文档版本	V1.0
发布日期	20191108

APT32F172 GPT 应用开发指南

目录

1 概述

本文介绍了在APT32F172中使用GPT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 GPT 进行配置。

3.1 GPT 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：CLKS= MCLK /1=20MHz

PB0.0 输出周期 100us，占空比 50us 方波

```
/*************************************************************/
//GPT Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*************************************************************/
void GPT_CONFIG(void)
{
    GPT_RESET_VALUE(GPTCH0);                    //GPT0 所有寄存器复位赋值
    GPTCHX_Clk_CMD(GPTCH0,ENABLE);              //GPT0 时钟使能
    GPTCHX_SoftwareReset(GPTCH0);               //GPT0 软件复位
    GPTCHX_CLK_Configure(GPTCH0,GPT_Mclk_Selecte_Pclk,GptClks_MCLK_DIV1,GPTCHX_CLKI_0,GPTCHX_B
    URST
    SET_None);
    //GTP0 选择 PCLK 作为 MCLK;CLKS=MCLK/1;CLK 上升沿计数;关闭群脉冲模式
    GPTCHX_COUNT_Configure(GPTCH0,CPC_Reload_ENABLE); //GPT0 RC 匹配重新计数
    GPTCHX_Set_RA_RB_RC(GPTCH0,0,0,1000);       //GPT0 RA=0,RB=0,RC=1000
    GPTCHX_ConfigInterrupt_CMD(GPTCH0,GPTCHX_INT_CPCS,ENABLE);//使能 GPT0 比较寄存器C 匹配中断
    GPTCHX_CountClk_CMD(GPTCH0,ENABLE);         //使能 GPT0 计数时钟
    GPTCHX_SWTRG(GPTCH0);                       //软件触发 GPT0
```

```
    GPTCH0_Int_Enable();                                    //使能 GPT0 中断向量
}
```

代码说明：

GPTCHX_CLK_Configure(GPTCH0,GPT_Mclk_Selecte_Pclk,GptClks_MCLK_DIV1,GPTCHX_CLKI_0,GPTCHX_BURST_SET_None);---------选择 PCLK 作为 MCLK 时钟，分频选择 DIV1，
GPTCHX_COUNT_Configure(GPTCH0,CPC_Reload_ENABLE); ---------RC 匹配重新计数配置
GPTCHX_CountClk_CMD(GPTCH0,ENABLE) ---------时钟使能
GPTCHX_SWTRG(GPTCH0); ---------触发时钟开始计数
GPTCHX_CountClk_CMD(GPTCH0,DISABLE) ---------停止计数
若下次需要重新启动需要调用 GPTCHX_SWTRG(GPTCH0);

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

SYSCON->PCER0=0xFFFFFFF;                        //使能 IP
SYSCON->PCER1=0xFFFFFFF;                        //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
GPT_CONFIG ();                                  //UART 初始化
}

volatile U32_T f_io_toggle;
/*************************************************************/
//GPT_0 Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*************************************************************/
void GPT_0IntHandler(void)
{
    if((GPTCH0->SR&GPTCHX_INT_CPCS)==GPTCHX_INT_CPCS)
```

```
    {
        GPTCH0->CSR = GPTCHX_INT_CPCS;
        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 GPT PWM 输出配置

开启内部主频 20MHz，并作为系统时钟。

计数器单周期时间：CLKS= MCLK /1=20MHz

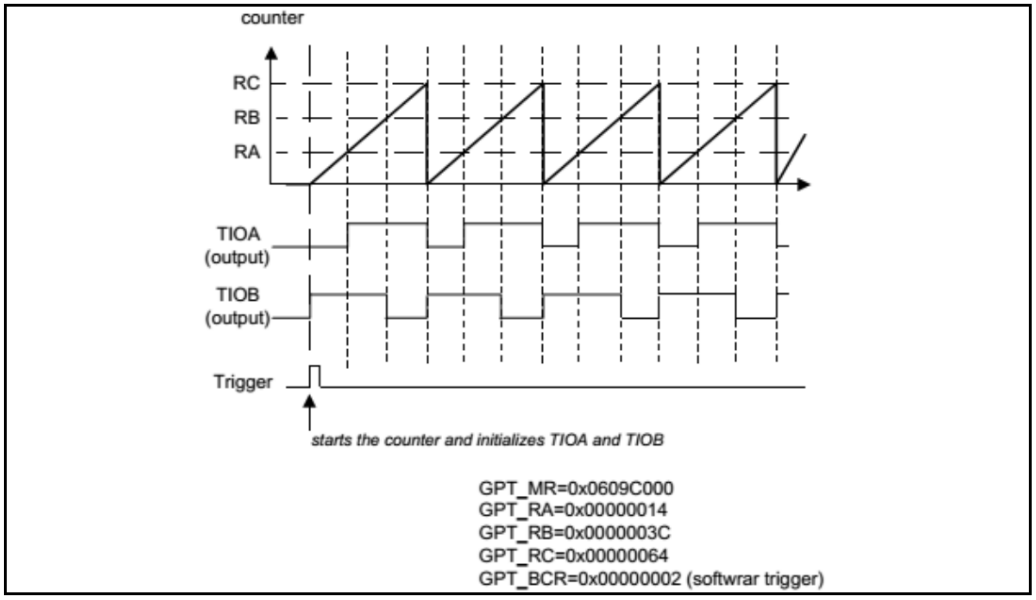
PB0.1 输出周期 100us，占空比 50us 方波

```
/*******************************************************************************/
//APT32F172_init
//EntryParameter:NONE
//ReturnValue:NONE
/*******************************************************************************/
void GPT_CONFIG(void)
{
    GPT_RESET_VALUE(GPTCH0);                    //GPT0 所有寄存器复位赋值
    GPT_IO_Init(GPT_IO_IO0A,0);                 //IO0A (PB0.1)做 PWM 输出口
    GPTCHX_Clk_CMD(GPTCH0,ENABLE);              //GPT0 时钟使能
    GPTCHX_SoftwareReset(GPTCH0);               //GPT0 软件复位
    GPTCHX_CLK_Configure(GPTCH0,GPT_Mclk_Selecte_Pclk,GptClks_MCLK_DIV1,GPTCHX_CLKI_0,GPTCHX_B
    URST_SET_None);
    //GTP0 选择 PCLK 作为 MCLK;CLKS=MCLK/1;CLK 上升沿计数;关闭群脉冲模式
    GPTCHX_COUNT_Configure(GPTCH0,CPC_Reload_ENABLE); //GPT0 RC 匹配重新计数
    GPTCHX_PWM_Configure(GPTCH0,CPC_STOP_DISABLE,CPC_DisCountClk_DISABLE,CPC_Reload_ENABLE,E
    EVT_Reload_DISABLE,EEVT_XC0_NONE,TIOA_SWTRG_OutPut_High,TIOA_EEVT_OutPut_NoChange,TIOA_CP
    A_OutPut_Low,TIOA_CPC_OutPut_High,TIOB_SWTRG_OutPut_High,TIOB_EEVT_OutPut_NoChange,TIOB_CPB
    _OutPut_Low,TIOB_CPC_OutPut_High);
    //GPT0RC 匹配停止计数禁止;RC 匹配停止计数时钟禁止;RC 匹配重新计数禁止;外部事件触发重新计
    数禁止;外部事、//件 XC0 选择禁止;软件触发 TIOA 为高电平;外部事件触发 TIOA 不改变;RA 匹配 TIOA
    输出低电平;RC 匹配 TIOA
```

```
//输出高;软件触发 TIOB 为高电平;外部事件触发 TIOB 不改变;RB 匹配 TIOB 输出低电平;RC 匹配 TIOB 输出高
GPTCHX_Set_RA_RB_RC(GPTCH0,1000,0,2000);          //GPT0 RA=1000,RB=0,RC=2000
GPTCHX_CountClk_CMD(GPTCH0,ENABLE);               //使能 GPT0 计数时钟
GPTCHX_SWTRG(GPTCH0);                             //软件触发 GPT0
}
```

波形翻转示意图如下：

TIOA由RA和RC的值决定翻转，TIOB由RB和RC决定。



代码范例：

```
GPTCHX_PWM_Configure(GPTCH0,CPC_STOP_DISABLE,CPC_DisCountClk_DISABLE,CPC_Reload_ENABLE,EEVT_Reload_DISABLE,EEVT_XCO_NONE,TIOA_SWTRG_OutPut_High,TIOA_EEVT_OutPut_NoChange,TIOA_CPA_OutPut_Low,TIOA_CPC_OutPut_High,TIOB_SWTRG_OutPut_High,TIOB_EEVT_OutPut_NoChange,TIOB_CPB_OutPut_Low,TIOB_CPC_OutPut_High);
GPTCH0----------选择计数器 0
CPC_STOP_DISABLE----------RC 匹配停止禁止
CPC_DisCountClk_DISABLE----------RC 匹配停止计数时钟禁止
CPC_Reload_ENABLE----------RC 匹配重载使能
EEVT_Reload_DISABLE----------外部事件触发重新计数禁止
EEVT_XCO_NONE----------外部事件 XCO 选择禁止
TIOA_SWTRG_OutPut_High----------软件触发 TIOA 为高电平
TIOA_EEVT_OutPut_NoChange----------外部事件触发 TIOA 不改变
TIOA_CPA_OutPut_Low----------RA 匹配 TIOA 输出低电平
TIOA_CPC_OutPut_High----------RC 匹配 TIOA 输出高
TIOB_SWTRG_OutPut_High----------软件触发 TIOB 为高电平
TIOB_EEVT_OutPut_NoChange----------外部事件触发 TIOB 不改变
TIOB_CPB_OutPut_Low ----------RA 匹配 TIOB 输出低电平
TIOB_CPC_OutPut_High----------RC 匹配 TIOB 输出高
```

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

SYSCON->PCER0=0xFFFFFFF;                    //使能 IP
SYSCON->PCER1=0xFFFFFFF;                    //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

GPT_CONFIG ();                              //GPT 初始化
}
```

3.3 GPT 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：CLKS= MCLK /1=20MHz。

PA0.11 捕捉周期 100us，占空比为 50us 方波。

RA_Capture 存储高电平计数。

RB_Capture 存储周期计数。

```
/*******************************************************************************/
//APT32F172_init
//EntryParameter:NONE
//ReturnValue:NONE
/*******************************************************************************/
void GPT_CONFIG(void)
{
    GPT_RESET_VALUE(GPTCH1);                    //GPT1 所有寄存器复位赋值
```

```
    GPT_IO_Init(GPT_IO_IO1A,0);                    //IO1A 初始化
    GPTCHX_Clk_CMD(GPTCH1,ENABLE);                 //GPT1 时钟使能
    GPTCHX_SoftwareReset(GPTCH1);                  //GPT1 软件复位
    GPTCHX_CLK_Configure(GPTCH1,GPT_Mclk_Selecte_Pclk,GptClks_MCLK_DIV1,GPTCHX_CLKI_0,GPTCHX_B
    URST_SET_None);
    //GTP1 选择 PCLK 作为 MCLK;CLKS=MCLK/1;CLK 上升沿计数;关闭群脉冲模式
    GPTCHX_Capture_Configure(GPTCH1,LDB_STOP_ENABLE,LDB_DisCountClk_DISABLE,ABETRG_TIOA_Rise,C
    PC_Reload_DISABLE,LDRA_TIOA_Fall,LDRB_TIOA_Rise);
      //GPT1;RB 载入停止计数使能;RB 载入停止时钟禁止;TIOA 上升沿触发重启计数;RC 匹配重新计数使
      能;RA 在 TIOA 下降沿载入;RB 在 TIOA 上升沿载入
    GPTCHX_ConfigInterrupt_CMD(GPTCH1,GPTCHX_INT_LDRAS,ENABLE);//使能 GPT1 载入寄存器 A 中断
    GPTCHX_ConfigInterrupt_CMD(GPTCH1,GPTCHX_INT_LDRBS,ENABLE);//使能 GPT1 载入寄存器 B 中断
    GPTCHX_CountClk_CMD(GPTCH1,ENABLE);        //使能 GPT1 计数时钟
    GPTCHX_SWTRG(GPTCH1);                      //软件触发 GPT1
    GPTCH1_Int_Enable();                       //使能 GPT1 中断向量
}
/*******************************************************************************/
//APT32F172_init
//EntryParameter:NONE
//ReturnValue:NONE
/*******************************************************************************/
void APT32F172_init(void)
{
    SYSCON_WDT_CMD(DISABLE);                   //关闭 WDT

    SYSCON->PCER0=0xFFFFFFF;                   //使能 IP
    SYSCON->PCER1=0xFFFFFFF;                   //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

    GPT_CONFIG ();                              //GPT 初始化
  }

volatile U32_T RA_Capture,RB_Capture；
/*************************************************************/
//GPT_0 Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*************************************************************/
void GPT_1IntHandler (void)
{
    if((GPTCH1->SR&GPTCHX_INT_LDRAS)==GPTCHX_INT_LDRAS)
    {
        GPTCH1->CSR = GPTCHX_INT_LDRAS;
        RA_Capture=GPTCH1->RA;
    }
    if((GPTCH1->SR&GPTCHX_INT_LDRBS)==GPTCHX_INT_LDRBS)
    {
        GPTCH1->CSR = GPTCHX_INT_LDRBS;
        RB_Capture=GPTCH1->RB;
    }
}
```

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. PWM 输出将示波器挂在对应 PWM 输出口上
4. 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 RA_Capture,RB_Capture 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