

文档版本	V1.0
发布日期	20210617

APT32S003 WWDT 应用指南

APTCHIP



目录

1 概述	1
2. 适用的硬件.....	1
3. 应用方案代码说明	1
3.1 WWDT 配置.....	1
3.2 WWDT 中断配置.....	3
4. 程序下载和运行	6

1 概述

本文介绍了在APT32S003中使用WWDT的应用范例。

2. 适用的硬件

该例程使用于 APT32S003 系列学习板

3. 应用方案代码说明

3.1 WWDT 配置

基于 APT32S003 完整的库文件系统，可以对 WWDT 进行配置。

硬件配置：

窗口型看门狗（WWDT）用于监测当前程序运行状况,它是一个 8 位的递减计数器，可以在预设时间周期结束时产生系统复位信号。

对看门狗计数器刷新时需要注意写 CNT 时，当前计数器值小于 WND 预设值，写入 CNT 的值必须在 0xFF 和 0x80 之间。

在连接 ICE Debugger 时，通过设置使能调试模式 CFGR[DBGEN]，将 WWDT 的工作时钟在调试暂停时挂起。

● 计算看门狗的溢出时间：

$$T_{WWDG} = TPCLK \times 4096 \times 2^{PSC} \times (CNT[6:0] + 1)$$

其中：TPCLK 为系统 PCLK 时钟周期，PSC 为 CFGR[PSC]的设置值，CNT 为 CR[CNT]寄存器设置值。

● 软件配置：

可在 apt32s003_initial.c 文件中 SYSCON_CONFIG 函数进行配置 WWDT。

```

/*****/

//syscon Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void SYSCON_CONFIG(void)
{
//-----SYSTEM CLK AND PCLK FUNTION-----/

    SYSCON_RST_VALUE(); //SYSCON all register clr

    SYSCON_General_CMD(ENABLE,ENDIS_ISOSC);

    SYSCON_HFOSC_SELECTE(HFOSC_SELECTE_48M); //HFOSC selected 48MHz

    //system clock set, Hclk div ,Pclk div set system clock=SystemCLK/Hclk div/Pclk div
    SystemCLK_HCLKDIV_PCLKDIV_Config(SYSCLK_HFOSC,HCLK_DIV_1,PCLK_DIV_1,HFOSC_48M);

//----- WDT FUNTION -----/

    //WDT TIME 1s,WDT alarm interrupt time=1s-1s*1/8=0.875S
    SYSCON_IWDCNT_Config(IWDT_TIME_2S,IWDT_INTW_DIV_7);

    SYSCON_WDT_CMD(DISABLE); //enable WDT

    SYSCON_IWDCNT_Reload(); //reload WDT

    //IWDT_Int_Enable();

//----- WWDT FUNTION -----/

    WWDT_CNT_Load(0xFF);

    WWDT_CONFIG(PCLK_4096_DIV0,0xFF,WWDT_DBGDIS);

    WWDT_Int_Config(ENABLE);

    //WWDT_CMD(ENABLE);

//----- CLO -----/

    //SYSCON_CLO_CONFIG(CLO_PA02);

    //SYSCON->OPT1=(SYSCON->OPT1&0xFFFF8000)|(0X01<<12)|(0X04<<8)|(0x00<<4);

//----- LVD FUNTION -----/

    //LVD LVR Enable/Disable
    SYSCON_LVD_Config(DISABLE_LVDEN,INTDET_LVL_3_3V,RSTDET_LVL_1_9V,DISABLE_LVD_INT,INTDET_POL_fall);

    //LVD_Int_Enable();

//----- EVTRG function -----/

    SYSCON->EVTRG=0X00|0x01<<20; //SYSCON_trgsrc0

    SYSCON->EVPS=0X00;

    SYSCON->IMER =EM_EVTRG0_ST;

//----- SYSCON Vector -----/

    //SYSCON_Int_Enable(); //SYSCON VECTOR

    //SYSCON_WakeUp_Enable(); //Enable WDT wakeup INT

}

/*****/

//WWDT Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

```

```
void WWDTHandler(void)
{
    if((WWDT->MISR&WWDT_EVI)==WWDT_EVI)
    {
        WWDT->ICR = WWDT_EVI;
    }
}
```

● 代码说明:

- WWDT_CNT_Load(); ----用于刷新计数器的值,清 WWDT
- WWDT_CONFIG(); ----用于配置 WWDT
- WWDT_Int_Config(); ----用于使能 WWDT 中断
- WWDT_CMD(); ----用于使能 WWDT
- WWDTHandler(); ----WWDT 中断服务函数

● 函数参数说明:



3.2 WWDT 应用

选择内部主频 48MHz 作为系统时钟,使能 WWDT 的中断, WWDT 计数器计数到 0x80 时, 可以产生中断,翻转 GPIO 测试并刷新 CNT, 避免复位。

```
/*-----*/
//syscon Functions
//EntryParameter:NONE
//Return Value:NONE
/*-----*/
```

```

void SYSCON_CONFIG(void)
{
//-----SYSTEM CLK AND PCLK FUNTION-----/
    SYSCON_RST_VALUE(); //SYSCON all register clr
    SYSCON_General_CMD(ENABLE,ENDIS_ISOSC);
    SYSCON_HFOSC_SELECTE(HFOSC_SELECTE_48M); //HFOSC selected 48MHz
    //system clock set, Hclk div ,Pclk div set system clock=SystemCLK/Hclk div/Pclk div
    SystemCLK_HCLKDIV_PCLKDIV_Config(SYSCLK_HFOSC,HCLK_DIV_1,PCLK_DIV_1,HFOSC_48M);
//----- WDT FUNTION -----/
    //WDT TIME 1s,WDT alarm interrupt time=1s-1s*1/8=0.875S
    SYSCON_IWDCNT_Config(IWDT_TIME_2S,IWDT_INTW_DIV_7);
    SYSCON_WDT_CMD(DISABLE); //enable WDT
    SYSCON_IWDCNT_Reload(); //reload WDT
    //IWDT_Int_Enable();
//----- WWDT FUNTION -----/
    WWDT_CNT_Load(0xFF);
    WWDT_CONFIG(PCLK_4096_DIV0,0xFF,WWDT_DBGDIS);
    WWDT_Int_Config(ENABLE);
    WWDT_CMD(ENABLE);
//----- CLO -----/
    //SYSCON_CLO_CONFIG(CLO_PA02);
    //SYSCON->OPT1=(SYSCON->OPT1&0xFFFF8000)|(0X01<<12)|(0X04<<8)|(0x00<<4);
//----- LVD FUNTION -----/
    SYSCON_LVD_Config(DISABLE_LVDEN,INTDET_LVL_3_3V,RSTDET_LVL_1_9V,DISABLE_LVD_INT,INTDET_POL_fall);
    //LVD_Int_Enable();
//----- EVTRG function -----/
    SYSCON->EVTRG=0X00|0x01<<20; //SYSCON_trgsrc0
    SYSCON->EVPS=0X00;
    SYSCON->IMER =EM_EVTRG0_ST;
//----- SYSCON Vector -----/
    //SYSCON_Int_Enable(); //SYSCON VECTOR
    //SYSCON_WakeUp_Enable(); //Enable WDT wakeup INT
}

void APT32S003_init(void)
{
    SYSCON->PCER0=0xFFFFFFFF; //PCLK Enable 0x00410071
    SYSCON->PCER1=0xFFFFFFFF; //PCLK Enable
    while(!(SYSCON->PCSR0&0x1)); //Wait PCLK enabled
//-----/
    //ISOSC/IMOSC/EMOSC/SYSCLK/IWDT/LVD/EM_CMFAIL/EM_CMRCV/CMD_ERR OSC stable interrupt
    //EntryParameter:NONE
    //Return Value:NONE
//-----/

```

```

SYSCON_CONFIG(); //syscon initial
CK_CPU_EnAllNormalIrq(); //enable all IRQ
//-----/
//Other IP config
//-----/
GPIO_Init(GPIOA0,4,0);
GPIO_Write_High(GPIOA0,4);
}

/*****/
//WWDT Interrupt
//EntryParameter:NONE
//Return Value:NONE
/*****/
void WWDTHandler(void)
{
    WWDT->ICR=0X01;
    WWDT_CNT_Load(0xFF);
    GPIO_Init(GPIOA0,4,0);
    GPIO_Reverse(GPIOA0,4);
    if((WWDT->MISR&WWDT_EVI)==WWDT_EVI)
    {
        WWDT->ICR = WWDT_EVI;
    }
}

/*****/
//main
/*****/
int main(void)
{
    APT32S003_init();
    //
    while(1)
    {

        //SYSCON_IWDCNT_Reload();
    }
}

```

● 波形验证：

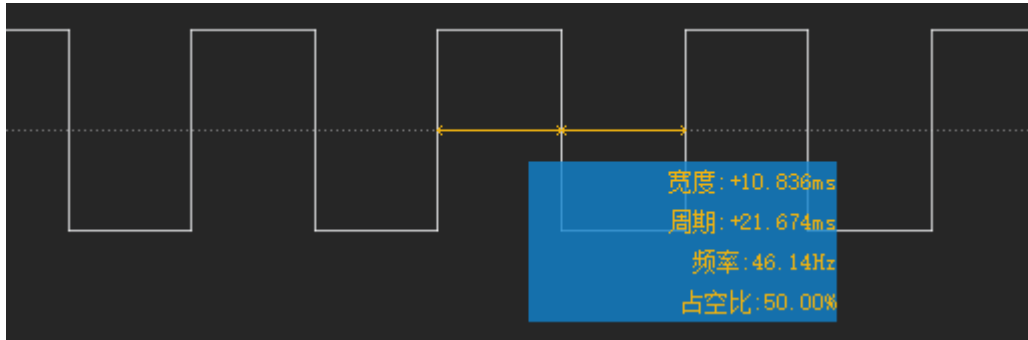


图 3.2.1 WWDT 定时翻转

- 溢出时间计算:

$$\begin{aligned} \text{TWWDG} &= \text{TPCLK} \times 4096 \times 2^{\text{PSC}} \times (\text{CNT}[6:0] + 1) \\ &= (1/48\text{M}) \times 4096 \times 2^0 \times (0\text{xFF} + 1) \\ &= 21\text{MS} \end{aligned}$$

4. 程序下载和运行

1. 将目标板与仿真器连接，分别为 VDD SCLK SWIO GND
2. 程序编译后仿真运行
3. 通过图 3.2.1 验证 WWDT 实现翻转 GPIO