

文档版本	V1.0
发布日期	20210609

APT32F102 BT 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 BT 配置.....	1
3.2 定时中断配置.....	3
3.3 PWM 输出配置	6
4. 程序下载和运行	8

1 概述

本文介绍了在APT32F102中使用SPI的应用范例。

2. 适用的硬件

该例程使用于 APT32F102x 系列学习板

3. 应用方案代码说明

3.1 BT 配置

基于 APT32F102 完整的库文件系统，可以对 BT 进行配置。

- 硬件配置：

BT 模块是一个具有自动重载 16 位可编程递增计数器，计数器从零开始计数。有一个比较器支持 PWM 输出。BT 的计数器的启动可以通过 ETCB 进行硬件上的触发和软件上使能触发。

- 管脚描述：

管脚名称	功能描述
BT_OUT	PWM 波形输出

图 3.1.1 输出管脚

- 模块框图：

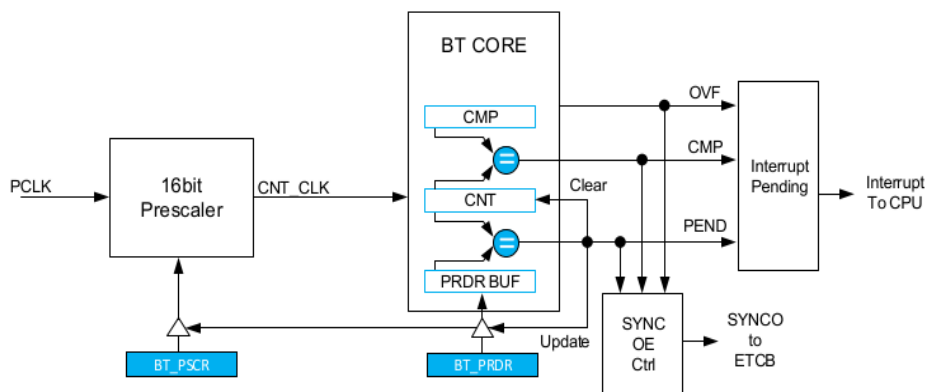


图 3.1.2 BT 框图

- 软件配置:

可在 apt32f102_initial.c 文件中进行 BT_CONFIG 初始化的配置(BT0 表示第一个 BT 模块)

```

/*****/
//BT Initial
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void BT_CONFIG(void)
{
    BT_DeInit(BT0);
    //BT_IO_Init(BT0_PB05);
    BT_Configure(BT0,BTCLK_EN,47,BT_IMMEDIATE,BT_CONTINUOUS,BT_PCLKDIV);//TCLK=PCLK/(0+1)
    BT_ControlSet_Configure(BT0,BT_START_HIGH,BT_IDLE_LOW,BT_SYNC_DIS,BT_SYNCMD_DIS,BT_OSTMDX_CONTINUOUS,BT_AREARM_DIS,BT_CNTRLD_EN);
    //BT_Trigger_Configure(BT0,BT_TRGSRC_PEND,BT_TRGOE_EN);
    BT_Period_CMP_Write(BT0,100,1);
    BT_Start(BT0);
    BT_ConfigInterrupt_CMD(BT0,ENABLE,BT_PEND);
    BT0_INT_ENABLE();

    /* BT_DeInit(BT1);
    BT_IO_Init(BT1_PA06);
    BT_Configure(BT1,BTCLK_EN,0,BT_IMMEDIATE,BT_CONTINUOUS,BT_PCLKDIV);
    BT_ControlSet_Configure(BT1,BT_START_LOW,BT_IDLE_HIGH,BT_SYNC_DIS,BT_SYNCMD_DIS,BT_OSTMDX_ONCE,BT_AREARM_DIS,BT_CNTRLD_EN);
    BT_Trigger_Configure(BT1,BT_TRGSRC_PEND,BT_TRGOE_EN);
    BT_Period_CMP_Write(BT1,1000,500);
    BT_Start(BT1);
    BT_ConfigInterrupt_CMD(BT1,BT_PEND,ENABLE);
    BT1_INT_ENABLE();*/
}

```

- 代码说明:

BT_DeInit(BT0); ----用于配置 BT 默认值

BT_IO_Init(); ----用于配置 GPIO 为 BT 功能

BT_Configure(); ----用于配置时钟

BT_ControlSet_Configure();----用于配置 BT 模式及触发

BT_Trigger_Configure();----用于配置 BT 触发源选择

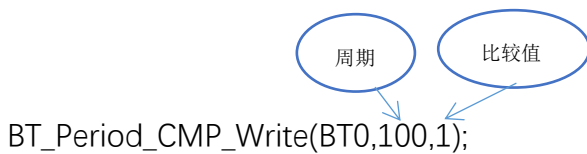
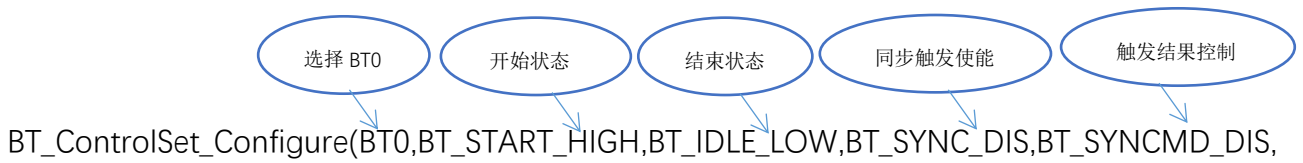
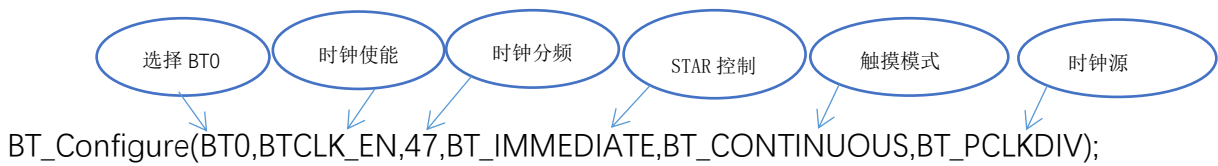
BT_Period_CMP_Write();----用于配置 BT 周期

BT_Start();----用于使能 BT

BT_ConfigInterrupt_CMD();----用于配置 BT 中断

BT0_INT_ENABLE();----用于使能 BT 中断

● 参数说明:



3.2 定时中断配置

系统时钟选择内部 48Mhz，利用 BT 定时使 GPIOA0.14 输出周期为 100us，占空比为 50% 的波形。

```

void BT_CONFIG(void)
{
    BT_DelInit(BT0);

    //BT_IO_Init(BT0_PB05);

    BT_Configure(BT0,BTCLK_EN,47,BT_IMMEDIATE,BT_CONTINUOUS,BT_PCLKDIV);//TCLK=PCLK/(0+1)

    BT_ControlSet_Configure(BT0,BT_START_HIGH,BT_IDLE_LOW,BT_SYNC_DIS,BT_SYNCMD_DIS,BT_OSTMDX_CONTINUOUS,BT_AREARM_DIS,BT_CNTRLD_EN);

    //BT_ControlSet_Configure(BT0,BT_START_HIGH,BT_IDLE_LOW,BT_SYNC_EN,BT_SYNCMD_DIS,BT_OSTMDX_ONCE,BT_AREARM_DIS,BT_CNTRLD_EN);

    //BT_Trigger_Configure(BT0,BT_TRGSRC_PEND,BT_TRGOE_EN);

    BT_Period_CMP_Write(BT0,100,1);

    BT_Start(BT0);

    BT_ConfigInterrupt_CMD(BT0,ENABLE,BT_PEND);

    BT0_INT_ENABLE();
}

/*****/
//GPIO Initial
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

void GPIO_CONFIG(void)
{
    GPIO_Init(GPIOA0,14,0);
}

void APT32F102_init(void)
{
    //-----/

    //Peripheral clock enable and disable
    //EntryParameter:NONE
    //ReturnValue:NONE
    //-----/

    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
    //ReturnValue:NONE
    //-----/

    SYSCON_CONFIG(); //syscon initial
    CK_CPU_EnAllNormalIrq(); //enable all IRQ
    //-----/

    //Other IP config

```

```
//-----/
GPIO_CONFIG();                               //GPIO initial
BT_CONFIG();                                 //BT initial
SYSCON_INT_Priority();
}
/*****/
//BT0 Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void BT0IntHandler(void)
{
    // ISR content ...
    static U8_T temp;
    if((BT0->MISR&BT_PEND)==BT_PEND)
    {
        BT0->ICR = BT_PEND;
        if(temp==0)
        {
            temp = 1;
            GPIO_Write_Low(GPIOA0,14);
        }
        else
        {
            temp = 0;
            GPIO_Write_High(GPIOA0,14);
        }
    }
    else if((BT0->MISR&BT_CMP)==BT_CMP)
    {
        BT0->ICR = BT_CMP;
    }
    else if((BT0->MISR&BT_OVF)==BT_OVF)
    {
        BT0->ICR = BT_OVF;
    }
    else if((BT0->MISR&BT_EVTRG)==BT_EVTRG)
    {
        BT0->ICR = BT_EVTRG;
    }
}
}
```

● 输出波形:

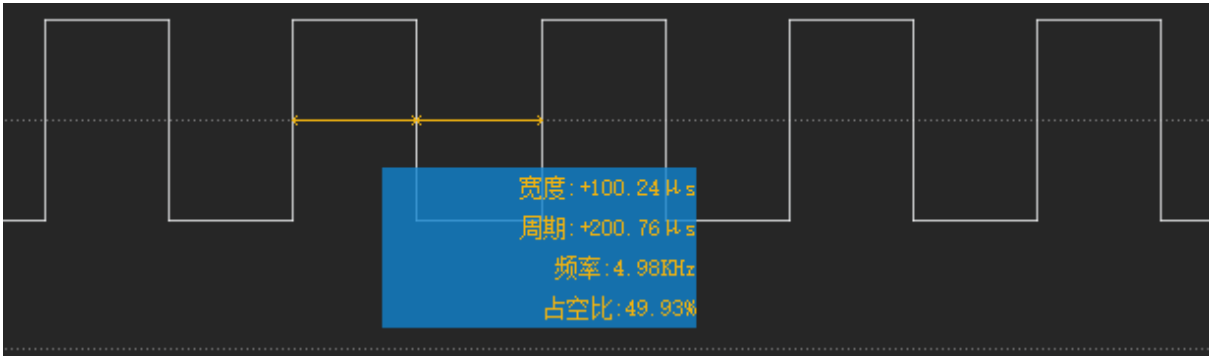


图 3.2.1 定时输出

3.3 PWM 输出配置

系统时钟选择内部 48Mhz，启用 BT 的 PWM 模式，通过 PB0.5，输出周期 200uS，占空比 50%。

```

/*****/
//BT Initial
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void BT_CONFIG(void)
{
    BT_DeInit(BT0);
    BT_IO_Init(BT0_PB05);
    BT_Configure(BT0,BTCLK_EN,47,BT_IMMEDIATE,BT_CONTINUOUS,BT_PCLKDIV);//TCLK=PCLK/(0+1)
    BT_ControlSet_Configure(BT0,BT_START_HIGH,BT_IDLE_LOW,BT_SYNC_DIS,BT_SYNCMD_DIS,BT_OSTMDX_CONTINUOUS,BT_AREARM_DIS,BT_CNTRLD_EN);
    //BT_ControlSet_Configure(BT0,BT_START_HIGH,BT_IDLE_LOW,BT_SYNC_EN,BT_SYNCMD_DIS,BT_OSTMDX_ONCE,BT_AREARM_DIS,BT_CNTRLD_EN);
    //BT_Trigger_Configure(BT0,BT_TRGSRC_PEND,BT_TRGOE_EN);
    BT_Period_CMP_Write(BT0,200,100);
    BT_Start(BT0);
}

void APT32F102_init(void)
{
    //-----/
    //Peripheral clock enable and disable
    //EntryParameter:NONE
    //ReturnValue:NONE
    
```



```
//-----/
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
//ReturnValue:NONE
//-----/
SYSCON_CONFIG(); //syscon initial
CK_CPU_EnAllNormalIrq(); //enable all IRQ
//-----/
//Other IP config
//-----/
BT_CONFIG(); //BT initial
}
/*****/
//main
/*****/
int main(void)
{
//delay_nms(5000);
APT32F102_init();
while(1)
{
SYSCON_IWDCNT_Reload();
//.....
}
}
}
```

● 输出波形:

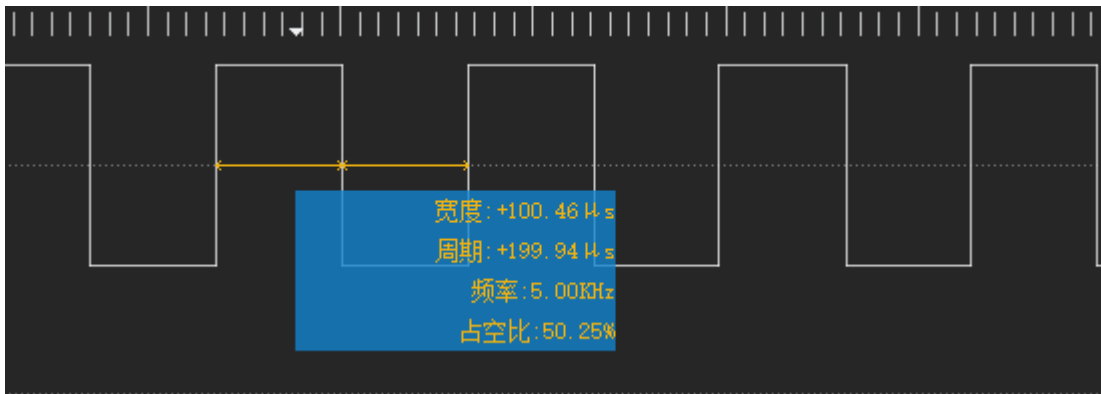


图 3.3.1 PWM 输出

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. 通过示波器或逻辑分析仪查看输出波形。如上图 3.2.1、图 3.3.1 所示