

文档版本	V1.0.0
发布日期	20221026

# APT32F110x 基于 CSI 库 LED 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 LED 配置 .....	1
3.2 LED 闪烁配置 .....	4
4. 程序下载和运行 .....	4

## 1 概述

本文介绍了在APT32F110x中LED控制器。

## 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

基于 APT32F110x CSI 的库文件系统，进行配置 LED

### 3.1 LED 配置

#### ● 硬件配置

内嵌一个 10x8 点的 LED 自动扫描控制器模块,最大支持 10 个 COM 的扫描信号，通过不同时间槽分割来点亮 10 个 8 段数码管。COM 扫描的通道可以自定义，每个 COM 管脚，均支持大电流驱动模式，可以在不外加驱动电路的前提下，直接驱动数码管（外部限流电阻仍然需要）。

LED 模块有 2 个中断。一个为 ICEND 中断，此中断将在每个 COM 扫描结束时触发。一个为 IPEND 中断，此中断将在所有的 COM 扫描结束后触发。

**注：LED 驱动控制模块资源只有 APT32F1101 及 APT321103 产品有，其余系列无此模块。**

#### ● 软件配置：

驱动 8 个 8 段数码管，调试使用时需要注意，PA1.7 PA1.8 配置改为 LED COM 后会无法进入仿真。具体配置如下

```
void led_config(void)
{
    csi_led_config_t ptLedCfg;           //定义结构体
```

```

//配置 LED_IO

csi_pin_set_mux(PA013, PA013_LED_S7);//seg
csi_pin_set_mux(PA012, PA012_LED_S6);
csi_pin_set_mux(PA011, PA011_LED_S5);
csi_pin_set_mux(PA010, PA010_LED_S4);
csi_pin_set_mux(PA09, PA09_LED_S3);
csi_pin_set_mux(PA08, PA08_LED_S2);
csi_pin_set_mux(PA07, PA07_LED_S1);
csi_pin_set_mux(PA06, PA06_LED_S0);

/*
//

csi_pin_set_mux(PA12, PA12_LED_S7);
csi_pin_set_mux(PA11, PA11_LED_S6);
csi_pin_set_mux(PA10, PA10_LED_S5);
csi_pin_set_mux(PB04, PB04_LED_S4);
csi_pin_set_mux(PC01, PC01_LED_S3);
csi_pin_set_mux(PB03, PB03_LED_S2);
csi_pin_set_mux(PA015, PA015_LED_S1);
csi_pin_set_mux(PA014, PA014_LED_S0);

//

csi_pin_set_mux(PA011, PA011_LED_S0);
csi_pin_set_mux(PA012, PA012_LED_S1);
csi_pin_set_mux(PB03, PB03_LED_S2);
csi_pin_set_mux(PA013, PA013_LED_S3);
csi_pin_set_mux(PB04, PB04_LED_S4);
csi_pin_set_mux(PA10, PA10_LED_S5);
csi_pin_set_mux(PA11, PA11_LED_S6);
csi_pin_set_mux(PA12, PA12_LED_S7);

*/

//

csi_pin_set_mux(PA16, PA16_LED_COM0);//com □
csi_pin_set_mux(PB05, PB05_LED_COM1);
//csi_pin_set_mux(PA17, PA17_LED_COM2); //SWD-DATA
csi_pin_set_mux(PB06, PB06_LED_COM3);
csi_pin_set_mux(PB07, PB07_LED_COM4);
// csi_pin_set_mux(PA18, PA18_LED_COM5); //SWD-CLK
csi_pin_set_mux(PB08, PB08_LED_COM6);
csi_pin_set_mux(PA111, PA111_LED_COM7);
csi_pin_set_mux(PA112, PA112_LED_COM8);
csi_pin_set_mux(PA113, PA113_LED_COM9);

//配置参数
ptLedCfg.byClk = LED_PCLK_DIV8; //基于 PCLK 的分频 48M/8 = 125US
ptLedCfg.hwComMask = 0x3DB; //
    
```

```
ptLedCfg.byBrn = LED_50;

ptLedCfg.byOnTime = 0x7f;           //DCOMCNT
ptLedCfg.byBreakTime = 50;         //NOVCNT
ptLedCfg.byInt = LED_INTSRC_NONE;
csi_led_init(LED, &ptLedCfg);      //LED 配置初始化
//
csi_led_lighton(LED);              //LED 扫描启动
//
csi_led_write_data(LED, 0, LedData[0]);
//mdelay(1000);
csi_led_write_data(LED, 1, LedData[1]);
csi_led_write_data(LED, 3, LedData[3]);
csi_led_write_data(LED, 4, LedData[4]);
csi_led_write_data(LED, 6, LedData[6]);
csi_led_write_data(LED, 7, LedData[7]);
csi_led_write_data(LED, 8, LedData[8]);
csi_led_write_data(LED, 9, LedData[9]);
}
```

- 代码说明:

`csi_pin_set_mux ();`-----用于配置 GPIO 功能

`csi_led_init ();`-----用于配置 LED 模块

`csi_led_lighton ();`-----用于使能 LED 模块

`csi_led_write_data ();` -----用于配置 LED 显示数据

- 函数参数说明:

`csi_led_init ()`

对结构体 `ptLedCfg` 进行赋值操作，从而配置 LED 模块。

`ptLedCfg.byClk = LED_PCLK_DIV8;` //基于 PCLK 的分频 48M/8 = 125US

`ptLedCfg.hwComMask = 0x3DB;` //配置对应 COM 扫描使能

`ptLedCfg.byBrn = LED_50;` //亮度

`ptLedCfg.byOnTime = 0x7f;` // COM 显示周期

```
ptLedCfg.byBreakTime = 50; //两个 COM 间隔长度
```

```
ptLedCfg.byInt = LED_INTSRC_NONE; //使能对应中断
```



```
csi_led_write_data (LED, 0, LedData[0]);
```

● 显示图:

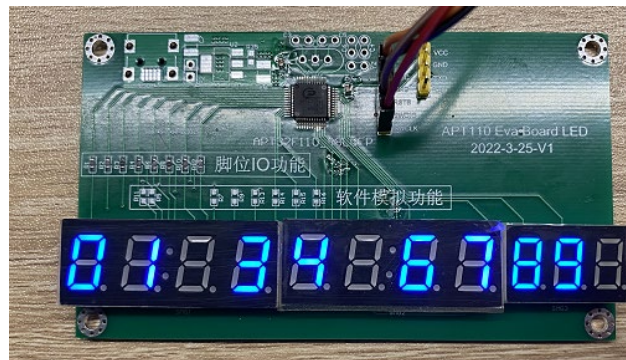


图 3.1.1 显示图

### 3.2 LED 闪烁配置

```
static inline void csp_led_com_on(csp_led_t *ptLedBase, uint16_t hwClrMsk)
{
    ptLedBase -> BLKDR = hwClrMsk;
}
static inline void csp_led_com_off(csp_led_t *ptLedBase, uint16_t hwDisMsk)
{
    ptLedBase -> BLKER = hwDisMsk;
}
```

● 代码说明:

csp\_led\_com\_on(); ----用于使能扫描输出

csp\_led\_com\_off(); ----用于禁用扫描输出

## 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 LED Demo 板查看显示数据。