一个上升沿（WWDT 开始计数）至复位时间为 25.8ms，小于预期时间，符合预期

### 2.4.3.2 在 CNTDAT 减至与 CMPDAT 相等的时候重新加载计数器

#### 测试目的：

验证递减计数值 CNTDAT 在减至与 Compare Value（CMPDAT）相等的时候，重新加载计数器，观察加载是否成功，连续多次加载后，停止加载，观察系统是否能在预期的时间后复位。

#### 测试预期：

验证递减计数值 CNTDAT 在减至与 Compare Value（CMPDAT）相等的时候，能够重新加载计数器，连续 5 次加载后，停止加载，系统能在预期的时间后复位。

#### 测试现象：

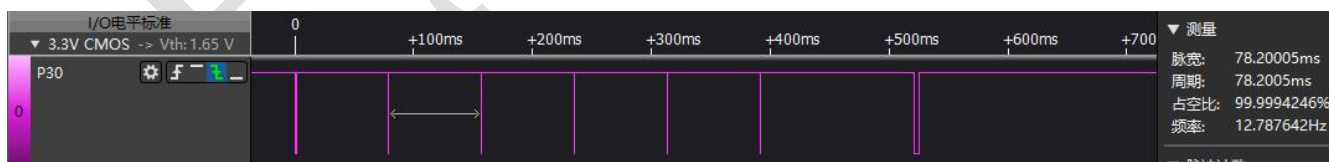
先正确连接 Test Board 与逻辑分析仪，然后输入 ‘B’ 命令，可以看到 Log 打印 APB Clock 频率，设定的预分频系数，比较计数值等参数。稍等片刻，发现打印 5 次当前计数值（25）并提示其与 Compare Value 相同，打印完成后再稍等片刻，Log 提示等待系统重启，系统重启：

```
b
Start WDT Counting (APB1_CLK = 32000000Hz)...
Prescaler: 32 (PSCSEL = 5), Compare Value: 25

WWDT Counter Value = 25 (Equal to Compare Value), Reload Counter...
WWDT Counter Value = 25 (Equal to Compare Value), Reload Counter...
WWDT Counter Value = 25 (Equal to Compare Value), Reload Counter...
WWDT Counter Value = 25 (Equal to Compare Value), Reload Counter...
WWDT Counter Value = 25 (Equal to Compare Value), Reload Counter...

wait For Reset...
```

再看 LA 的 P30 波形，发现开始的时候波形被拉低一段时间后拉高，然后每隔 78.2ms 出现一次向下的短脉冲，共出现了 5 次，保持高电平 P129ms 后拉低复位。



#### 测试分析：

从 Log 可知，程序在试图 Reload Counter 的时候，Counter 值（25）恰好与 Compare 值相等，于是此时的 Reload Counter 操作可以成功，WWDT 重新计数，连续 5 次。

又由 APB Clock 为 32MHz，Prescaler 为 32，CompareValue 为 25，于是比较匹配事件的产生时间（即 Counter 从最大值 0x3F 减至与 CompareValue 相同所需时间）为：

$$t_{event} = \frac{63 - CompareValue}{WWDT\_CLK/Prescaler} = \frac{63 - 25}{15625/32} s = 77.8ms$$

系统复位超时时间

$$t_{reset} = \frac{63}{WWDT\_CLK/Prescaler} = \frac{63}{15625/32} s = P129ms$$

再看 LA 波形，波形第一个上升沿表示 WWDT 开始计数，接着出现向下的脉冲表示 Reload Counter 操作，此段时间为 78.2ms，与计算的  $t_{event}$  (77.8ms) 时间基本相同，于是 Counter 被重新加载，在重复了 5 次后，出现一段高电平后，再被拉低一段时间（表示系统 reset），实际观察到系统复位。

### 2.4.3.3 在 CNTDAT 小于 CMPDAT 的时候重新加载计数器

#### 测试目的：

验证递减计数值 CNTDAT 在减至比 Compare Value (CMPDAT) 小的时候，重新加载计数器，观察加载是否成功，连续多次加载后，停止加载，观察系统是否能在预期的时间后复位。

#### 测试预期：

验证递减计数值 CNTDAT 在减至比 Compare Value (CMPDAT) 小的时候，能够重新加载计数器，连续 5 次加载后，停止加载，系统能在预期的时间后复位。

#### 测试现象：

先正确连接 Test Board 与逻辑分析仪，然后输入 ‘C’ 命令，可以看到 Log 打印 APB Clock 频率，设定的预分频系数，比较计数值等参数。稍等片刻，发现打印 5 次当前计数值 (20) 并提示其比 Compare Value (25) 小，打印完成后再稍等片刻，系统重启：

```
C
Start WDT Counting (APB1_CLK = 32000000Hz)...
Prescaler: 32 (PSCSEL = 5), Compare Value: 25

wwdt Counter value = 20, and is less than Compare value(25)
wwdt Counter value = 20, and is less than Compare value(25)
wwdt Counter value = 20, and is less than Compare value(25)
wwdt Counter value = 20, and is less than Compare value(25)
wwdt Counter value = 20, and is less than Compare value(25)
wait For Reset...
```

再看 LA 的 P30 波形，发现开始的时候波形被拉低一段时间后拉高，然后每隔 90.2ms 出现一次向下的短脉冲，共出现了 5 次，保持高电平 P129ms 后拉低复位。



### 测试分析:

从 Log 可知，程序在试图 Reload Counter 的时候，Counter 值（20）已经比 Compare 值（25）小，于是此时的 Reload Counter 操作可以成功，WWDT 重新计数，连续 5 次。

又由 APB Clock 为 32MHz，Prescaler 为 32，CompareValue 为 25，于是比较匹配事件的产生时间（即 Counter 从最大值 0x3F 减至与 CompareValue 相同所需时间）为：

$$t_{event} = \frac{63 - CompareValue}{WWDT\_CLK/Prescaler} = \frac{63 - 25}{15625/32} s = 77.8ms$$

系统复位超时时间

$$t_{reset} = \frac{63}{WWDT\_CLK/Prescaler} = \frac{63}{15625/32} s = P129ms$$

再看 LA 波形，波形第一个上升沿表示 WWDT 开始计数，接着出现向下的脉冲表示 Reload Counter 操作，此段时间为 90.2ms，比计算的  $t_{event}$ （77.8ms）时间大，但是比  $t_{reset}$ （P129ms）时间要小，于是 Counter 被重新加载，在重复了 5 次后，出现一段高电平后，再被拉低一段时间（表示系统 reset），实际观察到系统复位。

## 2.4.4 中断测试

在主菜单下，输入 ‘3’ 命令 进入 Subcase 菜单：

```

+-----+
| Press key to test specific function:                |
| Input 'A'   Enable wwdt Interrupt.                 |
| Press ESC key to back to the top level case list. |
+-----+
    
```

### 测试目的:

验证 WWDT 中断能否准确及时产生。

### 测试预期:

WWDT 计数开始，在计数 Counter 值减至与 Compare Value 相同的时候，触发中断。

### 测试现象:

先正确连接 Test Board 与逻辑分析仪，然后输入 ‘A’ 命令，可以看到 Log 打印 APB Clock 频率为 32MHz，稍后打印出 5 个 WWDT INT 的标志，随后 Log 提示等待 reset，但系统并未 reset。



```

a
Start WDT Counting (APB1_CLK = 32000000Hz)...
Prescaler: 32 (PSCSEL = 5), Compare Value: 25
WWDT INT
WWDT INT
WWDT INT
WWDT INT
WWDT INT
wait For Reset...
    
```

再看 LA 的 P30 波形，在第一次被拉低后，连续出现 5 次向上的脉冲，每两个脉冲间隔为 78.2ms，接着很短时间内波形被拉高，并一直保持高电平



### 测试分析:

从 Log 可知，程序在每隔一段时间触发一次中断，中断 Handler 中会有 Reload Counter 的操作，共触发了 5 次中断。

又由 APB Clock 为 32MHz，Prescaler 为 32，CompareValue 为 25，于是比较匹配事件的产生时间（即 Counter 从最大值 0x3F 减至与 CompareValue 相同所需时间）为：

$$t_{event} = \frac{63 - CompareValue}{WWDT\_CLK/Prescaler} = \frac{63 - 25}{15625/32} s = 77.8ms$$

系统复位超时时间

$$t_{reset} = \frac{63}{WWDT\_CLK/Prescaler} = \frac{63}{15625/32} s = P129ms$$

再看 LA 波形，P30 拉低至 A1 至时间，是程序在 WWDT 开始计数后的延时时间，这段时间内，共触发了 5 次中断，每两次间隔时间为 78.2ms，与计算的  $t_{event}$  (77.8ms) 时间基本相同；接着 P30 被拉高（程序关中断，中断 Handler 无法被执行，于是 Counter 无法被重新加载），在最后一次 Reload Counter 开始的 P129ms 时间内（A1 至 A2），都没有再次 Reload Counter 的操作，于是系统发生 Reset，符合预期。

## 第3章 注意事项

1、WWDT 的特点是只有在设定的窗口时间内 Reload Counter 才能成功维持系统工作，如果在窗口时间外做此操作将会立刻 Reset 系统

2、WWDT CTL 寄存器，只有 Int Enable 开关 (INTEN) 可以在开启之后关掉，其他的 bit (WWDTEN、PSCSEL、CMPDAT) 在写值后，直到下次 reset 之前均无法再被修改，也即 WWDT 一旦启用就无法被禁用，直到下次 reset