

文档版本	V1.0
发布日期	20191108

APT32F172 STC 应用开发指南



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 STC 定时配置	1
3.2 STC 捕捉输入配置	3
4 程序下载和运行	7
5 改版历史	8

1 概述

本文介绍了在APT32F172中使用STC的应用范例。

2. 适用的硬件

该例程使用于 APT32F172 开发板 APT-DB172

3. 应用方案代码说明

基于 APT32F172 完整的库文件系统，可以很方便的对 STC 进行配置。

3.1 STC 定时配置

软件配置：

开启内部主频 20MHz，并作为系统时钟。

计数器单周期时间： $Stc16_TimeClk = PCLK / 2^{10} / 20 = 1\mu s$

PB0.0 捕捉高电平 50us，低电平 50us 方波。

```
/*  
//STC Functions  
//EntryParameter:NONE  
//ReturnValue:NONE  
*/  
void STC16_CONFIG(void)  
{  
    STC16_RESET_VALUE();    //STC16 所有寄存器复位赋值  
    STC16_Softreset();      //STC16 软件复位  
    STC16_Clk_CMD(ENABLE); //STC16 时钟使能  
    STC16_Configure(STC16_Count_mode_Continue,STC16_Count_STOPTYPE_StopConmand,STC16_CM0_Mode_  
    Capture,STC16_CM1_Mode_Match,19,0);  
    //STC16 Configure;设置为连续计数模式;单次计数模式下,选择立即停止模式;通道 0 作为匹配模式;通  
    道 1 作为捕捉模式;stc16_timeclk=pclk/(19+1)/2^0=1us  
    STC16_CNR_CC0_CC1_Load(50,0,0); //STC16 CNTR=50,CC0R=0,CC1R=0  
    STC16_MINT_CMD(ST16_PENDI,ENABLE);//STC16 周期结束中断使能  
    STC16_Start();          //Start stc16  
    STC16_Int_Enable();     //ENABLE STC16 中断向量  
}
```

代码说明:

STC16_Configure(STC16_Count_mode_Continue, STC16_Count_STOPTYPE_StopCommand,
STC16_CM0_Mode_Capture, STC16_CM1_Mode_Match, 19, 0);

STC16_Count_mode_Continue-----连续计数模式

STC16_CM0_Mode_Capture-----通道 0 作为捕捉模式

时钟公式----- $stc16_timeclk = pclk / (19 + 1) / 2^0 = 1\mu s$

```

/*****/
//APT32F172_init
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void APT32F172_init(void)
{
    SYSCON_WDT_CMD(DISABLE);           //关闭 WDT

    SYSCON->PCER0=0xFFFFFFFF;         //使能 IP
    SYSCON->PCER1=0xFFFFFFFF;         //使能 IP
    while(!(SYSCON->PCSR0&0x1));       //判断 IP 是否使能

    SYSCON_Int_Enable();               //使能 SYSCON 中断向量
    SYSCON->IECR=ISOSC_ST|IMOSC_ST|EMOSC_ST|SYSCLK_ST;
    //使能 ISOSC 时钟稳定中断,使能 IMOSC 时钟稳定中断,使能 EMOSC 时钟稳定中断

    CK_CPU_EnAllNormalIrq();           //打开全局中断
    SYSCON_CONFIG();                   //syscon 参数 初始化

    GPIO_CONFIG();                     //GPIO 初始化
    STC16_CONFIG ();                   //stc16 初始化
}
/*****/
//STC16 Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void STC16IntHandler(void)
{
    if((ST16->MISR&ST16_PENDI)==ST16_PENDI)
    {
        ST16->ICR = ST16_PENDI;
        if(!f_io_toggle)
        {
            f_io_toggle=1;
        }
    }
}
    
```

```

        GPIO_Write_High(GPIOB0,0);
    }
    else
    {
        f_io_toggle=0;
        GPIO_Write_Low(GPIOB0,0);
    }
}
}

```

3.2 STC 捕捉输入配置

开启内部主频 20MHz, 并作为系统时钟。

计数器单周期时间: $Stc16_TimeClk = PCLK / 2^0 / 20 = 1\mu s$

PB0.0 捕捉高电平 50us, 低电平 50us 方波。

R_LowLevel_T 存储低电平时间。

R_HighLevel_T 存储高电平时间。

/*

*/

//stc16 Functions

//EntryParameter:NONE

//ReturnValue:NONE

/*

*/

void STC16_CONFIG(void)

{

STC16_RESET_VALUE(); *//STC16 所有寄存器复位赋值*

STC16_Softreset(); *//STC16 软件复位*

STC16_IO_Init(STC16_IO_CAP1,0); *//STC16 CAP1 初始化*

STC16_Clk_CMD(ENABLE); *//STC16 时钟使能*

STC16_Channel1_CMD(ENABLE); *//STC16 通道1 使能*

STC16_Configure(STC16_Count_mode_Continue,STC16_Count_STOPTYPE_StopCommand,STC16_CM0_Mode_Match,STC16_CM1_Mode_Capture,19,0);

//STC16 Configure; 设置为连续计数模式; 单次计数模式下, 选择立即停止模式; 通道0 作为匹配模式; 通道1 作为捕捉模式; stc16_timeclk=pclk/(19+1)/2^0=1us

STC16_CNR_CC0_CC1_Load(0xffff,0,0); *//STC16 CNTR=0xffff,CC0R=0,CC1R=0*

```

        STC16_CINT_CMD(ST16_CC1RI,ENABLE);    //STC16 通道1 上升沿中断使能
        STC16_CINT_CMD(ST16_CC1FI,ENABLE);    //STC16 通道1 下降沿中断使能
        STC16_Start();                        //Start stc16
        STC16_Int_Enable();                   //ENABLE STC16 中断向量
    }
    
```

代码说明:

作为捕捉模式 CNTR 一定要设置为 0xFFFF

```

STC16_Configure(STC16_Count_mode_Continue, STC16_Count_STOPTYPE_StopCommand, ST
C16_CM0_Mode_Match, STC16_CM1_Mode_Capture, 19, 0);
    
```

STC16_CM1_Mode_Match-----通道1 作为匹配模式

时钟公式----- $stc16_timeclk = pclk / (19 + 1) / 2^0 = 1\mu s$

```

/*****/
    
```

```

//syscon Functions
    
```

```

//EntryParameter:NONE
    
```

```

//ReturnValue:NONE
    
```

```

/*****/
    
```

```

void SYSCON_CONFIG(void)
    
```

```

{
    
```

```

    //-----SYSTEM CLK AND PCLK FUNTION-----/
    
```

```

    SYSCON_RST_VALUE();                //SYSCON 所有寄存器复位赋值
    
```

```

    SYSCON_General_CMD(ENABLE,ENDIS_IDLE_PCLK);    //SLEEP 模式下 PCLK 使能
    
```

```

    SYSCON_General_CMD(ENABLE,ENDIS_ISOSC);        //使能内部副频
    
```

```

    SYSCON_General_CMD(ENABLE,ENDIS_IMOSC);        //使能内部主频
    
```

```

    SYSON_EMOSC_32k_EN();                //使能 外部晶振外接 32.768K
    
```

```

    SYSCON_IMOSC_SETECTE(IMOSC_SETECTE_20M);      //选择内部主频为 20M
    
```

```

    SystemCLK_HCLKDIV_PCLKDIV_Config(SYSCON_IMOSC,HCLK_DIV_1,PCLK_DIV_1);
    
```

```

    //内部主振作为系统时钟, HCLK 1 分频, PCLK 1 分频
    
```

```

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

```

    SYSCON_IWDCNT_Config(IWDT_TIME_1S,IWDT_INTW_DIV_1);
    
```

```

    //WDT 溢出时间 1s;WDT TEIM:1S*(1-(8-1)/8)=0.75S
    
```

```

        SYSCON_WDT_CMD(DISABLE);

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

        LVR_Disable();

    }

/*****/

//APT32F172_init
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void APT32F172_init(void)
{
    SYSCON_WDT_CMD(DISABLE);           //关闭 WDT

    SYSCON->PCER0=0xFFFFFFFF;          //使能 IP
    SYSCON->PCER1=0xFFFFFFFF;          //使能 IP
    while(!(SYSCON->PCSR0&0x1));        //判断 IP 是否使能

    SYSCON_Int_Enable();                //使能 SYSCON 中断向量
    SYSCON->IECR=ISOSC_ST|IMOSC_ST|EMOSC_ST|SYSCLK_ST;
    //使能 ISOSC 时钟稳定中断,使能 IMOSC 时钟稳定中断,使能 EMOSC 时钟稳定中断

    CK_CPU_EnAllNormalIrq();           //打开全局中断
    SYSCON_CONFIG();                   //syscon 参数 初始化

    STC16_CONFIG ();                   //stc16 初始化
}

volatile U32_T R_LowLevel_T,R_HighLevel_T;
/*****/
    
```

```
//STC16 Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void STC16IntHandler(void)
{
    // ISR content ...
    if((ST16->CMISR&ST16_CC1RI)==ST16_CC1RI)
    {
        ST16->CICR = ST16_CC1RI;
        R_LowLevel_T=ST16->CC1R;
        STC16_stop();
        STC16_Channel1_Capture_LoadMode_set(STC16_C1SR_CaptureFall);
        //stc16 Channel0 下降沿捕捉
        STC16_Start();
    }
    if((ST16->CMISR&ST16_CC1FI)==ST16_CC1FI)
    {
        ST16->CICR = ST16_CC1FI;
        R_HighLevel_T=ST16->CC1R;
        STC16_stop();
        STC16_Channel1_Capture_LoadMode_set(STC16_C1SR_CaptureRise);
        //stc16 Channel0 下降沿捕捉
        STC16_Start();
    }
}
```


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. 定时测试将示波器挂在对应 TOGGLE IO 上
3. 将需检测波形输出接到捕捉口上
4. 程序编译后仿真运行
5. 定时和 PWM 观察示波器波形，捕捉功能观察 R_LowLevel_T、R_HighLevel_T 变量值是否与输入波形的周期和占空比匹配

5. 改版历史

版本	修改日期	修改概要
V1.0	2019-11-08	初版