
本章教程通过给PA0引脚置高电平进行喂狗，并通过串口调试助手打印显示程序运行状态，具体如下：

当PA0引脚有连接高电平，即有喂狗，串口打印显示：Feed dog success；

当PA0引脚未连接高电平，即无喂狗，串口打印显示：Feed dog fail,Reset。

1、IWDG简介及相关函数介绍

独立看门狗（IWDG）通常用来检测逻辑错误和外部环境干扰引起的软件故障，其时钟源来自于LSI，可独立于主程序之外运行，适用于对精度要求低的场合。独立看门狗可看作为一个12位递减计数器，当计数器的值从某个值递减到0后，系统会产生一个复位信号；若在计数器未递减到0之前刷新计数器的值，则不会产生复位信号，此过程即是喂狗。

CH32V103独立看门狗的时钟来源 LSI 时钟分频，其功能在停机和待机模式时仍能正常工作。其具有12位自减型计数器，当看门狗计数器自减到 0 时，将产生系统复位，所以超时时间为（重装载值+1）个时钟，最大可达 26.2s，最小可达 100us。关于CH32V103 独立看门狗具有说明，可参考CH32V103应用手册。

独立看门狗相关函数和定义分布在文件ch32v10x_iwdg.c和ch32v10x_iwdg.h文件中，关于这两文件的函数具体介绍如下：

1. void IWDG_WriteAccessCmd(uint16_t IWDG_WriteAccess);
2. void IWDG_SetPrescaler(uint8_t IWDG_Prescaler);
3. void IWDG_SetReload(uint16_t Reload);
4. void IWDG_ReloadCounter(void);
5. void IWDG_Enable(void);
6. FlagStatus IWDG_GetFlagStatus(uint16_t IWDG_FLAG);

复制代码

1.1、void IWDG_WriteAccessCmd(uint16_t IWDG_WriteAccess)
功 能：启用或禁用对IWDG_PSCR和IWDG_RLDR寄存器的写入访

问。

输入：WDG_WriteAccess：对IWDG_PSCR和IWDG_RLDR寄存器的写入访问的新状态。IWDG_WriteAccess_Enable：启用对IWDG_PSCR和IWDG_RLDR寄存器的写入访问；
IWDG_WriteAccess_Disable：禁用对IWDG_PSCR和IWDG_RLDR寄存器的写入访问。

1.2、void IWDG_SetPrescaler(uint8_t IWDG_Prescaler)

功能：设置IWDG预分频器值。

输入：IWDG_Prescaler：指定IWDG预分频器值（4、8、16、34、64、128、256）。

1.3、void IWDG_SetReload(uint16_t Reload)

功能：设置IWDG重新加载值。

输入：Reload：指定IWDG Reload值。此参数必须是介于0和0x0FFF之间的数字。

1.4、void IWDG_ReloadCounter(void)

功能：使用重新加载寄存器中定义的值重新加载IWDG计数器。

输入：无

1.5、void IWDG_Enable(void)

功能：启用IWDG（已禁用对IWDG_PSCR和IWDG_RLDR寄存器的写入访问）

输入：无

1.6、FlagStatus IWDG_GetFlagStatus(uint16_t IWDG_FLAG)

功能：检查是否设置了指定的IWDG标志。

输入：IWDG_FLAG：指定要检查的标志。IWDG_FLAG_PVU：正在进行预分频器值更新。IWDG_FLAG_RVU：正在进行重新加载值更新。

在进行IWDG相关程序编写时，若需用到上述函数，直接在程序进行调用即可。

2、硬件设计

本教程通过PA0引脚接VCC引脚进行喂狗，连接方式如下：

- 用杜邦线将PA0引脚与开发板VCC引脚连接喂狗。

3、软件设计

独立看门狗喂狗操作步骤如下：

- 取消寄存器写保护；

- 设置独立看门狗的预分频系数和重装载值;
- 重载计数值喂狗;
- 启动看门狗。

根据上述步骤，编写相关程序，具体如下：

iwdg.h文件

```

1. #ifndef __IWDG_H
2. #define __IWDG_H
3.
4. #include "ch32v10x_conf.h"
5.
6. #define PA0 GPIO_ReadInputDataBit( GPIOA, GPIO_Pin_0) //PA0
7.
8. void GPIOx_Init(void);
9. u8 GPIOx_Connet(void);
10. void IWDG_Feed_Init( u16 prer, u16 rlr );
11.
12. #endif

```

复制代码

iwdg.h文件主要用于保存相关函数的定义声明。

iwdg.c文件

```

1. #include "iwdg.h"
2.
3. void GPIOx_Init(void)
4. {
5.     GPIO_InitTypeDef GPIO_InitStructure;
6.
7.     RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA, ENABLE );
        //使能GPIOA时钟
8.
9.     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;    //配置GPIO引
        脚, PA0
10.     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPD; //配置GPIO
        模式为下拉输入
11.     GPIO_Init( GPIOA, &GPIO_InitStructure);    //GPIO初始化
12. }
13.
14. u8 GPIOx_Connet(void)
15. {
16.     if(PA0 == 1)    //当PA0引脚连接高电平，返回值为1，否则返回
        0;
17.     {
18.         Delay_Ms(10);
19.         return 1;
20.     }
21.     return 0;

```

```

22. }
23.
24. void IWDG_Feed_Init( u16 prer, u16 rlr )
25. {
26.     IWDG_WriteAccessCmd(IWDG_WriteAccess_Enable); //使能取消
        IWDG_PR和IWDG_RLR寄存器的写保护,
27.     IWDG_SetPrescaler(prer);                //设置预分频器值
28.     IWDG_SetReload(rlr);                    //设置重装载寄存器的值
29.     IWDG_ReloadCounter();                  //将重装载寄存器的值放入
        到计数器中
30.     IWDG_Enable();                          //使能IWDG
31. }

```

复制代码

iwdg.c文件是独立看门狗的配置程序，其具体配置流程如下：

1、使能GPIOA时钟；

```

1. RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA, ENABLE ); //
    使能GPIOA时钟

```

复制代码

2、初始化对应GPIO口；

```

1. GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;    //配置GPIO引脚,
    PA0
2. GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPD; //配置GPIO模
    式为下拉输入
3. GPIO_Init( GPIOA, &GPIO_InitStructure);     //GPIO初始化

```

复制代码

3、启用对IWDG_PSCR和IWDG_RLDR寄存器的写入访问；

```

1. IWDG_WriteAccessCmd(IWDG_WriteAccess_Enable); //使能取消
    IWDG_PR和IWDG_RLR寄存器的写保护,

```

复制代码

4、设置独立看门狗的预分频系数和重装载值；

```

1. IWDG_SetPrescaler(prer);                //设置预分频器值
2. IWDG_SetReload(rlr);                    //设置重装载寄存器的值

```

复制代码

5、重载计数值喂狗；

```

1. IWDG_ReloadCounter();                  //将重装载寄存器的值放入到
    计数器中

```

复制代码

6、启动看门狗；

```

1. IWDG_Enable();                          //使能IWDG

```

复制代码

7、编写判断函数，用于判断是否连接高电平喂狗。

```
1. u8 GPIOx_Connet(void)
2. {
3.     if(PA0 == 1)    //当PA0引脚连接高电平，返回值为1，否则返回
        0;
4.     {
5.         Delay_Ms(10);
6.         return 1;
7.     }
8.     return 0;
9. }
```

复制代码

iwdg.c文件主要进行独立看门狗的相关配置。

main.c文件

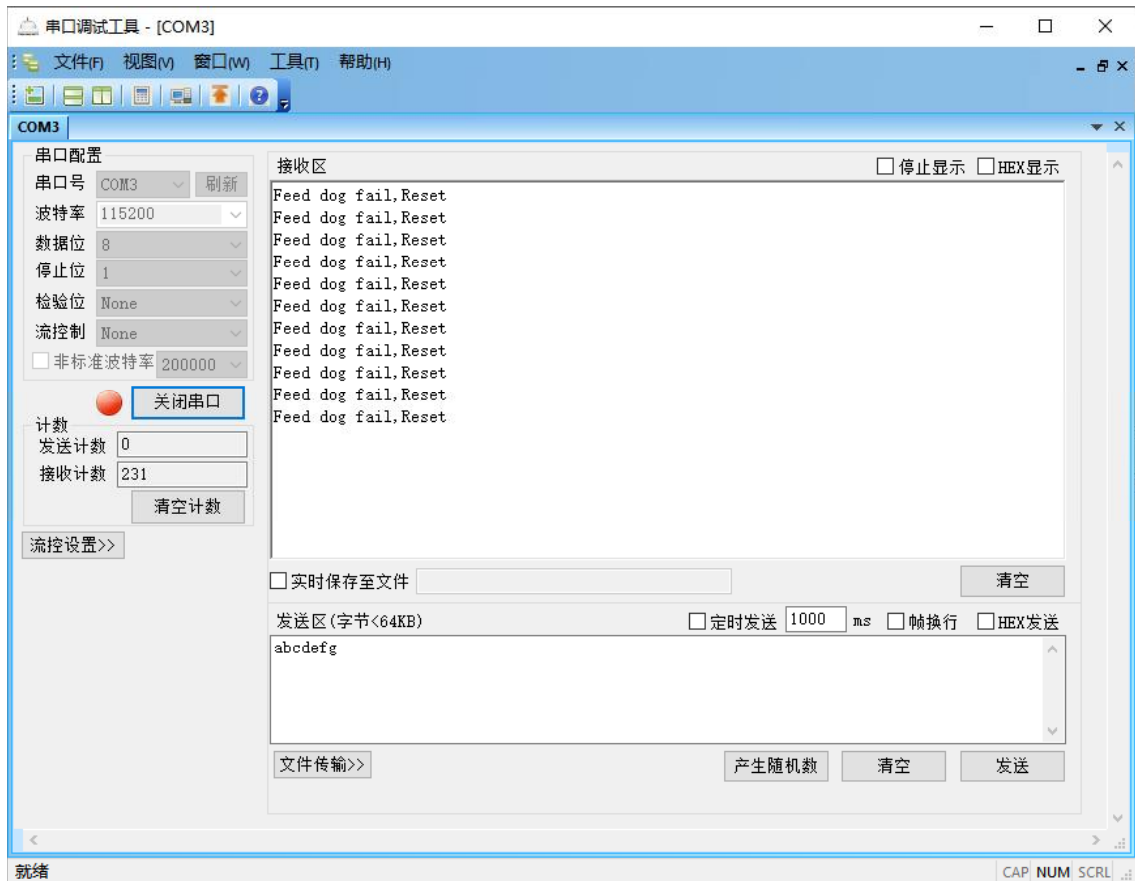
```
1. int main(void)
2. {
3.     USART_Printf_Init(115200);    //打印串口初始化，波特率为
        115200
4.     Delay_Init();                //延时函数初始化
5.     GPIOx_Init();                //相应引脚初始化
6.     IWDG_Feed_Init( IWDG_Prescaler_32, 4000 ); // 3.2s IWDG reset
7.
8.     while(1)
9.     {
10.        if( GPIOx_Connet() == 1 )    //PA0==1，即PA0连接高电
            平
11.        {
12.            IWDG_ReloadCounter();    //Feed dog
13.            printf("Feed dog success\r\n");
14.            Delay_Ms(10);
15.        }
16.        else
17.        {
18.            printf("Feed dog fail,Reset\r\n");
19.            Delay_Ms(1000);
20.        }
21.    }
22. }
```

复制代码

main.c文件主要是进行相关函数初始化以及进行喂狗操作。

4、下载验证

将编译好的程序下载到开发板并复位，打开串口调试助手，可看到串口调试助手一直在打印输出：Feed dog fail,Reset，如图所示：



当用杜邦线将PA0引脚与高电平连接进行喂狗操作，串口打印显示：
Feed dog success，具体如图所示：

