

### 使用 PY32F030/003/002A IWDG/WWDG 的方法

#### 前言

PY32F030/003/002A 内置两个看门狗, 提供了更高的安全性, 时间的精确性和使用的灵活性。两个看门狗(独立看门狗和窗口看门狗)可用于检测 and 解决软件错误引起的故障, 当计数器达到给定的超时值时, 触发一个中断(仅适用于窗口型看门狗)或产生系统复位。

本应用笔记提供了含有配置独立看门狗和窗口看门狗的代码例程。

在本文档中, PY32 仅指表 1 中列出的产品系列。

表 1. 适用产品

类型	产品系列
微型控制器系列	PY32F030、PY32F003、PY32F002A
	注: 002A 没有窗口看门狗

---

## 目录

1	WDG 功能简介	3
2	WDG 应用例程	4
2.1	独立看门狗	5
2.2	窗口看门狗	5
3	版本历史	6

PUYA CONFIDENTIAL

## 1 WDG 功能简介

- WDG 分为 IWDG 和 WWDG，他们的特性如下图 1-1 所示。

表 1-1 IWDG 和 WWDG 的特性对比

NO.	IWDG	WWDG
1.中文名	独立看门狗	窗口看门狗
2.时钟源	LSI	PCLK
3.计数方式	12 位递减	6 位递减
4.范围	只有下限	有上限和下限
5.中断	没有中断，超时直接复位	有中断，中断做复位前的函数操作
6.使用条件	避免程序跑飞或者进入死循环	避免程序不按预定逻辑执行

PUYA CONFIDENTIAL

## 2 WDG 应用例程

### 2.1 独立看门狗

- 配置独立看门狗步骤:

步骤	操作
1	使能 LSI 时钟
2	初始化独立看门狗(分频系数, 重装载值)
3	在规定时间内喂狗

- 独立看门狗配置代码介绍: 打开我们的 IWDG 例程代码, 此例程配置了独立看门狗重装载值为 1000ms, 我们必须在 1000ms 内进行一次喂狗操作, 即向 IWDG\_KR 寄存器中写入 0XAAAA, 否则将会产生复位。

- 打开例程代码, 在 py32f030\_hal\_msp.c 文件中, HAL\_Msplnit 函数使能了 LSI 时钟, 初始化了 LED。

```
void HAL_Msplnit(void)
{
    /* USER CODE BEGIN Msplnit 0 */
    BSP_LED_Init(LED_GREEN);

    /* 使能 LSI 时钟 */
    __HAL_RCC_LSI_ENABLE();

    /* 等待直到 LSI READY 置位 */
    while (READ_BIT(RCC->CSR, RCC_CSR_LSIRDY) == 0U) {}
    /* USER CODE END Msplnit 0 */
}
```

- 在 main.c 中初始化独立看门狗, 配置预分频系数为 32, 然后重装载值为 1000, 即计数器计数 1000 次, 即 1000ms 后如果没有进行喂狗会产生复位。

```
/*-3- Configure & Start the IWDG peripheral */
IwdgHandle.Instance = IWDG;
IwdgHandle.Init.Prescaler = IWDG_PRESCALER_32;//T=1MS
IwdgHandle.Init.Reload = (1000); //1ms*1000=1s
IwdgHandle.Init.Window = IWDG_WINDOW_DISABLE;
if(HAL_IWDG_Init(&IwdgHandle) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}
```

- 在 main.c 中, 程序执行的 while 循环里, 我们要在 1000ms 内进行喂狗, 否则将会发生复位。

```
/* Infinite loop */
while (1)
{
    /* Insert delay */
    HAL_Delay(900); //每 900ms 喂一次狗, 可以正常运行

    /* 翻转 LED 灯 */
    BSP_LED_Toggle(LED_GREEN);
    /* Refresh IWDG: reload counter */
    if(HAL_IWDG_Refresh(&IwdgHandle) != HAL_OK)
```

```

    {
        /* Refresh Error */
        Error_Handler();
    }
}

```

2.2 窗口看门狗

- 配置窗口看门狗步骤:

步骤	操作
1	使能 WWDG 时钟
2	配置窗口看门狗中断优先级, 使能中断
3	初始化窗口看门狗(时基, 计数值, 窗口值, 唤醒中断等)
4	在唤醒中断里喂狗, 即刷新计数器的值

- 独立看门狗配置代码介绍: 打开我们的 WWDG\_IT 例程代码, 此样例演示了 WWDG 的提前唤醒中断功能, 看门狗计数器向下计数到 0x40 时产生中断, 中断中喂狗, 可以确保看门狗不会复位。

- 打开例程代码, 在 py32f030\_hal\_msp.c 文件中, HAL\_WWDG\_MspInit 函数使能了 WWDG 时钟, 配置窗口看门狗的中断优先级。

```

void HAL_WWDG_MspInit(WWDG_HandleTypeDef *hwwdg)
{
    /* WWDG Peripheral clock enable */
    __HAL_RCC_WWDG_CLK_ENABLE();
    NVIC_SetPriority(WWDG_IRQn,0);
    NVIC_EnableIRQ(WWDG_IRQn);
}

```

- 在 main.c 中初始化窗口看门狗, 配置时基, 计数值, 窗口值和使能唤醒中断。

```

/* WWDG 模块初始化 */
WwdgHandle.Instance = WWDG;
WwdgHandle.Init.Prescaler = WWDG_PRESCALER_8;
WwdgHandle.Init.Window = 127;
WwdgHandle.Init.Counter = 127;
WwdgHandle.Init.EWIMode = WWDG_EWI_ENABLE;
if (HAL_WWDG_Init(&WwdgHandle) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}

```

- 窗口看门狗中断产生后, 代码运行到中断回调函数 HAL\_WWDG\_EarlyWakeupCallback 里, 在这里我们执行喂狗操作。

```

void HAL_WWDG_EarlyWakeupCallback(WWDG_HandleTypeDef *hwwdg)
{
    /* 刷新 WWDG 计数器 */
    if (HAL_WWDG_Refresh(hwwdg) != HAL_OK)
    {
        Error_Handler();
    }
    /* 翻转 LED 灯 */
    BSP_LED_Toggle(LED_GREEN);
}

```

### 3 版本历史

版本	日期	更新记录
V0.1	2021.10.20	初版
V1.0	2022.06.22	修改了应用例程的内容
V1.1	2022.10.24	增加 002A 内容



Puya Semiconductor Co., Ltd.

#### IMPORTANT NOTICE

Puya Semiconductor reserves the right to make changes without further notice to any products or specifications herein. Puya Semiconductor does not assume any responsibility for use of any its products for any particular purpose, nor does Puya Semiconductor assume any liability arising out of the application or use of any its products or circuits. Puya Semiconductor does not convey any license under its patent rights or other rights nor the rights of others.