

文档版本	V1.0.0
发布日期	20221026

APT32F110x 基于 CSI 库 LPT 应用指南



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 LPT 模块介绍.....	1
3.2 LPT 定时配置.....	3
3.3 LPT PWM 配置.....	4
4. 程序下载和运行	6

1 概述

本文介绍了在APT32F110x中使用LPT的应用范例。

2. 适用的硬件

该例程使用于 APT32F110x 系列学习板

3. 应用方案代码说明

3.1 LPT 模块介绍

基于 APT32F110x 完整的库文件系统，可以对 LPT 进行配置。

● 硬件配置：

低功耗定时器（Low Power Timer）作为 MCU 的低功耗外设，可以支持异步时钟计数操作，从而实现超低功耗下计数。由于可以基于外部 CLOCK 或独立的时钟源计数，LPT 可以支持在低功耗模式下将系统唤醒。当使用外部时钟计数时，LPT 可以作为外部脉冲计数使用。LPT 内部包含一个 16 位的定时/计数模块，支持 PWM 输出。

● 主要特性

- 16位递增计数器
- 4 Bit预分频控制，支持（1， 2， 4， 8， 16， 32， 64， 128， 256， 512， 1024， 2048， 4096分频）
- 支持多种计数时钟：
 - 内部时钟：ISCLK, IMCLK, EMCLK， ESCLK或PCLK
 - 外部时钟：LPT_IN（当没有内部时钟时，可以作为脉冲计数）
 - 一路独立的PWM输出

- 一个比较值寄存器
- 支持连续或单次计数模式
- 支持通过ETCB触发
- 支持脉冲和PWM输出模式

● 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

● 模块框图:

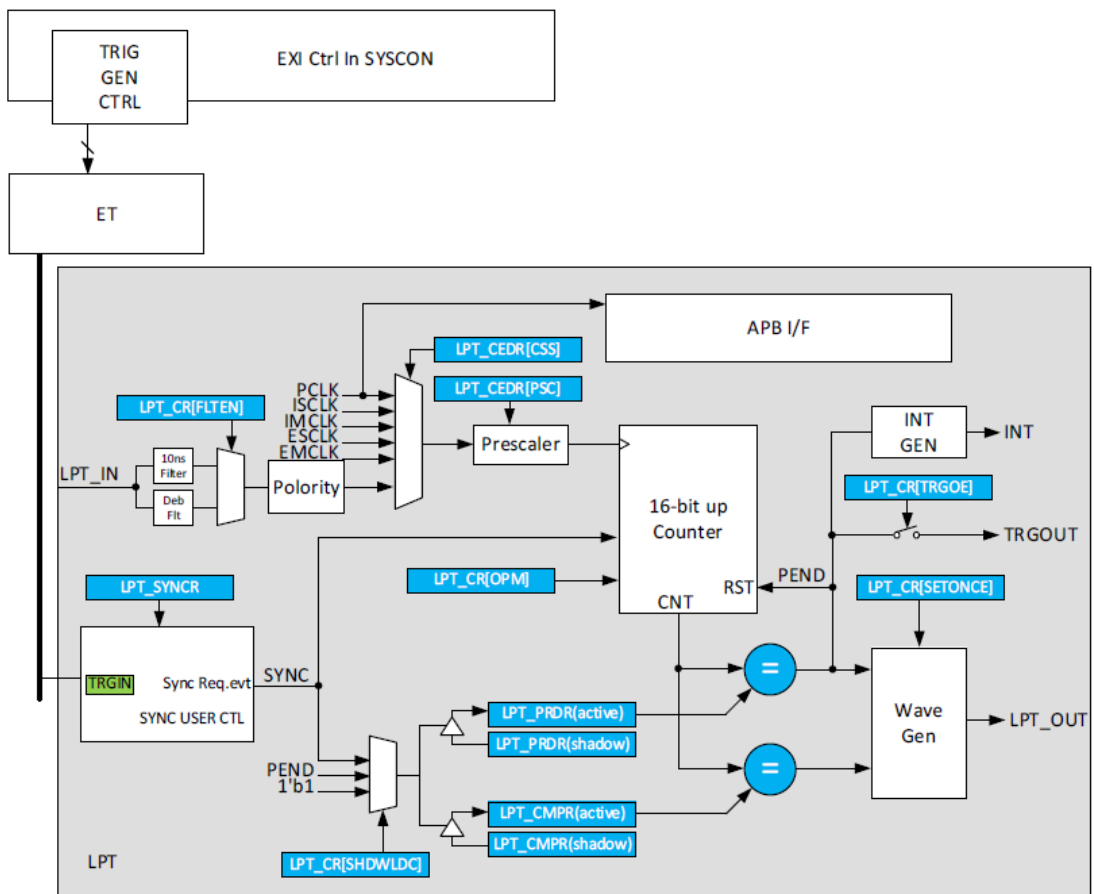


图 3.1.2 模块结构示意图

3.2 LPT 定时配置

系统时钟选择内部 48MHz，可在 user_demo.c 文件中 lpt_timer_demo()函数进行初始化的配置，利用 LPT 定时产生中断使 PA0.4 输出电平周期翻转。

```
int lpt_timer_demo(void)
{
    int iRet = 0;

    csi_lpt_timer_init(LPT,LPT_CLK_ISCLK,500);    //初始化 lpt,选用内部超低功耗时钟,定时 500ms,默认采用 PEND 中断
    csi_lpt_start(LPT);                          //启动 lpt

    return iRet;
}

void gpio_config(void)
{
    csi_pin_set_mux(PA04,PA04_OUTPUT);
    csi_pin_set_high(PA04);
    mdelay(100);
}

__attribute__((weak)) void lpt_irqhandler(csp_lpt_t *ptLptBase)
{
    volatile uint32_t wMisr = csp_lpt_get_misr(ptLptBase);

    if(wMisr & LPT_TRGEV_INT)
    {
        csp_lpt_clr_misr(ptLptBase, LPT_TRGEV_INT);
    }

    if(wMisr & LPT_MATCH_INT)
    {
        csp_lpt_clr_misr(ptLptBase, LPT_MATCH_INT);
    }

    if(wMisr & LPT_PEND_INT)
    {
        csp_lpt_clr_misr(ptLptBase, LPT_PEND_INT);
        csi_pin_toggle(PA04);
    }
}
```

- 代码说明：

csi_lpt_timer_init(): ----- 初始化 LPT 定时器功能。

csi_lpt_start(): ----- 启动 LPT 功能。

● 函数参数说明:

csi_lpt_timer_init(csp_lpt_t *ptLptBase,csi_lpt_clksrc_e eClk, uint32_t wTimeOut);

ptLptBase: LPT寄存器结构体指针，指向LPT的基地址。

eClk: 计数器时钟源选择，有PCLK/4、ISCLK、IMCLK/4、EMCLK、ESCLK、LPT_IN的上升沿、LPT_IN的下降沿多种选择。

wTimeOut: 定时时间，单位 ms。

csi_lpt_start(csp_lpt_t *ptLptBase);

ptLptBase: LPT 寄存器结构体指针，指向 LPT 的基地址。

● 输出波形:

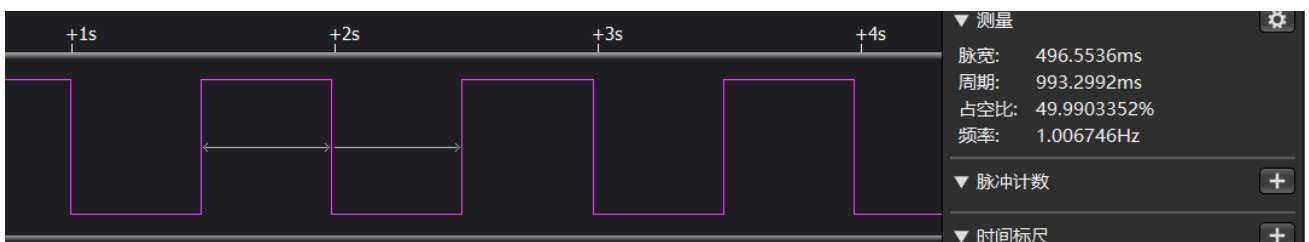


图 3.2.1 LPT 定时翻转 IO

3.3 LPT PWM 配置

选择内部主频 48MHz 作为系统时钟，可在 user_demo.c 文件中 lpt_pwm_demo()函数进行配置，PA0.1 输出占空比为 80%,周期为 1ms 的波形。

```

int lpt_pwm_demo(void)
{
    int iRet = 0;

    csi_lpt_pwm_config_t tLptPwmCfg;
    csi_pin_set_mux(PA01, PA01_LPT_INOUT); //将 PA01 设为 LPT_OUT

    tLptPwmCfg.byClksrc = LPT_CLK_PCLK_DIV4; //PWM 时钟选择
    tLptPwmCfg.byStartpol = LPT_PWM_START_HIGH; //初始低电平
    tLptPwmCfg.byIdlepol = LPT_PWM_IDLE_LOW; //停止时 highZ
    tLptPwmCfg.byCycle = 80; //PWM 输出占空比(0~100)
}
    
```

```

tLptPwmCfg.wFreq = 1000;                //PWM 输出频率
tLptPwmCfg.byInt  = LPT_INTSRC_NONE;    //PWM 配置无中断
if(csi_lpt_pwm_init(LPT, &tLptPwmCfg) == CSI_OK){ //初始化 lpt
    csi_lpt_start(LPT);                //启动 lpt
}

return iRet;
}

```

- 代码说明:

csi_pin_set_mux(): ----- 设置 PIN 脚功能。

csi_lpt_pwm_init(): ----- PWM 输出初始化。

csi_lpt_start(): ----- 启动 LPT 功能。

- 函数参数说明:

csi_pin_set_mux(pin_name_e ePinName, pin_func_e ePinFunc);

ePinName: PIN 脚名字。

ePinFunc: PIN 脚功能。

csi_lpt_pwm_init(csp_lpt_t *ptLptBase, csi_lpt_pwm_config_t *ptLptPara);

ptLptBase: LPT 寄存器结构体指针，指向 LPT 的基地址。

ptLptPara: LPT PWM 输出参数设置结构体指针。

ptLptPara->byClksrc: PWM 时钟选择;

ptLptPara->byStartpol: 波形输出极性控制;

ptLptPara->byIdlepol: 波形停止时，输出端口的状态;

ptLptPara->byCycle: PWM 输出占空比;

ptLptPara->wFreq: PWM 输出频率;

ptLptPara->byInt: 中断配置。

- 波形输出:

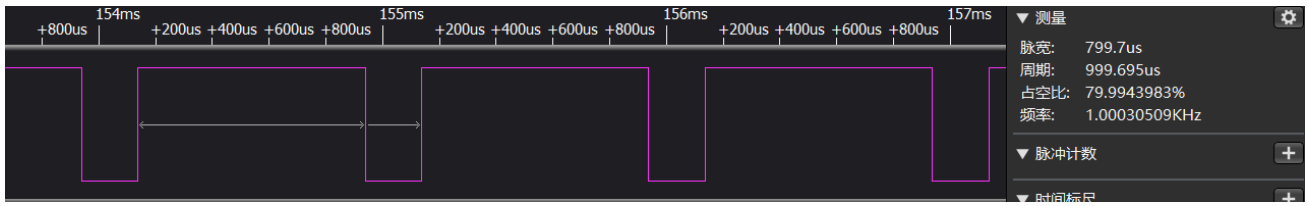


图 3.3.1 LPT 输出 PWM

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