

# STM32Cube高效开发教程（基础篇）

## 第20章 独立看门狗

---

王维波

中国石油大学（华东）控制科学与工程学院

# STM32Cube高效开发教程（基础篇）

作者：王维波，鄢志丹，王钊

人民邮电出版社

2021年9月出版

如果有读者需要本书课件的PPT版本用于备课，可以给作者发邮件免费获取，并可加入专门的教学和技术交流QQ群

邮箱：[wangwb@upc.edu.cn](mailto:wangwb@upc.edu.cn)

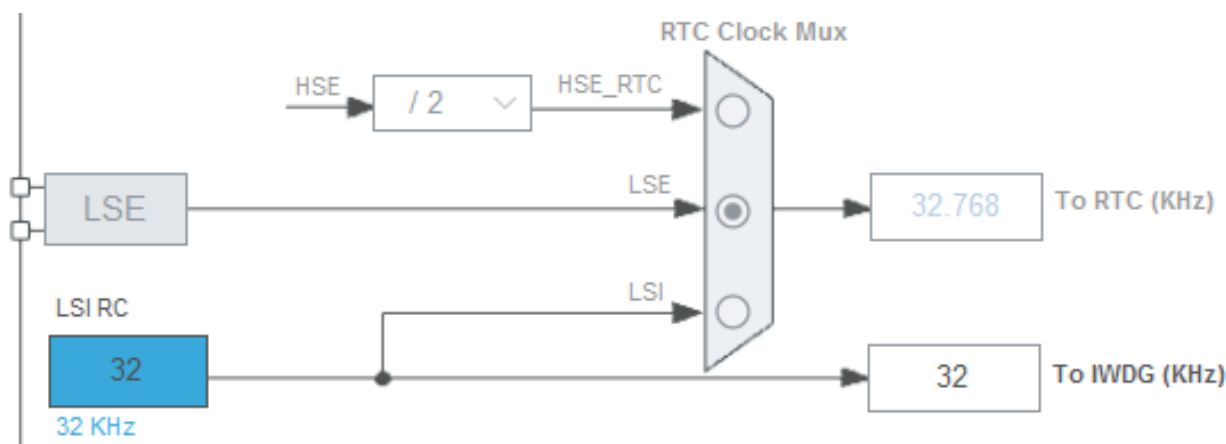


20.1 独立看门狗工作原理

20.2 独立看门狗HAL驱动程序

20.3 独立看门狗使用示例

独立看门狗（Independent Watchdog, IWDG）是由内部32kHz低速时钟LSI驱动的自由运行的12位递减计数器。LSI在时钟树上的位置如图所示。



系统复位时，IWDG的12位递减计数器的值是4095。启动IWDG后，计数器就递减计数，当计数器值变为0x000的时候就会使系统产生复位。

LSI时钟频率是32kHz，看门狗最大重载值是4095（对应0xFFF），根据预分频系数可以计算出IWDG的最长超时时间（timeout），如表所示。

预分频系数	超时时间（ms）
/4	512
/8	1024
/16	2048
/32	4096
/64	8192
/128	16384
/256	32768

20.1 独立看门狗工作原理

20.2 独立看门狗HAL驱动程序

20.3 独立看门狗使用示例

## 1. 初始化函数HAL\_IWDG\_Init()

函数HAL\_IWDG\_Init()用于初始化独立看门狗，其函数原型定义如下：

```
HAL_StatusTypeDef HAL_IWDG_Init(IWDG_HandleTypeDef *hiwdg);
```

参数hiwdg是IWDG\_HandleTypeDef结构体指针，是看门狗对象指针。独立看门狗初始化后就自动启动了，且无法关闭。

看门狗初始化主要需要设置2个参数：

- 预分频系数
- 重载寄存器值

## 2. 刷新看门狗的函数HAL\_IWDG\_Refresh()

函数HAL\_IWDG\_Refresh()用于刷新看门狗，就是将重载寄存器IWDG\_RLR的值重新载入看门狗计数器，避免产生系统复位。

```
HAL_StatusTypeDef HAL_IWDG_Refresh(IWDG_HandleTypeDef *hiwdg);
```



20.1 独立看门狗工作原理

20.2 独立看门狗HAL驱动程序

20.3 独立看门狗使用示例

## 20.3.1 示例功能和CubeMX项目设置

本节示例Demo20\_1IWDG演示独立看门狗的使用。

- 配置独立看门狗的超时时间为8190ms。
- 使用RTC周期唤醒功能，唤醒周期为1秒，使用一个全局变量Seconds进行秒计时。
- 按下任何按键时刷新看门狗，使全局变量Seconds归零。
- 超过8秒钟无按键刷新看门狗，系统将复位。

# RTC设置

启用RTC周期唤醒功能，唤醒周期为1秒

RTC Mode and Configuration

Mode

☒ Activate Clock Source

☒ Activate Calendar

Alarm A

Alarm B

WakeUp

Configuration

☒ Parameter Settings ☒ User Constants ☒ NVIC Settings

> Calendar Date

Wake UP

Wake Up Clock	1 Hz
Wake Up Counter	0

# 独立看门狗设置

## 设置预分频系数和递减计数器重载值

IWDG Mode and Configuration

Mode

☒ Activated

Configuration

Reset Configuration

☒ Parameter Settings

☒ User Constants

<

>

Clocking

IWDG counter clock prescaler64

IWDG down-counter reload value4095

## 20.3.2 程序功能实现

### 1. 主程序

主程序while循环里检测按键，任意键按下后刷新看门狗

```
while (1)
{
    KEYS curKey=ScanPressedKey(KEY_WAIT_ALWAYS);
    switch(curKey)
    {
        case KEY_DOWN:
        case KEY_UP:
            HAL_IWDG_Refresh(&hiwdg); //刷新看门狗
            break;

        case KEY_LEFT:
        case KEY_RIGHT:
            __HAL_IWDG_RELOAD_COUNTER(&hiwdg); //刷新看门狗
    }
    Seconds=0;//秒计数值清零
    LCD_ShowStr(30,PosY+LCD_SP20, (uint8_t *)"IWDG is refreshed");
    HAL_Delay(500); //消除按键抖动影响
}
```

## RTC唤醒中断里更新时间显示

```
/* RTC周期唤醒中断回调函数 */  
void HAL_RTCEx_WakeUpTimerEventCallback(RTC_HandleTypeDef *hrtc)  
{  
    Seconds++;    //秒计数值  
    LCD_ShowUintX(PosX,PosY,Seconds,3);  
}
```

## 2. 独立看门狗初始化

函数MX\_IWDG\_Init()是CubeMX自动生成的独立看门狗初始化函数，在文件iwdg.c中实现，其代码如下：

```
/* 文件： iwdg.c -----*/
#include "iwdg.h"
IWDG_HandleTypeDef hiwdg;           //IWDG外设对象变量

/* IWDG 初始化函数 */
void MX_IWDG_Init(void)
{
    hiwdg.Instance = IWDG;           //寄存器基址
    hiwdg.Init.Prescaler = IWDG_PRESCALER_64; //分频系数
    hiwdg.Init.Reload = 4095;        //重载入值
    if (HAL_IWDG_Init(&hiwdg) != HAL_OK)
        Error_Handler();
}
```

## 运行测试

```
Demo20_1:Independent Watchdog
IWDG should be refreshed in 8s
Press any key to refresh IWDG

Seconds after last refresh: 5
```

系统复位后就开启了独立看门狗，并开始计时。如果在8秒内没有刷新看门狗，系统就会复位

```
Demo20_1:Independent Watchdog
IWDG should be refreshed in 8s
Press any key to refresh IWDG

Seconds after last refresh: 6

IWDG is refreshed
```

在计数到8之前，按任意按键可以刷新看门狗，并重新开始计数