

文档版本	V1.0
发布日期	20210610

APT32S003 LPT 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 LPT 配置.....	1
3.2 LPT 定时配置.....	2
3.3 LPT PWM 配置.....	4
4. 程序下载和运行	5

1 概述

本文介绍了在APT32S003中使用LPT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 LPT 配置

基于 APT32S003 完整的库文件系统，可以对 LPT 进行配置。

- **硬件配置：**

LPT 定时器是 MCU 的低功耗外设，支持多个时钟源，支持在低功耗模式下将系统唤醒。支持一路独立的 PWM 输出。也可以基于外部 CLOCK 或独立的时钟源计数。

- **管脚描述：**

管脚名称	IO 方向	功能描述
LPT_OUT	输出	LPT 的波形输出
LPT_IN	输入	LPT 的时钟输入

图 3.1.1 管脚功能定义

- **软件配置：**

可在 apt32s003_initial.c 文件中 LPT_CONFIG 进行初始化的配置

```

/*****/
//LPT Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void LPT_CONFIG(void)
{

```

```

LPT_DeInit(); //LPT DeInit
LPT_IO_Init(LPT_OUT_PB01); //PA0.9 as LPT out
//LPT_IO_Init(LPT_IN_PA10); //PA0.10 as LPT in
LPT_Configure(LPTCLK_EN,LPT_PCLK_DIV4,LPT_IMMEDIATE,LPT_PSC_DIV0,0,LPT_OPM_CONTINUOUS); //
//LPT_Debug_Mode(ENABLE);
LPT_ControlSet_Configure(LPT_SWSYNDIS,LPT_IDLE_Z,LPT_PRDLT_DUTY_END,LPT_POL_HIGH,LPT_FLTDEB_00,LPT_PSCLD_0,LPT_CMPLD_IM
MEDIATELY);
//LPT_SyncSet_Configure(LPT_TRGEN_EN,LPT_OSTMD_ONCE,LPT_AREARM_DIS);
LPT_Trigger_Configure(LPT_SRCSEL_EN,LPT_BLKINV_DIS,LPT_CROSSMD_DIS,LPT_TRGSR0_CMP,LPT_ESYN0OE_EN,5,5,0X0F);
LPT_Period_CMP_Write(1000,500);
LPT_ConfigInterrupt_CMD(ENABLE,LPT_PEND);
LPT_Start();
LPT_INT_ENABLE();
}
    
```

● 代码说明:

LPT_DeInit(); ----用于恢复默认值

LPT_IO_Init(); ----用于配置 GPIO 口为 LPT 功能

LPT_Configure();----用于配置 LPT 时钟

LPT_ControlSet_Configure(); ----用于配置 LPT

LPT_Trigger_Configure(); ----用于配置 LPT 的事件触发

LPT_Period_CMP_Write(); ----用于配置 LPT 周期

LPT_ConfigInterrupt_CMD(); ----用于配置中断使能

LPT_Start(); ----用于启动 LPT 计数器

LPT_INT_ENABLE(); ----用于使能 LPT 中断

3.2 LPT 定时配置

选择内部主频 48MHz 作为系统时钟。 PA0.10 输出占空比为 50%,周期为 2ms 方波。

```

void GPIO_CONFIG(void)
{
    GPIO_Init(GPIOA0,10,0); //PA0.10 as output
    GPIO_Write_High(GPIOA0,10); //PA0.10 output High
}
/*****
    
```

```

//LPT Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
void LPT_CONFIG(void)
{
    LPT_DeInit(); //LPT DeInit
    LPT_Configure(LPTCLK_EN,LPT_PCLK_DIV4,LPT_IMMEDIATE,LPT_PSC_DIV0,0,LPT_OPM_CONTINUOUS); //
    LPT_Period_CMP_Write(12000,0);
    LPT_ConfigInterrupt_CMD(ENABLE,LPT_PEND);
    LPT_Start();
    LPT_INT_ENABLE();
}

/*****/
//LPT Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*****/
volatile U8_Tf_io_toggle;
void LPTIntHandler(void)
{
    // ISR content ...
    //Interrupt
    if((LPT->MISR&LPT_TRGEV0)==LPT_TRGEV0)
    {
        LPT->ICR = LPT_TRGEV0;
    }
    else if((LPT->MISR&LPT_MATCH)==LPT_MATCH)
    {
        LPT->ICR = LPT_MATCH;
    }
    else if((LPT->MISR&LPT_PEND)==LPT_PEND)
    {
        LPT->ICR = LPT_PEND;
    }

    if(!f_io_toggle)
    {
        f_io_toggle = 1;
        GPIO_Set_Value(GPIOA0,10,1);
    }
    else
    {
        f_io_toggle = 0;
    }
}

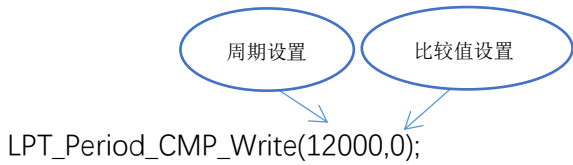
```

```

GPIO_Set_Value(GPIOA0,10,0);
}
}
}

```

● 函数参数说明:



● 输出波形:

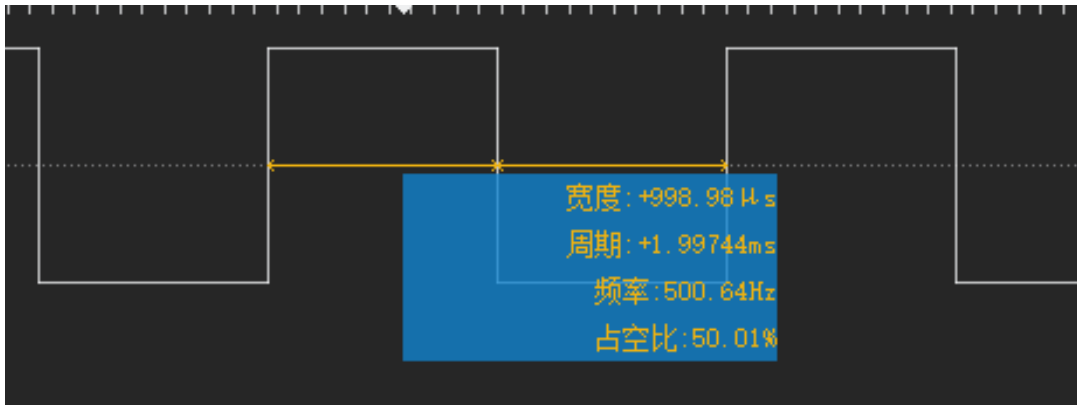


图 3.2.1 LPT 定时翻转 IO

3.3 LPT PWM 配置

选择内部主频 48MHz 作为系统时钟，PB0.1 输出占空比为 50%,周期为 1ms 方波

```

/*****/
//LPT Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****/

```

```

void LPT_CONFIG(void)
{
    LPT_DeInit(); //LPT DeInit
    LPT_IO_Init(LPT_OUT_PB01); //PA0.9 as LPT out
    LPT_Configure(LPTCLK_EN,LPT_PCLK_DIV4,LPT_IMMEDIATE,LPT_PSC_DIV0,0,LPT_OPM_CONTINUOUS); //
    LPT_ControlSet_Configure(LPT_SWSYNDIS,LPT_IDLE_Z,LPT_PRDLT_DUTY_END,LPT_POL_HIGH,LPT_FLTDEB_00,LPT_PSCLD_0,LPT_CMPLD_IM
MEDIATELY);
    LPT_Period_CMP_Write(12000,6000);
    LPT_Start();
}
    
```

● **函数参数说明：**



● **波形输出：**

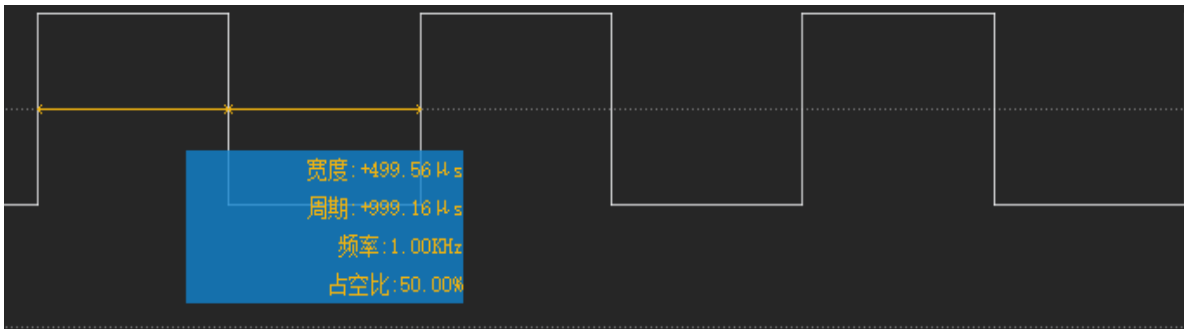


图 3.3.1 LPT 输出 PWM

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，如图 3.2.1、图 3.3.1 所示波形