

文档版本	V1.0
发布日期	20210621

# APT32S003 COUNTA 应用指南

**APT**CHIP



## 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 COUNTA 配置.....	1
3.2 BUZZ 输出.....	3
3.3 定时输出.....	4
3.4 载波输出.....	6
4. 程序下载和运行 .....	8

# 1 概述

本文介绍了在APT32S003中COUNTA应用

# 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 COUNTA 配置

- 硬件配置:

COUNTA 模块是一个 16 位的计数器/定时器，具有自动重载功能，可以产生一个计数器 A 的中断，可以用来产生载波频率，驱动蜂鸣器。

- 模块框图:

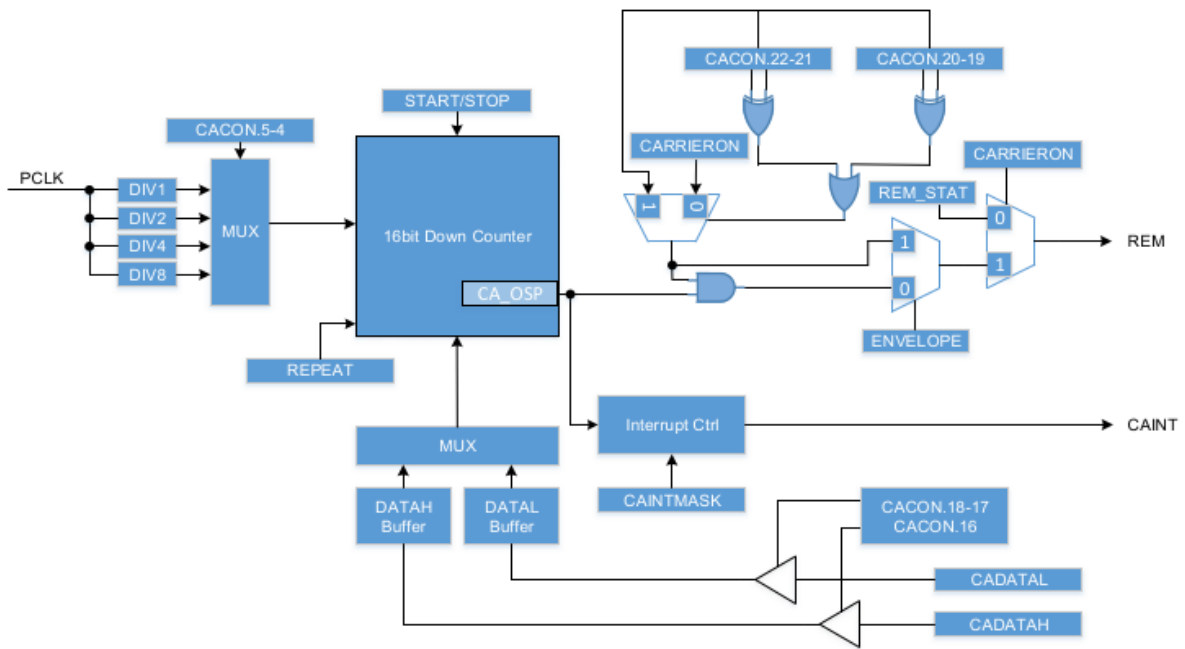


图 3.3.1 COUNTA 模块框图

**软件配置:**

可在 apt32S003\_initial.c 文件中 COUNTA\_CONFIG()进行初始化的配置.

```

/*****/
//COUNTA Initial
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

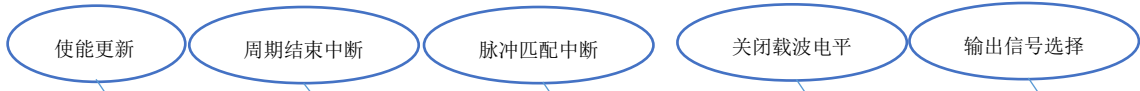
void COUNTA_CONFIG(void)
{
    COUNT_DeInit(); //clear all countA Register
    COUNTA_IO_Init(COUNTA_PB01); //set PB0.01 as counter IO
    COUNTA_Init(24000,24000,Period_H,DIV1,REPEAT_MODE,CARRIER_ON,OSP_LOW); //Data_H=Time/(1/sysclock)
    COUNTA_Config(SW_STROBE,PENDREM_OFF,MATCHREM_OFF,REMSTAT_0,ENVELOPE_0); //countA mode set
    COUNTA_Start(); //countA start
    //COUNTA_Stop(); //countA stop
    COUNTA_Int_Enable(); //countA INT enable
}
    
```

● **代码说明:**

- COUNT\_DeInit(); ----用于恢复默认值
- COUNTA\_IO\_Init(); ----用于配置 GPIO 为 COUNTA 功能
- COUNTA\_Init(); ----用于配置 COUNTA 周期及时钟
- COUNTA\_Config(); ----用于配置 COUNTA 中断
- COUNTA\_Start(); ----用于使能 COUNTA
- COUNTA\_Stop(); ----用于停止 COUNTA
- COUNTA\_Int\_Enable();----用于使能 COUNTA 中断

● **函数参数说明:**





COUNTA\_Config(SW\_STROBE,PENDREM\_OFF,MATCHREM\_OFF,REMSTAT\_0,ENVELOPE\_0);

### 3.2 BUZZ 输出

```

/*****/
//COUNTA Initial
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void COUNTA_CONFIG(void)
{
    COUNT_DeInit(); //clear all countA Register
    COUNTA_IO_Init(COUNTA_PB01); //set PB0.01 as counter IO
    COUNTA_Init(24000,24000,Period_H,DIV1,REPEAT_MODE,CARRIER_ON,OSP_LOW); //Data_H=Time/(1/sysclock)
    COUNTA_Config(SW_STROBE,PENDREM_OFF,MATCHREM_OFF,REMSTAT_0,ENVELOPE_0); //countA mode set
    COUNTA_Start(); //countA start
    //COUNTA_Stop(); //countA stop
    COUNTA_Int_Enable(); //countA INT enable
}
void APT32S003_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
    //-----/
    COUNTA_CONFIG(); //CountA initial
}

```

● 波形输出:

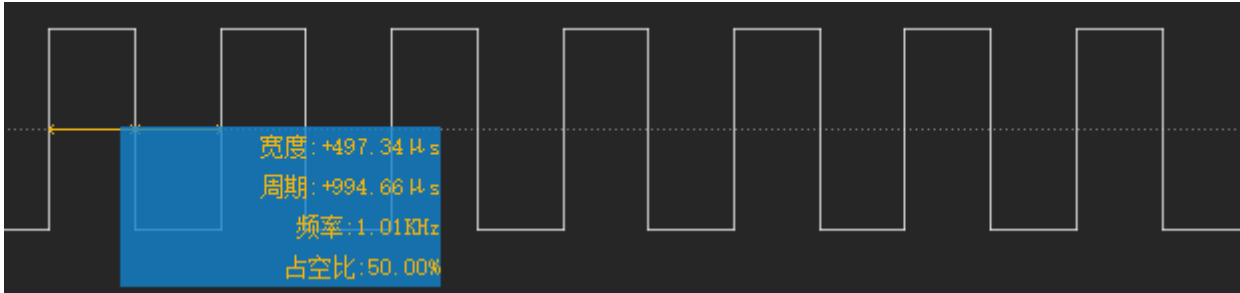


图 3.2.1 波形输出

### 3.3 定时输出

系统时钟选择内部 48Mhz，输出周期为 1ms，占空比为 500us，

```

void GPIO_CONFIG(void)
{
    GPIO_Init(GPIOA0,10,0);
    GPIO_Write_High(GPIOA0,10);
}
/*****/
//COUNTA Initial
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void COUNTA_CONFIG(void)
{
    COUNT_DeInit(); //clear all countA Register
    COUNTA_Init(24000,0,Period_H,DIV1,REPEAT_MODE,CARRIER_ON,OSP_LOW); //Data_H=Time/(1/sysclock)
    COUNTA_Config(SW_STROBE,PENDREM_OFF,MATCHREM_OFF,REMSTAT_0,ENVELOPE_0); //countA mode set
    COUNTA_Start(); //countA start
    //COUNTA_Stop(); //countA stop
    COUNTA_Int_Enable(); //countA INT enable
}
void APT32S003_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();
COUNTA_CONFIG(); //CountA initial
}
/*****/
//CONTA Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
volatile U8_T f_io_toggle;
void CNTAIntHandler(void)
{
    // ISR content ...
    if(!f_io_toggle)
    {
        f_io_toggle = 1;
        GPIO_Set_Value(GPIOA0,10,1);
    }
    else
    {
        f_io_toggle=0;
        GPIO_Set_Value(GPIOA0,10,0);
    }
}
}

```

● 波形输出：

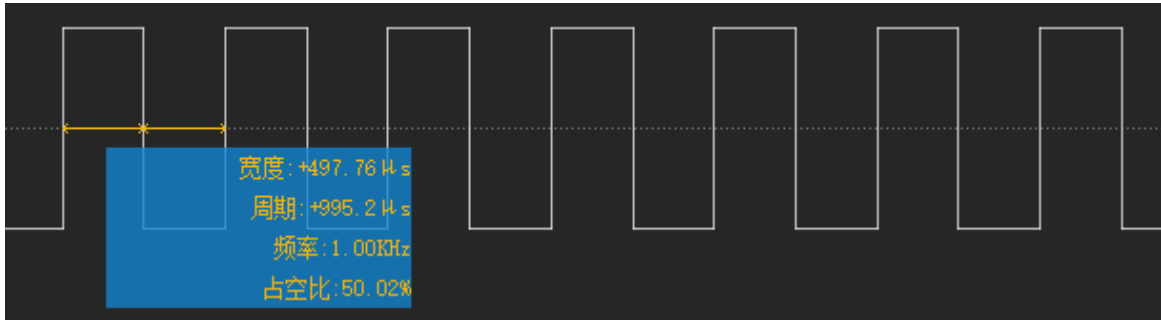


图 3.3.1 翻转 IO 输出

### 3.4 载波输出

系统时钟选择内部 48Mhz，由 BT 周期结束中断发生硬件触发，控制 PB01 输出 38KHZ 载波信号。

```

/*****
//COUNTA Initial
//EntryParameter:NONE
//ReturnValue:NONE
*****/
void COUNTA_CONFIG(void)
{
    //174*48,87*48
    COUNT_DelInit(); //clear all countA Register
    COUNTA_IO_Init(COUNTA_PB01); //set PB0.01 as counter IO
    COUNTA_Init(8*48,18*48,Period_NA,DIV1,REPEAT_MODE,CARRIER_ON,OSP_LOW); //Data_H=Time/(1/sysclock)
    COUNTA_Config(HW_STROBE_1,PENDREM_2,MATCHREM_1,REMSTAT_0,ENVELOPE_0); //countA mode set
    COUNTA_Start(); //countA start
    //COUNTA_Stop(); //countA stop
    //COUNTA_Int_Enable(); //countA INT enable
}
/*****
//BT Initial
//EntryParameter:NONE
//ReturnValue:NONE
*****/
void BT_CONFIG(void)
{
    BT_DelInit(BT0);
    //BT_IO_Init(BT0_PB02);
    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_ONCE,BT_AREARM_DIS,BT_CNTRL
D_EN);
    //BT_ControlSet_Configure(BT0,BT_START_HIGH,BT_IDLE_LOW,BT_SYNC_EN,BT_SYNCMD_DIS,BT_OSTMDX_ONCE,BT_AREARM_DIS,BT_CNTRL

```



```

D_EN);
    //BT_Trigger_Configure(BT0,BT_TRGSRV_PEND,BT_TRGOE_EN);
    BT_Period_CMP_Write(BT0,2250,560);
    BT_Start(BT0);
    BT_ConfigInterrupt_CMD(BT0,ENABLE,BT_PEND);
    BT0_INT_ENABLE();
}
void APT32S003_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/SYSCCLK/IWDT/LVD/EM_CMFAIL/EM_CMRCV/CMD_ERR OSC stable interrupt
//EntryParameter:NONE
//ReturnValue:NONE
//-----/
    SYSCON_CONFIG(); //syscon initial
    CK_CPU_EnAllNormalIrq(); //enable all I
    BT_CONFIG(); //BT initial
    COUNTA_CONFIG(); //CountA initial
}
    
```

● **波形输出:**

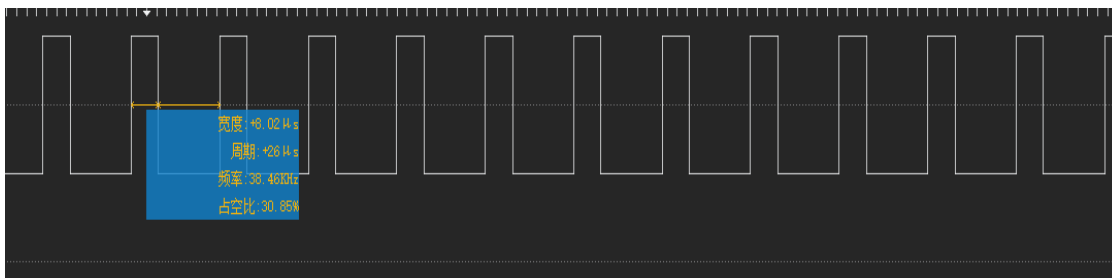


图 3.4.1 载波输出

## 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如图 3.2.1、图 3.3.1、图 3.4.1 波形