

零死角玩转STM32



独立看门狗

淘宝：firestm32.taobao.com

野火论坛：www.firebbs.cn

主讲内容



01

独立看门狗简介

02

独立看门狗功能框图讲解

参考资料: 《零死角玩转STM32》

“IWDG—独立看门狗” 章节

独立看门狗简介



看门狗：独立看门狗（宠物狗），窗口看门狗（警犬）



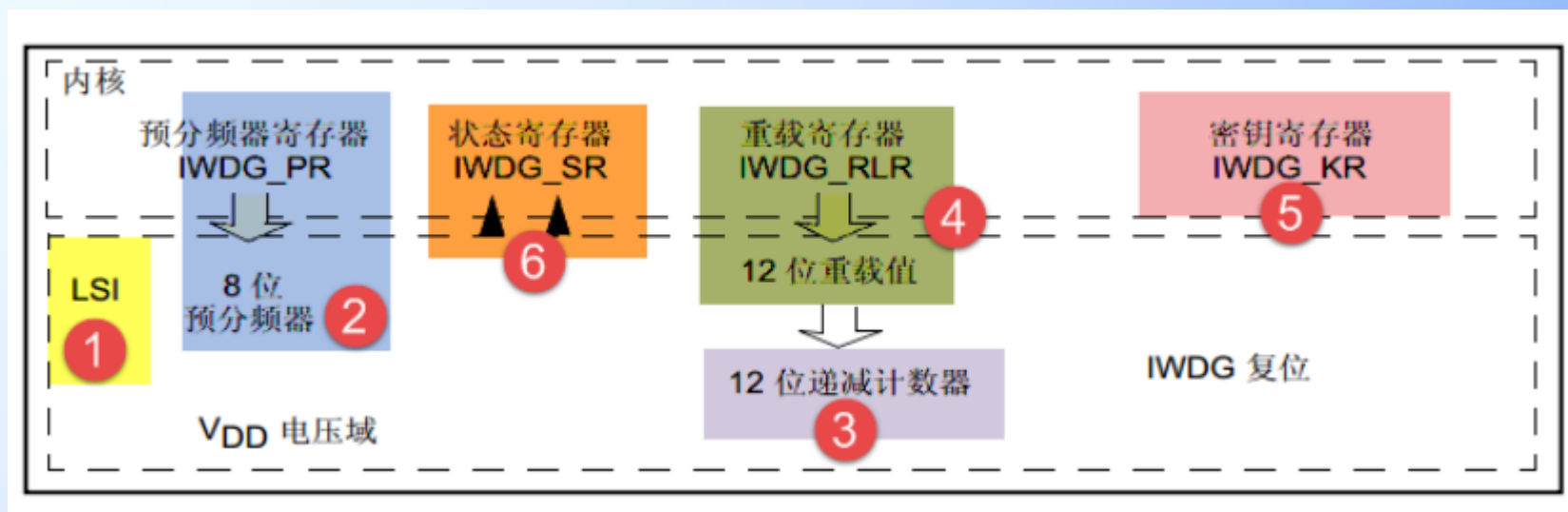
独立看门狗工作原理

独立看门狗简介



独立看门狗：独立看门狗用通俗一点的话来解释就是一个 12 位的递减计数器，当计数器的值从某个值一直减到 0 的时候，系统就会产生一个复位信号，即 IWDG_RESET。如果在计数没减到 0 之前，刷新了计数器的值的话，那么就不会产生复位信号，这个动作就是我们经常说的喂狗。看门狗功能由 VDD 电压域供电，在停止模式和待机模式下仍能工作。

独立看门狗功能框图讲解



1-独立看门狗时钟

2-计数器时钟

3-计数器

4-重装载寄存器

5-键寄存器

6-状态寄存器



独立看门狗时钟

独立看门狗的时钟由独立的 RC 振荡器 LSI 提供，即使主时钟发生故障它仍然有效，非常独立。LSI 的频率一般在 30~60KHZ 之间，根据温度和工作场合会有有一定的漂移，我们一般取 40KHZ，所以独立看门狗的定时时间并一定非常精确，只适用于对时间精度要求比较低的场合



计数器时钟

递减计数器的时钟由 LSI 经过一个 8 位的预分频器得到，我们可以操作预分频器寄存器 IWDG_PR 来设置分频因子，分频因子可以是：

[4,8,16,32,64,128,256,256]，计数器时钟

$$CK_CNT = 40 / (4 * 2^{PRV}).$$



计数器

独立看门狗的计数器是一个 12 位的递减计数器，最大值为 0XFFF，当计数器减到 0 时，会产生一个复位信号:IWDG_RESET，让程序重新启动运行，如果在计数器减到 0 之前刷新了计数器的值的话，就不会产生复位信号，重新刷新计数器值的这个动作我们俗称喂狗。



重装载寄存器

重装载寄存器是一个 12 位的寄存器，里面装着要刷新到计数器的值，这个值的大小决定着独立看门狗的溢出时间。超时时间 $T_{out} = (4 * 2^{prv}) / 40 * rlr (s)$ ，prv 是预分频器寄存器的值，rlr 是重装载寄存器的值

基本定时器功能框图讲解



键值寄存器（控制寄存器）

表格 34-1 键寄存器取值枚举

键值	键值作用
0XAAAA	把 RLR 的值重装载到 CNT
0X5555	PR 和 RLR 这两个寄存器可写
0XCCCC	启动 IWDG

通过写往键寄存器写 0XCCC 来启动看门狗是属于软件启动的方式，一旦独立看门狗启动，它就关不掉，只有复位才能关掉。



状态寄存器

状态寄存器 SR 只有位 0: PVU 和位 1: RVU 有效, 这两位只能由硬件操作, 软件操作不了。RVU: 看门狗计数器重装载值更新, 硬件置 1 表示重装载值的更新正在进行中, 更新完毕之后由硬件清 0。PVU: 看门狗预分频值更新, 硬件置 '1' 指示预分频值的更新正在进行中, 当更新完成后, 由硬件清 0。所以只有当 RVU/PVU 等于 0 的时候才可以更新重装载寄存器/预分频寄存器。

怎么使用独立看门狗



怎么使用IWDG

- 1-一般用来检测和解决由程序引起的故障
- 2-比如一个程序正常运行的时间是50ms，在运行完这个段程序之后紧接着进行喂狗，我们设置独立看门狗的定时溢出时间为60ms，比我们需要监控的程序50ms 多一点，如果超过 60ms 还没有喂狗，那就说明我们监控的程序出故障了，跑飞了，那么就会产生系统复位，让程序重新运行。

怎么使用独立看门狗



怎么使用IWDG (伪代码)

```
void main(void)
{
    /* 各种初始化 */
    {
        .....
    }

    /* 启动独立看门狗 */
    IWDG_Start();
    while(1)
    {
        /* 需要被监控的代码 */
        {
            .....
        }

        /* 喂狗 */
        IWDG_Feed();
    }
}
```

独立看门狗固件库函数



编程时需要用到的几个固件库函数

```
/* Prescaler and Counter configuration functions *****/
void IWDG_WriteAccessCmd(uint16_t IWDG_WriteAccess);
void IWDG_SetPrescaler(uint8_t IWDG_Prescaler);
void IWDG_SetReload(uint16_t Reload);
void IWDG_ReloadCounter(void);

/* IWDG activation function *****/
void IWDG_Enable(void);

/* Flag management function *****/
FlagStatus IWDG_GetFlagStatus(uint16_t IWDG_FLAG);
```


零死角玩转STM32



THANKS

野火论坛 : www.firebbs.cn

淘宝 : firestm32.taobao.com