版本	*V1.0*
日期	*2022.03.01*

APT32F171x 系列

CSI API 手册

相关文档：

APT32F171x系列芯片使用手册

APT32F171芯片数据手册

QuickStart_APT32F171x系列_CSI

Revision History

版本	日期	描述	作者
V1.0	2022.03.01	新建	APT AE Group

1 概述

APT32F171 驱动软件遵循 APT CSI（APT chip standard interface）接口规范，帮助客户快速形成产品方案。本文档旨在对该接口规范下的 API 函数进行说明。

几个注意点：

1. APT32F171 系列内包含多款 MCU，如果使用的 MCU 不包含某一外设，这部分外设相关的 api 函数无效。
2. 所有外设的 api 函数中都不包含管脚功能配置，使用前需要调用 csi_pin_set_mux()函数进行配置。
3. 有关工程结构、系统初始化、调试打印、中断相关的使用请参考 QuickStart_APT32F171x系列_CSI。

2 时钟

2.1 API列表

Table 2-1 时钟CSI接口函数

API	说明	函数位置
csi_sysclk_config	配置系统时钟SCLK。	sys_clk.c
csi_clo_config	设置管脚的CLO输出。	
csi_get_sclk_freq	获得当前SCLK频率值。	
csi_get_pclk_freq	获得当前PCLK频率值。	
csi_clk_enable	使能模块时钟（SYSCON端）	clk.c
csi_clk_disable	禁止模块时钟（SYSCON端）	
csi_emosc_enable	打开EMOSC。	
csi_emosc_disable	关闭EMOSC。	
csi_imosc_enable	打开IMOSC。	
csi_imosc_disable	关闭IMOSC。	
csi_hfosc_enable	打开HFOSC。	
csi_hfosc_disable	关闭HFOSC。	
csi_isosc_enable	打开ISOSC。	
csi_isosc_disable	关闭ISOSC。	

2.2 API详细说明

2.2.1 csi_sysclk_config

csi_error_t csi_sysclk_config(void)

2.2.1.1 功能描述

用于配置系统时钟 SCLK。调用此函数前，

1. 需要设置时钟参数 tClkConfig （位于board_config.c）。

```
/// system clock configuration parameters to define source, source freq(if selectable), sdiv and pdiv
csi_clk_config_t tClkConfig =
    //{SRC_HFOSC, HFOSC_48M_VALUE, SCLK_DIV1, PCLK_DIV1, 5556000, 5556000};
    //{SRC_HFOSC, HFOSC_48M_VALUE, SCLK_DIV1, PCLK_DIV1,5556000, 5556000};
    //{SRC_IMOSC, IMOSC_5M_VALUE, SCLK_DIV1, PCLK_DIV1,5556000, 5556000};
    //{SRC_IMOSC, IMOSC_4M_VALUE, SCLK_DIV1, PCLK_DIV1,5556000, 5556000};
    //{SRC_ISOSC, ISOSC_VALUE, SCLK_DIV1, PCLK_DIV1,5556000, 5556000};
    {SRC_EMOSC, 24000000, SCLK_DIV1, PCLK_DIV1, 5556000, 5556000};
```

Figure 2-1 时钟参数结构体

2. 如果时钟源为EMOSC，需要预先设置XIN，XOUT管脚AF功能。
3. 在board_config.h中自定义你的配置（频率和晶振管脚，功能的宏定义，在配置时会检查是否位晶振管脚）
4.

```
csi_error_t csi_emosc_enable(uint32_t wFreq)
{

    if ((csi_pin_get_mux(XIN_PIN) != XIN_PIN_FUNC) || (csi_pin_get_mux(XOUT_PIN) != XOUT_PIN_FUNC))
        return CSI_ERROR;
```

```
#define EMOSC_VALUE     24000000U
#define XIN_PIN         PC02
#define XOUT_PIN        PA011
#define XIN_PIN_FUNC    PC02_OSC_XI
//#define XIN_PIN_FUNC  PA010_OSC_XI
#define XOUT_PIN_FUNC   PA011_OSC_XO
```

函数执行完毕后，会更新tClkConfig中的wSclk和wPclk。

2.2.1.2 参数/返回值说明

1. 参数：无。
2. 返回值

   如果系统时钟选择IMOSC或HFOSC，但tClkConfig的第二次参数不是可选值时，会返回CSI_ERROR。否则返回CSI_OK。

3. 参数说明表

变量	说明	变量位置

tClkConfig	全局变量，结构体。
	`typedef struct {`
	`  cclk_src_e  eClkSrc;  //clock source`
	`  uint32_t  wFreq;  //clock frequency`
	`  hclk_div_e  eSdiv;  //SDIV`
	`  pclk_div_e  ePdiv;  //PDIV`
	`  uint32_t  wSclk;  //SCLK`
	`  uint32_t  wPclk;`
	`}csi_clk_config_t;`
		在board_config.c中定义
		在board_config.h中extern。如需使用，#include board_config.h

2.2.2 csi_clo_config

csi_error_t csi_clo_config(clo_src_e eCloSrc, clo_div_e eCloDiv)

2.2.2.1 功能描述

用于设置管脚的 CLO 输出。该函数常用于调试阶段，查看当前各种时钟信号的实际频率。

2.2.2.2 参数/返回值说明

1. 参数

   eCloSrc：CLO管脚上输出的时钟对象，枚举定义详见clo_src_e。

   eCloDiv：CLO管脚上输出的时钟对象分频值，枚举定义详见clo_div_e。

2. 返回值

   CSI_OK：初始化成功。

   CSI_ERROR：初始化失败。

3. 参数说明表

参数/返回值	说明	枚举变量位置
eCloSrc	clo_src_e枚举值。 typedef enum{ CLO_ISCLK = 0, CLO_IMCLK, CLO_EMCLK = 3, CLO_HFCLK, CLO_PCLK = 7, CLO_HCLK, CLO_IWDTCLK, CLO_SYSCLK = 0xd }clo_src_e;	csp_syscon,h
eCloDiv	clo_div_e枚举值。 typedef enum{ CLO_DIV1 = 1, CLO_DIV2, CLO_DIV4, CLO_DIV8, CLO_DIV16 }clo_div_e;	
csi_error_t	csi_error_t 中定义值	common.h

2.2.3 csi_get_sclk_freq

uint32_t csi_get_sclk_freq(void)

2.2.3.1 功能描述

获得当前SCLK频率值。函数执行完毕后，会更新tClkConfig中的wSclk。

2.2.3.2 参数/返回值说明

1. 参数：无。
2. 返回值：返回SCLK的频率值（单位：Hz）
3. 参数说明表

返回值类型	说明
uint32_t	返回系统时钟(SCLK)频率

2.2.4 csi_get_pclk_freq

uint32_t csi_get_pclk_freq(void)

2.2.4.1 功能描述

获得当前PCLK频率值。函数执行完毕后，会更新tClkConfig中的wPclk。我们的csi驱动中，如UART在配置时只需要传入波特率的数值就可以，是因为内部会调用这个函数，自动计算出在当前PCLK的频率下要取得用户设置的波特率时，需要对相关寄存器写入的值。很多通信外设和定时外设也都是这样。

2.2.4.2 参数/返回值说明

1. 参数：无。
2. 返回值：返回PCLK的频率值（单位：Hz）
3. 参数说明表

返回值类型	说明
uint32_t	返回外设时钟(PCLK)频率

2.2.5 csi_clk_enable

*void csi_clk_enable(uint32_t *pIpBase)*

2.2.5.1 功能描述

处于功耗的考虑，大部分外设的时钟都有开关。这些开关都集中位于 SYSCON_PCER0/1。所以要开启某一个外设，需要先打开这个开关。这个函数通常已经集成在外设的初始化函数中，不需要用户调用。

2.2.5.2 参数/返回值说明

1. 参数

   pIpBase：指向外设基地址的结构体指针，使用时应强制转换为uint32_t的指针类型。

2. 返回值：无

3. 参数说明表

参数	说明	位置

pIpBase	VIC_Type *VIC0 = (VIC_Type*)(VIC_ADDR_BASE);
	csp_coret_t *CORET0 = (csp_coret_t *)(CORET_ADDR_BASE);
	csp_crc_t *CRC = (csp_crc_t *)(AHB_CRC_BASE);
	csp_ifc_t *IFC = (csp_ifc_t *)(APB_IFC_BASE);
	csp_syscon_t *SYSCON = (csp_syscon_t *)(APB_SYS_BASE);
	csp_etb_t *ETCB = (csp_etb_t *)(APB_ETCB_BASE);
	csp_adc_t *ADC0 = (csp_adc_t *)(APB_ADC0_BASE);
	csp_gpio_t *GPIOA0 = (csp_gpio_t *)(APB_GPIOA0_BASE);
	csp_gpio_t *GPIOA1 = (csp_gpio_t *)(APB_GPIOA1_BASE);
	csp_gpio_t *GPIOB0 = (csp_gpio_t *)(APB_GPIOB0_BASE);
	csp_gpio_t *GPIOC0 = (csp_gpio_t *)(APB_GPIOC0_BASE);
	csp_igrp_t *GPIOGRP = (csp_igrp_t *)(APB_IGRP_BASE);
	csp_bt_t *BT0 = (csp_bt_t *)(APB_BT0_BASE);
	csp_bt_t *BT1 = (csp_bt_t *)(APB_BT1_BASE);
	csp_bt_t *BT2 = (csp_bt_t *)(APB_BT2_BASE);
	csp_bt_t *BT3 = (csp_bt_t *)(APB_BT3_BASE);
	csp_gpta_t *GPTA0 = (csp_gpta_t *)(APB_GPTA0_BASE);
	csp_ept_t *EPT0 = (csp_ept_t *)(APB_EPT0_BASE);
	csp_wwdt_t *WWDT = (csp_wwdt_t *)(APB_WWDT_BASE);
	csp_uart_t *UART0 = (csp_uart_t *)(APB_UART0_BASE);
	csp_usart_t *USART0 = (csp_usart_t *)(APB_USART0_BASE);
	csp_iwdt_t *IWDT = (csp_iwdt_t *)(APB_SYS_BASE);
	csp_cmp_t *CMP = (csp_cmp_t *)(APB_CMP_BASE);
	csp_opa_t *OPA0 = (csp_opa_t *)(APB_OPA0_BASE);
	csp_opa_t *OPA1 = (csp_opa_t *)(APB_OPA1_BASE);
		devices.c中定义

2.2.6 csi_clk_disable

*void csi_clk_disable(uint32_t *pIpBase)*

2.2.6.1 功能描述

关闭外设时钟，功能和 csi_clk_enable 函数相反。

2.2.6.2 参数/返回值说明

同 csi_clk_enable 函数。

2.2.7 csi_emosc_enable

csi_error_t csi_emosc_enable(uint32_t wFreq)

2.2.7.1 功能描述

使能外部主晶振。

2.2.7.2 参数/返回值说明

1. 参数

   wFreq：外部主振荡器频率值。

2. 返回值：

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 参数/返回值说明表

参数/返回值	说明	枚举变量位置
wFreq	uint32_t 类型数值	
csi_error_t	csi_error_t 中定义值	common.h

2.2.8 csi_emosc_disable

csi_error_t csi_emosc_disable(void)

2.2.8.1 功能描述

关闭外部主晶振。

2.2.8.2 参数/返回值说明

1. 参数：无
2. 返回值

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 返回值说明表

返回值	说明	枚举变量位置
csi_error_t	csi_error_t 中定义值	common.h

2.2.9 csi_imosc_enable

csi_error_t csi_imosc_enable(uint8_t byFre)

2.2.9.1 功能描述

使能内部低速振荡器。

2.2.9.2 参数/返回值说明

1. 参数

   byFre：振荡器可选择 4 中不同频率值，0~3 对应频率为：5.5M/4.2M/2.09M/131K

2. 返回值：

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 参数/返回值说明表

返回值	说明	枚举变量位置
byFre	uint8_t 类型数值	
csi_error_t	csi_error_t 中定义值	common.h

2.2.10 csi_imosc_disable

csi_error_t csi_imosc_disable(void)

2.2.10.1 功能描述

关闭内部低速振荡器。

2.2.10.2 参数/返回值说明

1. 参数：无
2. 返回值

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 返回值说明表

返回值	说明	枚举变量位置
csi_error_t	csi_error_t 中定义值	common.h

2.2.11 csi_hfosc_enable

csi_error_t csi_hfosc_enable(uint8_t byFre)

2.2.11.1 功能描述

使能内部高速振荡器。

2.2.11.2 参数/返回值说明

1. 参数

   byFre：振荡器可选择 4 中不同频率值，0~3 对应频率为：48M/24M/12M/6M

2. 返回值：

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 参数/返回值说明表

返回值	说明	枚举变量位置
byFre	uint8_t 类型数值	
csi_error_t	csi_error_t 中定义值	common.h

2.2.12 csi_hfosc_disable

csi_error_t csi_hfosc_disable(void)

2.2.12.1 功能描述

关闭内部高速振荡器。

2.2.12.2 参数/返回值说明

1. 参数：无
2. 返回值

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 返回值说明表

返回值	说明	枚举变量位置
csi_error_t	csi_error_t 中定义值	common.h

2.2.13 csi_isosc_enable

csi_error_t csi_isosc_enable(void)

2.2.13.1 功能描述

使能内部超低速振荡器，频率值为 27K。

2.2.13.2 参数/返回值说明

1. 参数：无
2. 返回值

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 返回值说明表

返回值	说明	枚举变量位置
csi_error_t	csi_error_t 中定义值	common.h

2.2.14 csi_isosc_disable

csi_error_t csi_isosc_disable(void)

2.2.14.1 功能描述

关闭内部超低速振荡器。

2.2.14.2 参数/返回值说明

1. 参数：无
2. 返回值

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 返回值说明表

返回值	说明	枚举变量位置
csi_error_t	csi_error_t 中定义值	common.h

3 CORET

3.1 概述

系统定时器（CORET），24位循环递减计数器，用于系统计时，提供延时函数等。系统启动时默认打开，定时器默认配置为100Hz，即定时时间是10ms； APT CSI接口提供了TICK的相关配置和操作。

3.2 API列表

Table 3-1 CORET CSI接口函数

API	说明	函数位置
csi_tick_init	初始化tick	tickc
csi_tick_get	获取系统tick计数值	
csi_tick_get_ms	获取系统tick计时时间，单位：ms	
csi_tick_get_us	获取系统tick计时时间，单位：us	
csi_tick_attach_callback	注册tick中断处理回调函数	
mdelay	毫秒延时函数	
udelay	微秒延时函数	

3.3 API详细说明

3.3.1 csi_tick_init

csi_error_t csi_tick_init(void)

3.3.1.1 功能描述

初始化系统tick

3.3.1.2 参数/返回值说明

1. 参数

无参数，要修改tick定时时间，去修改宏定义CONFIG_SYSTICK_HZ的值，在tick.h中，默认100，单位Hz。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 返回值说明

返回值	说明	概述及其枚举定义位置
csi_error_t 中	csi_error_t 中定义值	在common.h中定义

3.3.2 csi_tick_get

uint32_t csi_tick_get(void)

3.3.2.1 功能描述

获取系统tick计数值

3.3.2.2 参数/返回值说明

1. 参数

   无参数。

2. 返回值

   tick计数值，即每到达tick的定时值，计数值加1。如，tick定时值为10ms，tick计数值每10ms加1。

3. 返回值说明

返回值	说明	概述
reruen value	uint32_t 类型数值	系统tick计数值

3.3.3 csi_tick_get_ms

uint32_t csi_tick_get_ms(void)

3.3.3.1 功能描述

获取系统tick运行时间

3.3.3.2 参数/返回值说明

1. 参数

   无参数。

2. 返回值

   系统tick运行时间，即系统tick总的运行时间，单位：ms

3. 返回值说明

返回值	说明	概述
reruen value	uint32_t 类型数值，系统tick运行时间	tick运行时间，单位：ms

3.3.4 csi_tick_get_us

uint64_t csi_tick_get_us(void)

3.3.4.1 功能描述

获取系统tick运行时间

3.3.4.2 参数/返回值说明

1. 参数

   无参数

2. 返回值

   系统tick运行时间，即系统tick总的运行时间，单位：us

3. 返回值说明

返回值	说明	概述
reruen value	uint64_t 类型数值，系统tick运行时间	tick运行时间，单位：us

3.3.5 csi_tick_attach_callback

*void csi_tick_attach_callback(void *callback)*

3.3.5.1 功能描述

注册tick中断处理回调函数，用户可注册一个函数，被注册的函数在 tick 中断中调用。

3.3.5.2 参数返回值说明

1. 参数

   callback：回调函数指针，指向要注册的函数，回调函数详见结构体定义 csi_tick_t。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述

callback	`typedef struct {`
	`  void (*callback)(void *pArg);`
	`} csi_tick_t;`
		回调函数参数为tick全局计数值，每个tick中断计数一次，给用户使用。
		tick.h中定义

3.3.6 mdelay

void mdelay(uint32_t ms)

3.3.6.1 功能描述

毫秒延时函数

3.3.6.2 参数/返回值说明

4. 参数

ms：延时时间，单位：ms

5. 返回值

无返回值。

6. 参数说明

参数	说明	概述
ms	uint32_t 类型数值，延时值	延时时间，单位：ms

3.3.7 udelay

void udelay(uint32_t us)

3.3.7.1 功能描述

微秒延时函数

3.3.7.2 参数/返回值说明

1. 参数

us：延时时间，单位：us；us应大于10us。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述
us	uint32_t 类型数值，延时值	延时时间，单位：us

4 系统可靠性

4.1 LVD

LVD模块包括LVD和LVR，分别是低电压检测和低电压复位的意思，这里的电压指的是MCU供电电压。LVD通常用于电平下降或上升时产生中断事件。LVR一旦使能，则会在电压下降到某个电平值后，使芯片进入复位状态，直到电压重新回到设定值以上。无论是LVR还是LVD，存在一个迟滞电压。具体参数请参考芯片的数据手册。

LVD和LVR可以设置为不同的电平值。LVD和LVR可以同时工作。

4.1.1 API列表

Table 4-1 系统可靠性 CSI接口函数

API	说明	函数位置
csi_lvd_int_enable	配置低电压检测的电平值，并使能中断	reliability.c
csi_lvd_disable	关闭低电压检测功能	
csi_get_lvdlevel	读取当前低电压检测的电平值	
csi_lvd_flag	LVD当前状态查询	
csi_lvr_enable	配置低电压复位的电平值	
csi_lvr_disable	关闭低电压复位功能	
csi_get_lvrlevel	读取当前低电压复位的电平值	

4.1.2 API详细说明

4.1.2.1 csi_lvd_int_enable

void csi_lvd_int_enable(csi_lvd_pol_e ePol, csi_lvd_level_e eLvl)

4.1.2.1.1 功能描述

该函数实现了3个功能：配置低电压检测的电平值、设置中断事件发生极性（下降到预设值或上升到预设值），并使能中断。

4.1.2.1.2 参数/返回值说明

参数	说明	枚举量位置

ePol	可选值为csi_lvd_pol_e枚举值中的一个。
	typedef enum {
	LVD_INTF = 1, //电压下降到预设值时产生事件
	LVD_INTR, //电压上升到预设值时产生事件
	LVD_INTFR //电压下降或上升到预设值时产生事件
	}clvd_pol_e;
		reliability.h

eLvl	可选值为csi_lvd_level_e枚举值中的一个。 typedef enum{ LVD_24 = 0, LVD_21, LVD_27, LVD_30, LVD_33, LVD_36, LVD_39, LVD_10 //PB00 管脚的 lvdin 电压与内部参考比较（run:1v deepsleep:内部 0.9v） }clvd_level_e;	

4.1.2.2 csi_lvd_disable

void csi_lvd_disable(void)

4.1.2.2.1 功能描述

关闭芯片的低电压检测功能。注意，因为LVD和LVR实际是一个模块，所以关闭LVD的同时LVR功能也会被关闭。

4.1.2.2.2 参数/返回值说明

无。

4.1.2.3 csi_lvr_enable

void csi_lvr_enable(csi_lvr_level_e eLvl)

4.1.2.3.1 功能描述

该函数用于配置触发复位的低电压值。

4.1.2.3.2 参数/返回值说明

1. 参数：

   eLvl：低电压复位的门限电平值。

2. 返回值：无。

3. 参数说明表

参数	说明	位置
eLvl	可选值为csi_lvr_level_e枚举值中的一个。 typedef enum { LVR_19 = 0, LVR_22, LVR_25, LVR_28, LVR_31, LVR_34, LVR_37, LVR_40 }clvr_level_e;	reliability.h

4.1.2.4 csi_get_lvdlevel

uint32_t csi_get_lvdlevel(void)

4.1.2.4.1 功能描述

该函数用于获取lvd的电压值。

4.1.2.4.2 参数/返回值说明

1. 参数：无
2. 返回值：lvd的电平值乘以十倍。

4.1.2.5 ccsi_lvd_flag

uint32_t csi_lvd_flag (void)

4.1.2.5.1 功能描述

该函数用于获取lvd的当前状态。

4.1.2.5.2 参数/返回值说明

3. 参数：无
4. 返回值：lvd的当前状态。

   0h：VDD 的当前电压高于 INTLVL 设置的检测阈值。非0：VDD 的当前电压低于 INTLVL 设置的检测阈值。

4.1.2.6 csi_get_lvrlevel

uint32_t csi_get_lvrlevel(void)

4.1.2.6.1 功能描述

该函数用于获取lvr的电压值。

4.1.2.6.2 参数/返回值说明

1. 参数：无
2. 返回值：lvr的电平值乘以十倍。

4.2 EMOSC 时钟监测

EMOSC时钟监测模块一旦使能，会持续监控外部EMOSC的状态。如果检测到振荡异常，可以有两种动作：复位，自动切换系统时钟到IMOSC，同时可以配置产生中断。

4.2.1 API列表

API	说明	函数位置
csi_emcm_2_imosc_int	使能EMCM功能，当监测到EMOSC异常时，将系统时钟切换到IMOSC，且触发中断。	reliability.c
csi_emcm_rst	使能EMCM功能，当监测到EMOSC异常时，系统复位。	
csi_emcm_disable	关闭EMCM功能。	

4.2.2 API详细说明

略。见API列表中的说明。

4.3 内存检验

内存校验模块一旦使能，可以对SRAM和Flash的内容进行实时的检测。可以设置允许的检验错误次数上限。如果SRAM内容错误次数超过上限，可以有两种动作：复位和中断事件。如果Flash内容错误次数超过上限，芯片会直接复位。

4.3.1 API列表

API	说明	函数位置
csi_sramcheck_set_times	设置SRAM校验允许的错误次数上限。复位值为0xffff。	
csi_sramcheck_rst	设置SRAM校验错误次数超过设置的上限时，复位芯片。	
csi_sramcheck_int	设置SRAM校验错误次数超过设置的上限时，产生中断。	
csi_sramcheck_disable	禁止SRAM校验功能。	
csi_flashcheck_set_times	设置Flash校验允许的错误次数上限。复位值为0xffffff。	
csi_flashcheck_rst	设置Flash校验错误次数超过设置的上限时，复位芯片。	
csi_flashcheck_disable	禁止Flash校验功能。	reliability.c

4.3.2 API详细说明

略。见API列表中的说明。

4.4 复位源

芯片在每次复位的时候都会对复位的原因进行记录。

4.4.1 API列表

API	说明	函数位置
csi_get_rst_reason	获取上次复位的原因。	reliability.c
csi_clr_rst_reason	清除复位信息	

4.4.2 API详细说明

4.4.2.1 csi_get_rst_reason

uint16_t csi_get_rst_reason(void)

4.4.2.1.1 功能描述

返回上次复位的原因。该功能可用于定位导致异常复位的原因，也可以是系统可靠运行的一部分：不同的复位条件转向不同的复位流程。

4.4.2.1.2 参数/返回值说明

1. 参数：无
2. 返回值：

   返回chip复位原因，复位信息的MASK值，BIT0~14，复位源详情请参考csi_rsr_src_e枚举中定义。

3. 返回值说明表

返回值	说明	概述

复位信息	uint16_t 类型数值，复位信息的 position 值；具体复位信息，请参考 csi_rsr_e 中具体定义
	`typedef enum{`
	`RST_SRC_POR = (0x01ul << 0),`
	`RST_SRC_LVD = (0x01ul << 1),`
	`RST_SRC_EXT = (0x01ul << 2),`
	`RST_SRC_IWDT = (0x01ul << 4),`
	`RST_SRC_EMCM = (0x01ul << 6),`
	`RST_SRC_CPU = (0x01ul << 7),`
	`RST_SRC_SW = (0x01ul << 8),`
	`RST_SRC_CPUFAULT = (0x01ul << 9),`
	`RST_SRC_RAMERR = (0x01ul << 11),`
	`RST_SRC_EFLERR = (0x01ul << 12),`
	`RST_SRC_WWDT = (0x01ul << 13)`
	`}csi_rsr_src_e;`
		reliability.h

4.4.2.2 csi_clr_rst_reason

void csi_clr_rst_reason(uint16_t hwRstSrc)

4.4.2.2.1 功能描述

清除 chip 复位信息。

4.4.2.2.2 参数/返回值说明

1. 参数

   hwRstSrc：复位信息的 MASK 值，BIT0~13，复位源详情请参考 csi_rsr_src_e 枚举中定义。

2. 返回值：无

3. 参数说明

参数	说明	概述
复位信息	uint16_t 类型数值，复位信息的 MASK 值	

4.5 CHIP软件复位

4.5.1 API列表

API	说明	函数位置
csi_sys_swrst	Chip软件复位	reliability.c

4.5.2 API详细说明

4.5.2.1 csi_sys_swrst

void csi_sys_swrst(void)

4.5.2.1.1 功能描述

软件复位 mcu。

4.5.2.1.2 参数/返回值说明

1. 参数：无
2. 返回值：无

4.6 SWD lock/unlock函数

4.6.1 API列表

API	说明	函数位置
csi_swd_lock	Swd 管脚lock	reliability.c
csi_swd_unlock	Swd管脚 unlock	

4.6.2 API详细说明

4.6.2.1 csi_swd_lock

void csi_swd_lock (void)

4.6.2.1.1 功能描述

调用该函数之后，swd 管脚将不能被配置成其它 alternate function

4.6.2.1.2 参数/返回值说明

3. 参数：无

返回值：无

4.6.3 API详细说明

4.6.3.1 csi_swd_unlock

void csi_swd_unlock (void)

4.6.3.1.1 功能描述

调用该函数之后，swd 管脚将可以被配置成其它 alternate function，注意下次将会连不上 CDK，所以为了方便起见，当你使用了 swd 的复用功能后，可以在修改成复用功能之前加上合适的延时。

4.6.3.1.2 参数/返回值说明

4. 参数：无

返回值：无

4.7 UREG操作

4.7.1 API列表

API	说明	函数位置
csi_ureg_write	写用户寄存器	reliability.c
csi_ureg_read	读取用户寄存器	

4.7.2 API详细说明

4.7.2.1 csi_ureg_write

csi_error_t csi_ureg_write(csi_user_reg_e eUreg, uint32_t wValue)

4.7.2.1.1 功能描述

写用户寄存器，除 POR 复位外，其余原因导致的复位，寄存器值保持不变。

4.7.2.1.2 参数/返回值说明

1. 参数

   eUreg：用户寄存器选项，详见枚举定义 csi_user_reg_e

   wValue：用户写入寄存器值，eUreg = USER_REG0/ USER_REG1,为 32 位值，USER_REG2 时，为 16 位值。

2. 返回值：

   CSI_OK：初始化成功。

   CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	枚举量位置
eUreg	typedef enum{ USER_REG0 USER_REG1, USER_REG2, }csi_user_reg_e;	reliability.h
wValue	写入寄存器值，USER_REG0/ USER_REG1 为 32 位，其余为 16 位	

csi_error_t	csi_error_t 中定义值	在common.h中定义

4.7.2.2 csi_ureg_read

uint32_t csi_ureg_read(csi_user_reg_e eUreg)

4.7.2.2.1 功能描述

读取用存寄储器除 POR 复位外，其余原因导致的复位，寄存器值保持不变。

4.7.2.2.2 参数/返回值说明

1. 参数

   eUreg：用户寄存器选项，详见枚举定义 csi_user_reg_e

2. 返回值：寄存器中存储值

3. 参数/返回值说明

参数/返回值	说明	枚举量位置
eUreg	typedef enum{ USER_REG0 USER_REG1, USER_REG2, }csi_user_reg_e;	reliability.h
返回值	寄存器存储值，USER_REG0/ USER_REG1 为32位，其余为16位	

5 PM

APT32F171x 支持两种低功耗模式，SLEEP、DEEPSLEEP。在 csi 驱动中，有一个全局的宏 CONFIG_USER_PM。一旦开启，就必须定义进出低功耗模式前后的自定义处理函数。

5.1 API列表

Table 5-1 PM CSI接口函数

API	说明	函数位置
csi_pm_attach_callback	设置进出某个低功耗模式前后的用户自定义处理函数。	pm.c
csi_pm_config_wakeup_source	设置唤醒源。	
csi_pm_enter_sleep	进入低功耗模式。	
csi_pm_clk_enable	睡眠模式下使能/关闭时钟	

5.2 API详细说明

5.2.1 csi_pm_attach_callback

*void csi_pm_attach_callback(csi_pm_mode_e eMd, void *pBeforeSlp, void *pWkup)*

5.2.1.1 功能描述

这个函数受一个宏（CONFIG_USER_PM）控制。只有打开这个宏，函数才会加入编译。可以考虑在以下应用场景时打开这个宏：希望在系统进入低功耗模式前做一些状态和数据保存的操作，在系统从低功耗模式中恢复时恢复一些状态。建议在工程设置 compiler tab 下的 Define 中加入：CONFIG_USER_PM=1。

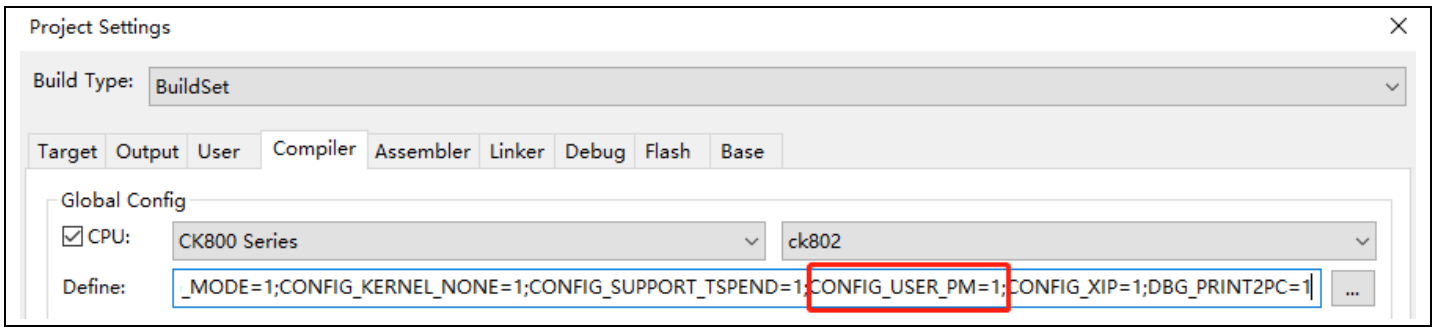


Figure 5-1 开启全局宏

5.2.1.2 参数/返回值说明

1. 参数：

   eMd：低功耗模式。

   pfBeforeSlp：进入低功耗模式前需调用的函数名。

   pfWkup：从低功耗模式出来后需调用的函数名。

2. 返回值：无。

3. 参数说明表

参数	说明	位置
eMd	可选值为csi_pm_mode_t枚举值中的一个。 typedef enum { PM_MODE_RUN = 0 PM_MODE_SLEEP, PM_MODE_DEEPSLEEP, } csi_pm_mode_e;	pm.h
pfBeforeSlp	(void *)空指针。传入进入低功耗模式前需要调用函数（用户自定义）的名字。	
pfWkup	(void *)空指针。从低功耗模式退出后调用函数（用户自定义）的名字。	用户自定义函数位置

5.2.2 csi_pm_enter_sleep

csi_error_t csi_pm_enter_sleep(csi_pm_mode_e eMode)

5.2.2.1 功能描述

调用函数即进入指定的低功耗模式。如果工程打开了宏（CONFIG_USER_PM），如Figure 5-1所示，会在进出低功耗模式前分别调用用户自定义函数。此时必须用csi_pm_attach_callback() 指定自定义函数，否则芯片会出现异常。如果不需要自定义函数，则需要关闭全局宏，即Figure 5-1中删除CONFIG_USER_PM=1。

5.2.2.2 参数/返回值说明

1. 参数：

   eMd：低功耗模式。

2. 返回值：

   csi_error_t型数据。当传入低功耗模式不在枚举变量范围时，会返回CSI_ERROR。

3. 参数说明表

参数	说明	位置
eMode	可选值为csi_pm_mode_e枚举值中的一个。 typedef enum { PM_MODE_RUN = 0 PM_MODE_SLEEP, PM_MODE_DEEPSLEEP, } csi_pm_mode_e;	pm.h

5.2.3 csi_pm_config_wakeup_source

csi_error_t csi_pm_config_wakeup_source(wakeupn_type_e eWkupSrc, bool bEnable)

5.2.3.1 功能描述

设置将系统从 DEEP-SLEEP 模式中唤醒的源。

注：任何使能的中断都可以将系统从 SLEEP 模式中唤醒。无需在这里配置。

5.2.3.2 参数/返回值说明

1. 参数：

   eWkupSrc：选择唤醒源。

   bEnable：使能/禁止。

2. 返回值：

   csi_error_t型数据。当传入唤醒源不在枚举变量范围时，会返回CSI_ERROR。

3. 参数说明表

参数/返回值	说明	位置
eWkupSrc	可选值为wakeupn_type_e枚举值中的一个。 typedef enum { WKUP_EXI0 = 0, WKUP_EXI1, WKUP_EXI2, WKUP_EXI3, WKUP_EXI4, WKUP_IWDT = 8, WKUP_LVD = 11 } csi_wakeup_src_e	pm.h
bEnable	ENABLE/DISABLE	
csi_error_t	csi_error_t 中定义值	csi_error_t

5.2.4 csi_pm_clk_enable

void csi_pm_clk_enable(csi_pm_clk_e eOsc, bool bEnable)

5.2.4.1 功能描述

睡眠模式下使能/禁止某时钟。

5.2.4.2 参数/返回值说明

1. 参数

    1. eOsc：选择某个时钟，枚举定义详见 csi_pm_clk_e。

    2. bEnable：使能/禁止（ENABLE/DISABLE）

2. 返回值：无。

3. 参数/返回值说明表

参数	说明	位置

ePower	可选值为csi_snooze_power_e枚举值中的一个。
	typedef enum {
	SP_IDLE_PCLK = (0x01ul << 8), ///< IDLE_PCLK for sleep
	SP_IDLE_HCLK = (0x01ul << 9), ///< DLE_HCLK for sleep
	DP_ISOSC = (0x01ul << 12), ///< ISOSC for deepslee
	DP_IMOSC = (0x01ul << 13), ///< IMOSC for deepsleep
	DP_EMOSC = (0x01ul << 15) ///< EMOSCfordeepsleep
	} csi_pm_clk_e;
		pm.h

bEnable	Bool 类型数值，ENBALE/DISABLE	common.h

6 CRC

ATP CSI接口CRC的设计中，提供CRC模块的初始化、CRC-CCITT、CRC-16、CRC-32计算函数接口。

6.1 API列表

Table6-1CRCCSI接口函数

API	说明	函数位置
csi_crc_init	CRC模块初始化	crc.c
csi_crc_rst	CRC模块复位	
csi_crc16	CRC-16转换函数	
csi_crc16_ccitt	CRC-16/CCITT转换函数	
csi_crc16_itu	CRC-16 XMODEM转换函数	
csi_crc32_be	CRC-32转换函数	

6.2 API详细说明

6.2.1 csi_crc_init

void csi_crc_init(void)

6.2.1.1 功能描述

CRC模块初始化。

6.2.1.2 参数/返回值说明

1. 参数：无。
2. 返回值：无。

6.2.2 csi_crc_rst

void csi_crc_rst(void)

6.2.2.1 功能描述

软件复位CRC模块。

6.2.2.2 参数/返回值说明

1. 参数：无参数。
2. 返回值：无返回值。

6.2.3 csi_crc16

uint16_t csi_crc16(uint16_t hwCrcSeed, uint8_t pbyData, uint32_t wSize)*

6.2.3.1 功能描述

CRC-16模式计算。

6.2.3.2 参数/返回值说明

1. 参数

   hwCrcSeed：CRC计算种子值。

   pbyData：需要计算的数据指针。

   wSize：需要计算的数据长度。

2. 返回值

   CRC计算结果值。

6.2.4 csi_crc16_ccitt

*uint16_t csi_crc16_ccitt(uint16_t hwCrcSeed, uint8_t *pbyData, uint32_t wSize)*

6.2.4.1 功能描述

CRC-16/CCITT模式计算。

6.2.4.2 参数/返回值说明

1. 参数

   hwCrcSeed：CRC计算种子值。

pbyData：需要计算的数据指针。

wSize：需要计算的数据长度。

2. 返回值

CRC计算结果值。

6.2.5 csi_crc16_itu

uint16_t csi_crc16_itu(uint16_t hwCrcSeed, uint8_t pbyData, uint32_t wSize)*

6.2.5.1 功能描述

CRC-16 XMODEM模式计算。

6.2.5.2 参数/返回值说明

1. 参数

hwCrcSeed：CRC计算种子值。

pbyData：需要计算的数据指针。

wSize：需要计算的数据长度。

2. 返回值

CRC计算结果值。

6.2.6 csi_crc32_be

uint32_t csi_crc32_be(uint32_t wCrcSeed, uint8_t pbyData, uint32_t wSize)*

6.2.6.1 功能描述

CRC-32模式计算

6.2.6.2 参数/返回值说明

1. 参数

hwCrcSeed：CRC计算种子值。

pbyData：需要计算的数据指针。

wSize：需要计算的数据长度。

2. 返回值

7 HWDIV

当前 csi 驱动中，客户可以把 HWDIV 看作是透明的。直接用操作符“/”即可，如 68/2，编译器会根据数据类型自动调用 hwdiv 的库函数。

8 IFC

APT32F102x 的内部 flash 包括两个区域，一个是 64K/32K 的 PFLASH，用于程序代码的存储，一个是 4K 的 DFLASH，用于用户数据的存储。这两个区域最大的区别是扇区大小不同。

无论是 DFLASH 还是 PFLASH，一次写操作之前必需有一次擦除操作（以扇区为单位）。从 FLASH 寿命角度考虑，擦和写的次数需要对等。正是出于这个考虑，驱动不再支持单独的擦除操作，写操作函数中实现先擦除再写。

8.1 API列表

Table 8-1 IFC CSI接口函数

API	说明	函数位置
csi_ifc_read	读取flash区域的内容，支持PFLASH和DFLASH	ifc.c
csi_ifc_program	往flash区域写入内容，支持PFLASH和DFLASH	
csi_ifc_get_status	获得flash状态（error, busy）	

8.2 API详细说明

8.2.1 csi_ifc_read

*csi_error_t csi_ifc_read(csp_ifc_t *ptIfcBase, uint32_t wAddr, uint32_t *wData, uint32_t wDataNum)*

8.2.1.1 功能描述

读取 flash 区域的内容，支持 PFLASH 和 DFLASH。发生以下情况返回错误：

1. 传入地址没有字对齐。
2. 地址不在系统 flash（PFLASH+DFLASH）空间

8.2.1.2 参数说明

参数	说明	位置
ptIfcBase	指向IFC控制寄存器结构体的指针，用于进行寄存器操作。	csp_ifc.h
wAddr	读取Flash数据的首址	
wData	指向目标数据首址的指针	

wDataNum	一次读取数据的长度（单位：byte）	

8.2.1.3 返回值说明

返回值类型	说明	位置
csi_error_t	CSI_ERROR/CSI_OK	common.h

8.2.2 csi_ifc_program

*csi_error_t csi_ifc_program(csp_ifc_t *ptIfcBase, uint32_t wAddr, uint32_t *pwData, uint32_t wDataNum)*

8.2.2.1 功能描述

往 flash 区域写入内容，带校验功能。支持 PFLASH 和 DFLASH。发生以下情况返回错误：

- 传入地址没有字对齐
- 地址不在系统 flash（PFLASH+DFLASH）空间
- 校验失败

8.2.2.2 参数说明

参数	说明	位置
ptIfcBase	指向IFC控制寄存器结构体的指针，用于进行寄存器操作。	csp_ifc.h
wAddr	操作Flash的目标首址	
wData	指向源数据首址的指针	
wDataNum	一次写入的数据的长度（单位：byte）	

8.2.2.3 返回值说明

返回值类型	说明	位置
csi_error_t	CSI_ERROR/CSI_OK	common.h

8.2.3 csi_ifc_get_status

*csi_ifc_status_t csi_ifc_get_status(csp_ifc_t *ptIfcBase)*

8.2.3.1 功能描述

读取 Flash 的状态。

8.2.3.2 参数说明

参数	说明	位置
ptIfcBase	指向IFC控制寄存器结构体的指针，用于进行寄存器操作。	csp_ifc.h

8.2.3.3 返回值说明

返回值类型	说明	位置

csi_ifc_status_t	typedef struct {
	uint32_t busy : 1; // busy 状态
	uint32_t error : 1; //错误状态（IFC_RISR[xxx_ERR]）标志
	} csi_ifc_status_t;
		ifc.h

9 IWDT

9.1 概述

嵌入式系统中，为了使系统在异常情况下能自动复位，保证系统健壮性，一般都需要引入看门狗。IWDT是一个独立看门狗，它的时钟源是ISOSC。上电时IWDT是否使能由User Option的高16位决定，可以通过读取SYSCON_OPT0[IWDTEN]获得User Option中对应的缺省状态。

IWDT运行后计数器开始计数，计数器溢出前没有被复位，计数器溢出会对MCU产生复位信号使系统复位。系统正常运行时，需要在计数器溢出前对计数器清零(喂狗)，不让复位产生。如系统正常运行，喂狗正常。一旦程序异常，不能正常喂狗，则系统复位。

在默认情况下，IWDT是使能的，如果没有喂狗的操作，会在8s后复位芯片。

当IWDT工作时，ISOSC缺省使能。任何尝试关闭ISOSC的操作都会触发命令错误中断。

APT CSI接口提供了IWDT相关配置和操作。

9.2 API列表

Table 9-1 IWDT CSI接口函数

API	说明	函数位置
csi_iwdt_init	初始化看门狗	iwdt.c
csi_iwdt_open	打开看门狗(开始工作)	
csi_iwdt_close	关闭看门狗(停止工作)	
csi_iwdt_feed	看门狗喂狗	
csi_iwdt_irq_enable	使能看门狗中断	
csi_iwdt_is_running	检测看门狗工作状态	
csi_iwdt_get_remaining_time	获取看门狗复位剩余时间	
csi_iwdt_debug_enable	看门狗debug模式使能	

9.3 API详细说明

9.3.1 csi_iwdt_init

csi_error_t csi_iwdt_init(csi_iwdt_to_e eTimeOut)

9.3.1.1 功能描述

初始化看门狗。

9.3.1.2 参数/返回值说明

1. 参数

eTimeOut：看门狗溢出时间，枚举定义详见csi_iwdt_to_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
eTimeOut	typedef enum{ IWDT_TO_128 = 0, //128ms IWDT_TO_256, //256ms IWDT_TO_512, //512ms IWDT_TO_1024, //1024ms IWDT_TO_2048, //2048ms IWDT_TO_3072, //3072ms IWDT_TO_4096, //4096ms IWDT_TO_8192 //8192ms }csi_iwdt_to_e;	看门狗溢出(系统复位)时间，有8档选择，在iwdt.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

9.3.2 csi_iwdt_open

void csi_iwdt_open(void)

9.3.2.1 功能描述

打开看门狗，即启动看门狗。

9.3.2.2 参数/返回值说明

1. 参数：无参数。
2. 返回值：无返回值。

9.3.3 csi_iwdt_close

void csi_iwdt_close(void)

9.3.3.1 功能描述

关闭看门狗，即停止看门狗

9.3.3.2 参数/返回值说明

1. 参数：无参数。
2. 返回值：无返回值。

9.3.4 csi_iwdt_feed

void csi_iwdt_feed(void)

9.3.4.1 功能描述

看门狗喂狗函数

9.3.4.2 参数/返回值说明

1. 参数：无参数
2. 返回值：无返回值

9.3.5 csi_iwdt_irq_enable

void csi_iwdt_irq_enable(csi_iwdt_alarm_e eAlarmTo, bool bEnable)

9.3.5.1 功能描述

使能看门狗报警中断。

9.3.5.2 参数/返回值说明

1. 参数

eAlarmTo：报警中断时间，枚举定义详见csi_iwdt_alarm_e。

bEnable：使能/禁止中断，ENABLE/DISABLE。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置
eAlarmTo	typedef enum{ IWDT_ALARMTO_1_8 = 0, IWDT_ALARMTO_2_8, IWDT_ALARMTO_3_8, IWDT_ALARMTO_4_8, IWDT_ALARMTO_5_8, IWDT_ALARMTO_6_8, IWDT_ALARMTO_7_8 }csi_iwdt_alarm_e;	看门狗报警中断时间有7档选择，即占总溢出时间的n/8，n的范围：1~7 在iwdt.h中定义
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

9.3.6 csi_iwdt_is_running

bool csi_iwdt_is_running(void)

9.3.6.1 功能描述

检测看门狗工作状态。

9.3.6.2 参数/返回值说明

1. 参数：无参数。

2. 返回值

ture：正在工作。

false：停止工作。

3. 返回值说明

返回值类型	说明	概述
bool	Ture：false(1/0)	ture：数值为1(真) false：数值为0(假)

9.3.7 csi_iwdt_get_remaining_time

uint32_t csi_iwdt_get_remaining_time(void)

9.3.7.1 功能描述

获取看门狗复位剩余时间

9.3.7.2 参数/返回值说明

1. 参数：无参数。

2. 返回值：

复位剩余时间，单位ms。

3. 返回值说明

返回值类型	说明	概述
return value	uint32_t 类型数值	复位剩余时间，单位ms

9.3.8 csi_iwdt_debug_enable

void csi_iwdt_debug_enable(bool bEnable)

9.3.8.1 功能描述

使能看门狗debug模式

9.3.8.2 参数/返回值说明

1. 参数bEnable：使能/禁止debug模式，默认禁止，ENABLE/DISABLE。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述及其定义位置
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能 DISABLE：禁止 在common.h中定义

10 WWDT

10.1 概述

窗口型看门狗，作为可靠性保护逻辑，用于监测当前程序运行状况。当外部干扰或不可预见的逻辑错误发生时，造成当前程序运行错误，看门狗逻辑可以在预设时间周期结束时产生系统复位信号。看门狗计数器通过软件刷新防止溢出而产生复位，刷新必须在预设的时间窗口没进行才有效，否则刷新事件也会触发系统复位信号。看门狗一旦开启，不能被关闭，除非系统复位，系统复位时看门狗默认关闭。APT CSI接口提供了WWDT相关配置和操作。

Table 10-1 WWDT CSI接口函数

API	说明	函数位置
csi_wwdt_init	初始化看门狗	wwdt.c
csi_wwdt_set_window_time	设置窗口时间	
csi_iwdt_open	打开看门狗(开始工作)	
csi_wwdt_feed	看门狗喂狗	
csi_wwdt_irq_enable	使能看门狗中断	
csi_wwdt_is_running	检测看门狗工作状态	
csi_wwdt_get_remaining_time	获取看门狗复位剩余时间	
csi_wwdt_debug_enable	看门狗debug模式使能	

10.2 API详细说明

10.2.1 csi_wwdt_init

csi_error_t csi_wwdt_init(uint32_t wTimeOut)

10.2.1.1 功能描述

初始化看门狗

10.2.1.2 参数/返回值说明

1. 参数

wTimeOut：看门狗溢出时间，单位ms。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
wTimeOut	uint32_t 类型数值，单位：ms	看门狗溢出(系统复位)时间，单位ms
csi_error_t	csi_error_t 中定义值	在common.h中定义

10.2.2 csi_wwdt_set_window_time

csi_error_t csi_wwdt_set_window_time(uint32_t wTimeOut)

10.2.2.1 功能描述

设置看门狗窗口时间

10.2.2.2 参数/返回值

1. 参数

wTimeOut：看门狗窗口时间，单位ms。

2. 返回值

CSI_OK：设置成功。

CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
wTimeOut	uint32_t 类型数值，单位：ms	看门狗窗口时间，单位ms
csi_error_t	csi_error_t中定义值	在common.h中定义

10.2.3 csi_wwdt_open

void csi_wwdt_open(void)

10.2.3.1 功能描述

打开看门狗，即启动看门狗

10.2.3.2 参数/返回值说明

1. 参数：无参数。
2. 返回值：无返回值。

10.2.4 csi_wwdt_feed

void csi_wwdt_feed(void)

10.2.4.1 功能描述

看门狗喂狗函数

10.2.4.2 参数/返回值说明

1. 参数：无参数。
2. 返回值：无返回值。

10.2.5 csi_wwdt_irq_enable

void csi_wwdt_irq_enable(bool bEnable)

10.2.5.1 功能描述

使能看门狗报警中断。

10.2.5.2 参数/返回值说明

1. 参数

   bEnable：使能/禁止中断，ENABLE/DISABLE。

2. 返回值：无返回值。
3. 参数说明

参数	说明	概述及其定义位置
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

10.2.6 csi_wwdt_is_running

bool csi_wwdt_is_running(void)

10.2.6.1 功能描述

检测看门狗工作状态

10.2.6.2 参数/返回值说明

1. 参数：无参数。
2. 返回值

   ture：正在工作。

   false：停止工作。
3. 返回值说明

返回值	说明	概述
return value	Bool 类型数值，ture/false(1/0)	ture：数值为1(真) false：数值为0(假)

10.2.7 csi_wwdt_get_remaining_time

uint32_t csi_wwdt_get_remaining_time(void)

10.2.7.1 功能描述

获取看门狗复位剩余时间

10.2.7.2 参数/返回值说明

1. 参数：无参数。
2. 返回值

   复位剩余时间，单位ms。
3. 返回值说明

返回值	说明	概述
return value	uint32_t 类型数值，单位：ms	复位剩余时间，单位ms

10.2.8 csi_wwdt_debug_enable

void csi_wwdt_debug_enable(bool bEnable)

10.2.8.1 功能描述

使能看门狗debug模式

10.2.8.2 参数/返回值说明

1. 参数

   bEnable：使能/禁止debug模式，默认禁止，ENABLE/DISABLE。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述及其定义位置
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能 DISABLE：禁止 在common.h中定义

11 ETCB

11.1 概述

ETCB的功能是实现一个IP触发另一个IP，相当于在两个不同的外设之间架起了一座桥。这种硬件的桥接，大大增加了系统的反应速度，同时减少了CPU处理中断的压力。

API CSI接口函数提供了相关配置和操作。

11.2 API列表

Table 11-1 ETCB CSI接口函数

API	说明	函数位置
csi_etb_init	初始化ETB	etb.c
csi_etb_ch_alloc	申请一个ETB通道	
csi_etb_ch_free	释放一个ETB通道	
csi_etb_ch_config	配置ETB通道	
csi_etb_ch_swtrg	软件触发ETB通道	
csi_etb_ch_start	启动(使能)ETB通道	
csi_etb_ch_stop	关闭(禁止)ETB通道	

11.3 API详细说明

11.3.1 csi_etb_init

void csi_etb_init(void)

11.3.1.1 功能描述

初始化ETB模块(使能模块)

11.3.1.2 参数/返回值说明

1. 参数：无参数。

2. 返回值：无返回值。

3. 参数/返回值说明

11.3.2 csi_etb_ch_alloc

int32_t csi_etb_ch_alloc(csi_etb_ch_type_e eChType)

11.3.2.1 功能描述

申请一个ETB通道

11.3.2.2 参数/返回值说明

1. 参数

eChType：ETB通道类型，枚举详见csi_etb_ch_type_e。

2. 返回值

通道号/错误信息。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举定义位置
eChType	typedef enum { ETB_ONE_TRG_ONE = 0, ETB_ONE_TRG_MORE, } csi_etb_ch_type_e;	有两种通道类型： 单个源触发单个目标(通道3~31) 单个源触发多个目标(通道0~2) 在etb.h中定义
return value	int32_t 类型数值，范围：-1 ~ 31	-1：获取通道失败 0~31：为通道号，申请成功

11.3.3 csi_etb_ch_free

void csi_etb_ch_free(csi_etb_chid_e eChId)

11.3.3.1 功能描述

释放一个ETB通道

11.3.3.2 参数/返回值说明

1. 参数

eChId：ETB通道ID号，枚举详见csi_etb_chid_e。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置
eChId	typedef enum { ETB_CH0 = 0, ETB_CH1, ETB_CH2, ETB_CH3, ETB_CH4, ETB_CH5, ETB_CH6, ETB_CH7, ETB_CH8, ETB_CH9, ETB_CH10, ETB_CH11, ETB_CH12, ETB_CH13, ETB_CH14, ETB_CH15, ETB_CH16, ETB_CH17, ETB_CH18, ETB_CH19, ETB_CH20, ETB_CH21, ETB_CH22, ETB_CH23, ETB_CH24, ETB_CH25, ETB_CH26, ETB_CH27, ETB_CH28, ETB_CH29, ETB_CH30, ETB_CH31 } csi_etb_ch_e;	ETB有32个通道： ETB_CH0 ~ ETB_CH31 数值：0~31 在etb.h中定义

11.3.4 csi_etb_ch_config

*csi_error_t csi_etb_ch_config(csi_etb_chid_e eChId, csi_etb_config_t *ptConfig)*

11.3.4.1 功能描述

配置ETB通道。

11.3.4.2 参数/返回值说明

1. 参数

eChId：ETB通道ID号，枚举详见csi_etb_chid_e。

ptConfig：ETB通道配置参数结构体指针，结构体详见csi_etb_config_t。

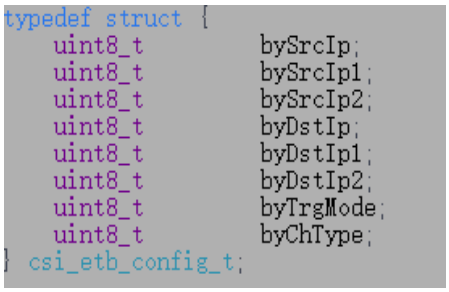
2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
eChId	typedef enum { ETB_CH0 = 0, ETB_CH1, ETB_CH2, ETB_CH3, ETB_CH4, ETB_CH5, ETB_CH6, ETB_CH7, ETB_CH8, ETB_CH9, ETB_CH10, ETB_CH11, ETB_CH12, ETB_CH13, ETB_CH14, ETB_CH15, ETB_CH16, ETB_CH17, ETB_CH18, ETB_CH19, ETB_CH20, ETB_CH21, ETB_CH22, ETB_CH23, ETB_CH24, ETB_CH25, ETB_CH26, ETB_CH27, ETB_CH28, ETB_CH29, ETB_CH30, ETB_CH31 } csi_etb_ch_e;	ETB有32个通道： ETB_CH0 ~ ETB_CH31 数值：0~31 在etb.h中定义

ptConfig		8个配置参数： bySrcIp：触发源0事件 bySrcIp1：触发源1事件 bySrcIp2：触发源2事件 byDstIp：触发目标0事件 byDstIp1：触发目标1事件 byDstIp2：触发目标2事件 byTrgMode：触发模式 byChType：通道类型 触发模式：硬件、软件两种模式；详见枚举csi_etb_trg_mode_e 通道类型：参阅7.3.2.2参数说明
csi_error_t	csi_error_t 中定义值	在common.h中定义

11.3.5 csi_etb_ch_swtrg

void csi_etb_ch_swtrg(csi_etb_chid_e eChId)

11.3.5.1 功能描述

软件触发ETB通道

11.3.5.2 参数/返回值说明

1. 参数

   eChId：ETB通道ID号，枚举详见csi_etb_chid_e。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置

eChId	typedef enum { ETB_CH0 = 0, ETB_CH1, ETB_CH2, ETB_CH3, ETB_CH4, ETB_CH5, ETB_CH6, ETB_CH7, ETB_CH8, ETB_CH9, ETB_CH10, ETB_CH11, ETB_CH12, ETB_CH13, ETB_CH14, ETB_CH15, ETB_CH16, ETB_CH17, ETB_CH18, ETB_CH19, ETB_CH20, ETB_CH21, ETB_CH22, ETB_CH23, ETB_CH24, ETB_CH25, ETB_CH26, ETB_CH27, ETB_CH28, ETB_CH29, ETB_CH30, ETB_CH31 } csi_etb_ch_e;	ETB有32个通道： ETB_CH0 ~ ETB_CH31 数值：0~31 在etb.h中定义

11.3.6 csi_etb_ch_start

void csi_etb_ch_start(csi_etb_chid_e eChId)

11.3.6.1 功能描述

启动(使能)ETB通道

11.3.6.2 参数/返回值说明

1. 参数

   eChId：ETB通道ID号，枚举详见csi_etb_chid_e。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置
eChId	typedef enum { ETB_CH0 = 0, ETB_CH1, ETB_CH2, ETB_CH3, ETB_CH4, ETB_CH5, ETB_CH6, ETB_CH7, ETB_CH8, ETB_CH9, ETB_CH10, ETB_CH11, ETB_CH12, ETB_CH13, ETB_CH14, ETB_CH15, ETB_CH16, ETB_CH17, ETB_CH18, ETB_CH19, ETB_CH20, ETB_CH21, ETB_CH22, ETB_CH23, ETB_CH24, ETB_CH25, ETB_CH26, ETB_CH27, ETB_CH28, ETB_CH29, ETB_CH30, ETB_CH31 } csi_etb_ch_e;	ETB有32个通道： ETB_CH0 ~ ETB_CH31 数值：0~31 在etb.h中定义

11.3.7 csi_etb_ch_stop

void csi_etb_ch_stop(csi_etb_chid_e eChId)

11.3.7.1 功能描述

关闭(禁止)ETB通道

11.3.7.2 参数/返回值说明

1. 参数

   p eChId：ETB通道ID号，枚举详见csi_etb_chid_e。

2. 返回值：无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置
eChId	typedef enum { ETB_CH0 = 0, ETB_CH1, ETB_CH2, ETB_CH3, ETB_CH4, ETB_CH5, ETB_CH6, ETB_CH7, ETB_CH8, ETB_CH9, ETB_CH10, ETB_CH11, ETB_CH12, ETB_CH13, ETB_CH14, ETB_CH15, ETB_CH16, ETB_CH17, ETB_CH18, ETB_CH19, ETB_CH20, ETB_CH21, ETB_CH22, ETB_CH23, ETB_CH24, ETB_CH25, ETB_CH26, ETB_CH27, ETB_CH28, ETB_CH29, ETB_CH30, ETB_CH31 } csi_etb_ch_e;	ETB有32个通道： ETB_CH0 ~ ETB_CH31 数值：0~31 在etb.h中定义

12 ADC

12.1 概述

ADC是Analog-to-Digital Converter的缩写。指模/数转换器或者模拟/数字转换器，是指将连续变量的模拟信号转换为离散的数字信号的器件，改ADC位12位ADC。APT CSI接口提供了ADC包括轮询、中断模式，单通道以及多通道采样相关配置和操作。

12.2 API列表

Table 12-1 时钟CSI接口函数

API	说明	函数位置
csi_adc_init	初始化ADC	adc.c
csi_adc_set_seqx	配置ADC转换序列	
csi_adc_set_buffer	配置ADC采样数据缓存(buffer)	
csi_adc_start	启动AD转换	
csi_adc_stop	停止AD转换	
csi_adc_conv_mode	配置ADC转换模式	
csi_adc_conv_pri	配置ADC序列转换优先级	
csi_adc_read_channel	获取ADC转换序列某通道数据	
csi_adc_read_seqx	获取ADC转换序列所有通道数据	
csi_adc_set_vref	配置ADC参考电压	
csi_adc_freq_div	配置ADC采样分频系数	
csi_adc_get_freq	获取ADC采样频率	
csi_adc_int_enable	使能ADC中断	
csi_adc_set_cmp0	配置ADC采样比较寄存器0比较功能	
csi_adc_set_cmp1	配置ADC采样比较寄存器1比较功能	
csi_adc_get_status	获取ADC数据转换状态	
csi_adc_clr_status	清除ADC数据转换状态	
csi_adc_set_sync	配置ADC外部同步触发输入	
csi_adc_rearm_sync	设置同步触发模式自动REARM	

csi_adc_set_evtrg	配置ADC事件触发输出	
csi_adc_fvrout_enable	使能ADC固定参考电压源	
csi_adc_bufout_enable	使能ADC 内部电压输出（1v）	

12.3 API详细说明

12.3.1 csi_adc_init

*csi_error_t csi_adc_init(csp_adc_t *ptAdcBase, csi_adc_config_t *ptAdcCfg)*

12.3.1.1 功能描述

初始化ADC

12.3.1.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

ptAdcCfg：ADC配置结构体指针，结构体定义详见csi_adc_config_t。

2. 返回值

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置

ptAdcBase	参数：ADC0
	csp_adc_t *ADC0 = (csp_adc_t *)(APB_ADC0_BASE);
	typedef struct
	{
	__OM uint32_t ECR;
	__OM uint32_t DCR;
	__IM uint32_t PMSR;
	__IM uint32_t RSVD0;
	__OM uint32_t CR;
	__IOM uint32_t MR;
	__IOM uint32_t SHR;
	__OM uint32_t CSR;
	__IM uint32_t SR;
	__OM uint32_t IER;
	__OM uint32_t IDR;
	__IM uint32_t IMR;
	__IOM uint32_t SEQ[16];
		系统有一个ADC，即ADC(0)，定义了对应的结构体指针ADC0，指向系统ADC基地址。
		ADC0的指针定义在devices.c,指针类型定义csp_adc.h

	__IOM uint32_t PRI; __IOM uint32_t TDL0; __IOM uint32_t TDL1; __IOM uint32_t SYNCR; __IOM uint32_t TRGFCR; __IOM uint32_t TRGFWR; __IOM uint32_t EVTRG; __IOM uint32_t EVPS; __IOM uint32_t EVSWF; __IOM uint32_t RSVD2[27]; __IM uint32_t DR[16]; __IOM uint32_t CMP0; __IOM uint32_t CMP1; __IOM uint32_t DRMASK; } csp_adc_t;	
ptAdcCfg	typedef struct { uint8_t byClkDiv; uint8_t bySampHold; uint8_t byConvMode; uint8_t byVrefSrc; uint32_t wInt; csi_adc_seq_t *ptSeqCfg; } csi_adc_config_t;	初始化配置参数： byClkDiv：分频系数 bySampHold：采样保持时间 byConvMode：转换模式(连续/单次) byVrefSrc：参考电压源 wInter：中断选择 ptSeqCfg：csi_adc_seq_t类型指针 在adc.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.2 csi_adc_set_seqx

*csi_error_t csi_adc_set_seqx(csp_adc_t *ptAdcBase, csi_adc_seq_t *ptSeqx, uint8_t byChNum)*

12.3.2.1 功能描述

配置ADC转换序列

12.3.2.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

ptSeqx：ADC转换序列结构体指针，结构体定义详见csi_adc_seq_t。

byChNum：转换序列总得通道数。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	

ptSeqx	typedef struct {
	uint8_t byInChnl; //adc input channel
	uint8_t byRepCnt; //continuous repeat sample count
	uint8_t byAvgCof; //average coefficient
	uint8_t byTrgSrc; //trigger source
	} csi_adc_seq_t;
		转换序列参数：
		byInChnl：输入通道
		byRepCnt：连续重复采样次数
		byAvgCof：平均系数
		byTrgSrc：同步输入触发源
		在adc.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.3 csi_adc_set_buffer

*csi_error_t csi_adc_set_buffer(uint16_t *phwData, uint16_t hwRdLen)*

12.3.3.1 功能描述

配置ADC采样数据缓存(用户定义数据缓存(buffer和采样深度)，通过此API函数传入)

12.3.3.2 参数/返回值说明

1. 参数

   phwData：ADC转换序列采样值缓存指针，指向采样buffer首地址；采样数据存放在该指针指向的buffer中。

   hwRdLen：ADC转换序列通道采样值深度，即每通道采样数值次数。

2. 返回值

   CSI_OK：配置成功。

   CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
phwData	uint16_t 类型指针，指向用户定义的采样数值buffer	指针，指向采样数据缓存首地址，数据缓存由用户定义，缓存数组可以是一维、或者二维，取决于采样深度(通道采样次数)。
hwRdLen	uint16_t 类型数据，通道采样深度	每通道采样次数 1：数据缓存为一维数组 大于1：数据缓存为二维数组
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.4 csi_adc_start

*csi_error_t csi_adc_start(csp_adc_t *ptAdcBase)*

12.3.4.1 功能描述

启动AD转换

12.3.4.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

2. 返回值

   CSI_OK：启动成功。

   CSI_ERROR：启动失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.5 csi_adc_stop

*csi_error_t csi_adc_stop(csp_adc_t *ptAdcBase)*

12.3.5.1 功能描述

停止AD转换(连续转换模式有效，单次转换模式不可关闭，转换序列结束时停止)

12.3.5.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

2. 返回值

   CSI_OK：设置成功。

   CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.6 csi_adc_conv_mode

*void csi_adc_conv_mode(csp_adc_t *ptAdcBase, csi_adc_conv_mode_e eConvMode)*

12.3.6.1 功能描述

配置ADC转换模式

12.3.6.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

   eConvMode：转换模式，枚举定义详见csi_adc_conv_mode_e。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
eConvMode	typedef enum { ADC_CONV_ONESHOT= 0, ADC_CONV_CONTINU= 1 }csi_adc_conv_mode_e;	两种模式：单次转换、连续转换 在adc.h中定义。

12.3.7 csi_adc_conv_pri

*void csi_adc_conv_pri(csp_adc_t *ptAdcBase, uint8_t byPri)*

12.3.7.1 功能描述

配置ADC转换序列优先级

12.3.7.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

byPri：优先级数值，小于该值的转换序列只有被触发时才会转换，大于等于该值的序列正常转换。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
byPri	uint8_t 类型数值，优先级控制数值	用户AD采样序列配置为如下：SEQ0~SEQ5，byPri = 2，那么AD启动时采样从SEQ2启动，即SEQ2~SEQ5正常转换；SEQ0~SEQ1只有在被触发的时候才会转换，否则不会转换。
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.8 csi_adc_read_channel

uint8_t csi_pin_get_num(pin_name_e ePinName)

12.3.8.1 功能描述

获取ADC转换序列指定通道采样数值

12.3.8.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

   byChIdx：ADC转换序列中的通道ID号，即具体哪个通道，用户选择。

2. 返回值

   ADC转换序列中某一通道采样值(用户指定获取)。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t类型指针，ADC0；请参阅12.3.1.2参数说明	
byChIdx	uint8_t 类型数值，采样通道号：0~15	转换序列中任一通道号
return value	uint16_t 类型数值，数值范围：0~7FF	ADC某通道采样数值

12.3.9 csi_adc_read_seqx

*csi_error_t csi_adc_read_seqx(csp_adc_t *ptAdcBase)*

12.3.9.1 功能描述

获取ADC采样序列所有通道数据

12.3.9.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

2. 返回值

   CSI_OK：获取成功，采样数值存放于用户定义缓存中，请参阅8.3.3 csi_adc_set_buffer部分。

   CSI_ERROR：获取失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.10 csi_adc_set_vref

*void csi_adc_set_vref(csp_adc_t *ptAdcBase, csi_adc_vref_e eVrefSrc)*

12.3.10.1 功能描述

配置ADC参考电压

12.3.10.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

eVrefSrc：参考电压源，枚举定义详见csi_adc_vref_e。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
eVrefSrc	typedef enum{ ADCVERF_VDD_VSS = (0x00ul), ADCVREF_VREFP_VSS = (0x01ul), ADCVREF_FVR2048_VSS = (0x02ul), ADCVREF_FVR4096_VSS = (0x03ul), ADCVREF_INTVREF_VSS = (0x04ul), ADCVREF_VDD_VREFN = (0x08ul), ADCVREF_VREFP_VREFN = (0x09ul), ADCVREF_FVR2048_VREFN = (0x0aul), ADCVREF_FVR4096_VREFN = (0x0bul), ADCVREF_INTVREF_VREFN = (0x0cul) }csi_adc_vref_e;	外部参考源有10中选择，默认选择VDD_VSS。 在adc.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.11 csi_adc_freq_div

*uint32_t csi_adc_freq_div(csp_adc_t *ptAdcBase, uint8_t byDiv)*

12.3.11.1 功能描述

配置ADC采样分频系数

12.3.11.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

   byDiv：分频系数。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
byDiv	uint8_t 类型数值，范围：0~62	byDiv = 0和byDiv=1一样，分频系数为1

12.3.12 csi_adc_get_freq

*uint32_t csi_adc_get_freq(csp_adc_t *ptAdcBase)*

12.3.12.1 功能描述

获取ADC采样频率

12.3.12.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

2. 返回值

   ADC采样频率，即ADC工作频率。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
return value	uint32_t 类型数值，单位：Hz	ADC工作频率

12.3.13 csi_adc_int_enable

*void csi_adc_int_enable(csp_adc_t *ptAdcBase, csi_adc_intsrc_e eIntSrc, bool bEnable)*

12.3.13.1 功能描述

使能ADC中断

12.3.13.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

eIntSrc：BT中断源，枚举定义详见csi_adc_intsrc_e。

bEnable：使能/禁止中断，ENABLE/DISABLE。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	

eIntSrc	typedef enum{
	ADC_INTSRC_NONE = (0x00uL << 0), //no interrupt
	ADC_INTSRC_EOC = (0x01uL << 0),
	ADC_INTSRC_READY= (0x01uL << 1),
	ADC_INTSRC_OVR = (0x01uL << 2),
	ADC_INTSRC_CMP0H= (0x01uL << 4),
	ADC_INTSRC_CMP0L= (0x01uL << 5),
	ADC_INTSRC_CMP1H= (0x01uL << 6),
	ADC_INTSRC_CMP1L= (0x01uL << 7),
	//SEQX0-15
	ADC_INTSRC_SEQ0 = (0x01uL<<16),
	ADC_INTSRC_SEQ1 = (0x01uL << 17),
	ADC_INTSRC_SEQ2 = (0x01uL << 18),
	ADC_INTSRC_SEQ3 = (0x01uL << 19),
		有23个中断源，包含16个ADC序列转换中断(序列号0~15)和其余7个中断，ADC转换采样中断模式，建议采样序列转换中断。
		在adc.h中定义

	ADC_INTSRC_SEQ4 = (0x01uL << 20), ADC_INTSRC_SEQ5 = (0x01uL << 21), ADC_INTSRC_SEQ6 = (0x01uL << 22), ADC_INTSRC_SEQ7 = (0x01uL << 23), ADC_INTSRC_SEQ8 = (0x01uL << 24), ADC_INTSRC_SEQ9 = (0x01uL << 25), ADC_INTSRC_SEQ10= (0x01uL << 26), ADC_INTSRC_SEQ11= (0x01uL << 27), ADC_INTSRC_SEQ12= (0x01uL << 28), ADC_INTSRC_SEQ13= (0x01uL << 29), ADC_INTSRC_SEQ14= (0x01uL << 30), ADC_INTSRC_SEQ15= (0x01uL << 31) }csi_adc_intsrc_e;	
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

12.3.14 csi_adc_set_cmp0

*csi_error_t csi_adc_set_cmp0(csp_adc_t *ptAdcBase, uint8_t byCmpChnl, uint32_t wCmpData, csi_adc_cmp_dir_e eDir)*

12.3.14.1 功能描述

配置ADC采样比较寄存器0比较功能

12.3.14.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

byCmpChnl：要比较的通道号，转换序列中的某一通道，用户选择。

wCmpData：要比较的参考值，即指定的采样通道AD数值要和这个值进行比较。

eDir：比较方向选择，大于/小于比较参考值，枚举详见csi_adc_cmp_dir_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
byCmpChnl	uint8_t 类型数值，被比较的通道，用户选择。	转换序列中通道号，0~15
wCmpData	uint32_t 类型数值，比较的参考值	ADC采样范围中任意数值
eDir	typedef enum { ADC_CMP_H = 0, ADC_CMP_L, }csi_adc_cmp_dir_e;	两种选择：大于wCmpData、小于wCmpData 在adc.h中定义
csi_error_t	csi_error_t中定义值	在common.h中定义

12.3.15 csi_adc_set_cmp1

*csi_error_t csi_adc_set_cmp1(csp_adc_t *ptAdcBase, uint8_t byCmpChnl, uint32_t wCmpData, csi_adc_cmp_dir_e eDir)*

12.3.15.1 功能描述

配置ADC采样比较寄存器1比较功能

12.3.15.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

byCmpChnl：要比较的通道号，转换序列中的某一通道，用户选择。

wCmpData：要比较的参考值，即指定的采样通道AD数值要和这个值进行比较。

eDir：比较方向选择，大于/小于比较参考值(wCmpData)，枚举详见csi_adc_cmp_dir_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
byCmpChnl	uint8_t 类型数值，被比较的通道，用户选择。	转换序列中通道号，0~15
wCmpData	uint32_t 类型数值，比较的参考值	ADC采样范围中任意数值
eDir	typedef enum { ADC_CMP_H = 0, ADC_CMP_L, }csi_adc_cmp_dir_e;	两种选择：大于wCmpData、小于wCmpData 在adc.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.16 csi_adc_get_status

*csi_adc_state_e csi_adc_get_status(csp_adc_t *ptAdcBase)*

12.3.16.1 功能描述

获取ADC转换状态

12.3.16.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

2. 返回值

ADC转换状态，枚举详见csi_adc_state_e。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	

return value	typedef enum{
	ADC_STATE_IDLE = 0, //idle
	ADC_STATE_DOING, //working
	ADC_STATE_DONE //complete
	}csi_adc_state_e;
		三种状态：空闲、工作中、转换完成
		在adc.h中定义

12.3.17 csi_adc_clr_status

*void csi_adc_clr_status(csp_adc_t *ptAdcBase)*

12.3.17.1 功能描述

清除ADC工作状态(设置ADC为空闲状态)

12.3.17.2 参数/功能描述

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	

12.3.18 csi_adc_set_sync

*csi_error_t csi_adc_set_sync(csp_adc_t *ptAdcBase, csi_adc_trgin_e eTrgIn, csi_adc_trgmode_e eTrgMode, uint8_t byDelay)*

12.3.18.1 功能描述

配置ADC外部同步触发输入

12.3.18.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

eTrgin：同步触发输入，枚举定义详见csi_adc_trgin_e。

eTrgMode：同步触发输入模式，枚举定义详见csi_adc_trgmode_e。

byDelay：触发延时，即触发时延时一段时间才开始ADC转换。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
eTrgin	typedef enum{ ADC_TRG_SYNCEN0 = 0, ADC_TRG _SYNCEN1, ADC_TRG _SYNCEN2, ADC_TRG _SYNCEN3, ADC_TRG _SYNCEN4, ADC_TRG _SYNCEN5 }csi_adc_trgin_e;	ADC同步触发输入有6个端口：SYNCIN0~SYNCIN5 在adc.h中定义

eTrgMode	typedef enum{
	ADC_TRG_CONTINU= 0, //continuous trG mode
	ADC_TRG_ONCE //once trgmode
	}csi_adc_trgmode_e;
		两种触发模式：连续触发、一次性触发
		在adc.h中定义

byDelay	uint8_t 类型数值，范围：0~0xff	触发延时启动，即触发时，延时一段时间才开始ADC转换
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.19 csi_adc_rearm_sync

*void csi_adc_rearm_sync(csp_adc_t *ptAdcBase, csi_adc_trgin_e eTrgIn)*

12.3.19.1 功能描述

使能硬件自动REARM

12.3.19.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

eTrgin：同步触发输入，枚举定义详见csi_adc_trgin_e

2. 返回值、

无返回值。

3. 参数说明

参数	说明	枚举定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
eTrgin	typedef enum{ ADC_SYNCEN0 = 0, ADC_SYNCEN1, ADC_SYNCEN2, ADC_SYNCEN3, ADC_SYNCEN4, ADC_SYNCEN5 }csi_adc_trgin_e;	ADC同步触发输入有6个端口： SYNCIN0~SYNCIN5在adc.h中定义

12.3.20 csi_adc_set_evtrg

*csi_error_t csi_adc_set_evtrg(csp_adc_t *ptAdcBase, csi_adc_trgout_e eTrgOut, csi_adc_trgsrc_e eTrgSrc)*

12.3.20.1 功能描述

配置ADC事件触发输出

12.3.20.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

eTrgOut：输出端口选择，共有两个端口(数值：0~1)

eTrgSrc：ADC触发源，枚举定义详见csi_adc_trgsrc_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
eTrgOut	uint8_t 类型数值，有效数值：0~1 typedef enum{ ADC_TRGOUT0 = 0, ADC_TRGOUT1 }csi_adc_trgout_e;	两个触发输出端口(0/1)
eTrgSrc	typedef enum{ ADC_TRGSRC_NONE = 0, ADC_TRGSRC_EOC, ADC_TRGSRC_READY, ADC_TRGSRC_OVR, ADC_TRGSRC_CMP0H, ADC_TRGSRC_CMP0L, ADC_TRGSRC_CMP1H, ADC_TRGSRC_CMP1L,	ADC事件触发源有23个：包含16个序列通道采样结束事件(序列通道号0~15)和其余7个事件 在adc.h中定义。

	ADC_TRGSRC_SEQEND0, ADC_TRGSRC_SEQEND1, ADC_TRGSRC_SEQEND2, ADC_TRGSRC_SEQEND3, ADC_TRGSRC_SEQEND4, ADC_TRGSRC_SEQEND5, ADC_TRGSRC_SEQEND6, ADC_TRGSRC_SEQEND7, ADC_TRGSRC_SEQEND8, ADC_TRGSRC_SEQEND9, ADC_TRGSRC_SEQEND10, ADC_TRGSRC_SEQEND11, ADC_TRGSRC_SEQEND12, ADC_TRGSRC_SEQEND13, ADC_TRGSRC_SEQEND14, ADC_TRGSRC_SEQEND15 }csi_adc_trgsrc_e;	
csi_error_t	csi_error_t 中定义值	在common.h中定义

12.3.21 csi_adc_fvrout_enable

*void csi_adc_fvrout_enable(csp_adc_t *ptAdcBase, csi_adc_fvrsel_e eLvl, bool bEnable)*

12.3.21.1 功能描述

使能ADC FVROUT

12.3.21.2 参数/返回值说明

1. 参数

ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

eLvl：FVR(固定参考电压源)电平选择，枚举详见csi_adc_fvrsel_e。

bEnable：使能/禁止，ENABLE/DISABLE。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	
eLvl	typedef enum{ ADC_FVR2048 = 0, ADC_FVR4096 }csi_adc_fvrsel_e;	电压值有两种选择：2.048V、4.096V 在adc.h中定义
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能 DISABLE：禁止 在common.h中定义

12.3.22 csi_adc_bufout_enable

*void csi_adc_bufout_enable(csp_adc_t *ptAdcBase , csi_adc_bufsel_e eBufSel, bool bEnable)*

12.3.22.1 功能描述

使能ADC BUFFER输出,有两种选择，一个是内部1v电压，另一个是温度传感器的电压

12.3.22.2 参数/返回值说明

1. 参数

   ptAdcBase：ADC寄存器结构体指针，指向ADC基地址，结构体定义详见csp_adc_t。

   eBufSel：内部输出电压选择，详见枚举类型csi_adc_bufsel_e

   bEnable：使能/禁止输出，ENABLE/DISABLE。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptAdcBase	csp_adc_t 类型指针，ADC0；请参阅12.3.1.2参数说明	

eBufSel	typedef enum{
	ADCIN_INTERIOR_1V0 = 2, //interior 1V0
	}csi_adc_bufsel_e;
		ADCIN_INTERIOR_1V0：内部1v电压。

bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能输出 DISABLE：禁止输出 在common.h中定义

13 PINCTRL

13.1 概述

ATP CSI接口PINCTRL的设计中，提供IO丰富的配置及其操作。配置方面包括IO复用功能选择，上拉/下拉，输入/输出模式，速度以及驱动力配置。操作方面提供单个IO翻转，输出高低电平等。

13.2 API列表

Table 2-1 时钟CSI接口函数

API	说明	函数位置
csi_pin_set_mux	设置pin复用功能	pinctrl.c
csi_pin_set_iomap	设置pin的IO重定义	
csi_pin_get_mux	获取pin复用功能	
csi_pin_pull_mode	设置pin上拉/下拉模式	
csi_pin_input_filter	使能/禁止pin输入滤波功能	
csi_pin_output_mode	设置pin输出模式	
csi_pin_input_mode	设置pin输入模式	
csi_pin_speed	设置pin速度	
csi_pin_drive	设置pin驱动能力	
csi_pin_get_num	通过pin name获取pin number	
csi_pin_read	通过pin name读取pin输入电平状态	
csi_pin_irq_mode	设置pin中断模式	
csi_pin_irq_enable	使能pin外部中断	
csi_pin_toggle	翻转pin输出电平状态	
csi_pin_set_high	设置pin输出电平状态为高	
csi_pin_set_low	设置pin输出电平状态为低	
csi_exi_set_evtrg	配置EXI(外部中断)事件触发	
csi_exi_soft_evtrg	EXI(外部中断)事件软件触发	
csi_exi_flt_enable	EXI通道的数字滤波器使能	

13.3 API详细说明

13.3.1 csi_pin_set_mux

void csi_pin_set_mux(pin_name_e ePinName, pin_func_e ePinFunc)

13.3.1.1 功能描述

设置pin复用功能。

13.3.1.2 参数/返回值说明

1. 参数

   ePinName：每一款MCU都定义了对应的pin name，枚举定义详见pin_name_e。

   ePinFunc：每一款MCU对应的pin脚都定义了自己的复用编号，枚举定义详见pin_func_e。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置

ePinName	typedef enum {
	PA00 = 0U, //PA00 to PA011 (12 pin)
	PA01 = 1U,
	PA02 = 2U,
	PA03 = 3U,
	PA04 = 4U,
	PA05 = 5U,
	PA06 = 6U,
	PA07 = 7U,
	PA08 = 8U,
	PA09 = 9U,
	PA010 = 10U,
	PA011 = 11U,
	
	PA10 = 12U, //PA10 to PA15 (6 pin)
	PA11 = 13U,
	PA12 = 14U,
	PA13 = 15U,
	PA14 = 16U,
	PA15 = 17U,
	
	PB00 = 18U, //PB00 to PB07 (8 pin)
	PB01 = 19U,
	PB02 = 20U,
	PB03 = 21U,
	PB04 = 22U,
	PB05 = 23U,
	PB06 = 24U,
	PB07 = 25U,
	
	PC00 = 26U, //PC00 to PC03 (4 pin)
	PC01 = 27U,
	PC02 = 28U,
	PC03 = 29U
	} pin_name_e;
		gpio全部的引脚都定义了自己的pin name，pin name都包含在枚举类型pin_name_e中；
		在soc.h中定义

ePinFunc	PA00_GPD = 0U,
	PA00_INPUT = 1U, //input
	PA00_OUTPUT = 2U, //output
	PA00_OUTPUT_MONI = 3U, //output with monitor
	PA00_EPT_CHBX = 6U,
	PA00_EPT_CHAX = 8U,
	PA00_GROUP0 = 9U,
	PA00_ADC_AIN0 = 10U,
	PA00_CPINN4 = 10U,
		pin脚所有的功能都包含在枚举类型pin_func_e中；这里只列举出PA00所有功能定义，其它的pin脚功能定义类似
		在soc.h中定义

13.3.2 csi_pin_set_iomap

13.3.2.1 功能描述

配置pin的IOMAP功能

csi_error_t csi_pin_set_iomap(pin_name_e ePinName, csi_gpio_iomap_e eIoMap)

13.3.2.2 参数/返回值说明

1. 参数

   ePinName：每一款MCU都定义了对应的pin name，枚举定义详见pin_name_e。

   eIoMap：IOMAP 共有两组 group0/1；没有 group 支持 8 个 pin 脚，每个 pin 脚支持 8 个预设功能，枚举定义详见csi_gpio_iomap_e。

2. 返回值

   CSI_OK：设置成功。

   CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	枚举定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

eIoMap	typedef enum {
	IOMAP0_RSVD0 =0U, //IOMAP GROUP0
	IOMAP0_RSVD1,
	IOMAP0_USART0_TX,
	IOMAP0_USART0_RX,
	IOMAP0_RSVD4,
	IOMAP0_RSVD5,
	IOMAP0_GPT_CHA,
	IOMAP0_GPT_CHB,
	IOMAP1_USART0_TX, //IOMAP GROUP1
	IOMAP1_USART0_RX,
	IOMAP1_USART0_CK,
	IOMAP1_RSVD3,
	IOMAP1_RSVD4,
	IOMAP1_RSVD5,
	IOMAP1_EPT_CHAX,
	IOMAP1_EPT_CHAY
	} csi_gpio_iomap_e;
		系列mcu支持两组IOMAP，每组对应8个预设功能，每组支持8个不同的pin脚(IO口)。
		比如：PA00支持group0，PA00可以配置为IOMAP GROUP0中的8个预设功能中任一个。IOMAP功能详情，在用户手册syscon章节的IO重定义有详细介绍。
		枚举在gpio.h 中定义。

return value	pin_func_e 中枚举值，请参阅13.3.1.2参数说明	

13.3.3 csi_pin_get_mux

pin_func_e csi_pin_get_mux(pin_name_e ePinName)

13.3.3.1 功能描述

获取pin复用功能。

13.3.3.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

2. 返回值

返回pin脚复用功能编号，类型pin_func_e，具体定义详见pin_func_e。

3. 参数/返回值说明

参数/返回值	说明	枚举定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义
return value	pin_func_e 中枚举值，请参阅13.3.1.2参数说明	

13.3.4 csi_pin_pull_mode

csi_error_t csi_pin_pull_mode(pin_name_e ePinName, csi_gpio_pull_mode_e ePullMode)

13.3.4.1 功能描述

设置pin上拉/下拉模式

13.3.4.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

ePullMode：上/下拉模式，枚举定义详见csi_gpio_pull_mode_e。

2. 返回值

CSI_OK：设置成功。

CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

ePullMode	typedef enum {
	GPIO_PULLNONE = 0, //Pull none
	GPIO_PULLUP, //Pull up
	GPIO_PULLDOWN, //Pull down
	} csi_gpio_pull_mode_e;
		三种模式：上拉、下拉和无上下拉；默认无上下拉。
		在gpio.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

13.3.5 csi_pin_input_filter

void csi_pin_input_filter(pin_name_e ePinName, bool bEnable)

13.3.5.1 功能描述

使能/禁止PIN输入滤波功能，输入信号有30ns滤波。

13.3.5.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

bEnable：使能/禁止，ENABLE/DISABLE。

2. 返回值：无。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义
bEnable	bool类型数值，ENBALE/DISABLE	ENBALE：使能 DISABLE：禁止 在common.h中定义

13.3.6 csi_pin_output_mode

csi_error_t csi_pin_output_mode(pin_name_e ePinName, csi_gpio_output_mode_e eOutMode)

13.3.6.1 功能描述

设置pin输出模式

13.3.6.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

eOutMode：输出模式，枚举定义详见csi_gpio_output_mode_e。

2. 返回值

CSI_OK：设置成功。

CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

eOutMode	typedef enum {
	GPIO_PUSH_PULL = 0, //push-pull
	GPIO_OPEN_DRAIN, //open drain
	} csi_gpio_output_mode_e;
		两种模式：推挽输出、开漏输出；默认为推挽输出
		在gpio.h中定义。

csi_error_t	csi_error_t 中定义值	在common.h中定义

13.3.7 csi_pin_input_mode

csi_error_t csi_pin_output_mode(pin_name_e ePinName, csi_gpio_output_mode_e eOutMode)

13.3.7.1 功能描述

设置 pin 输入模式

13.3.7.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

eInputMode：输出模式，枚举定义详见csi_gpio_input_mode_e。

2. 返回值

CSI_OK：设置成功。

CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

eInputMode	typedef enum {
	GPIO_INPUT_CMOS = 0U, //cmos
	GPIO_INPUT_TTL1, //ttl1
	GPIO_INPUT_TTL2 //ttl2
	}csi_gpio_input_mode_e;
		三种模式：推CMOS、TTL1和TTL2；
		在gpio.h中定义。

csi_error_t	csi_error_t 中定义值	在common.h中定义

13.3.8 csi_pin_speed

void csi_pin_speed(pin_name_e ePinName, csi_gpio_speed_e eSpeed)

13.3.8.1 功能描述

设置pin作为输出时的速度模式

13.3.8.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

eSpeed：速度模式，枚举定义详见csi_gpio_speed_e。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其枚举定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

eSpeed	typedef enum {
	GPIO_SPEED_LV0 = 0U, //normal
	GPIO_SPEED_LV1, //fast
	} csi_gpio_speed_e;
		两种模式：慢速(普通)、快速；默认为慢速
		在gpio.h中定义。

13.3.9 csi_pin_drive

csi_error_t csi_pin_drive(pin_name_e ePinName, csi_gpio_drive_e eDrive)

13.3.9.1 功能描述

设置pin作为输出时的驱动能力

13.3.9.2 参数/返回值说明

1. 参数

   ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

   eDrive：驱动能力，枚举定义详见csi_gpio_drive_e。

2. 返回值

   CSI_OK：设置成功。

   CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

eDrive	typedef enum {
	GPIO_DRIVE_LV0 = 0U, //normal
	GPIO_DRIVE_LV1, //strong
	} csi_gpio_drive_e;
		两种模式：弱驱(普通)、强驱模式：默认为弱驱模式
		在gpio.h中定义。

csi_error_t	csi_error_t 中定义值	在common.h中定义

13.3.10 csi_pin_get_num

uint8_t csi_pin_get_num(pin_name_e ePinName)

13.3.10.1 功能描述

通过pin name获取pin number

13.3.10.2 参数/返回值说明

1. 参数

   ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

2. 返回值

   pin number(如：(PA00，返回0)；(PA010，返回10)；(PB01，返回1))。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明表	在soc.h中定义
return value	uint8_t 类型数值，数值范围：0~15	pin number：0~15

13.3.11 csi_pin_read

uint32_t csi_pin_read(pin_name_e ePinName)

13.3.11.1 功能描述

读取pin输入电平状态

13.3.11.2 参数/返回值说明

1. 参数

   ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

2. 返回值

   电平状态：高低(1/0)；返回值格式：(0 或 1 << (pin number))。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举定义位置
ePinName	pin_name_e中枚举值，请参阅13.3.1.2参数说明表	在soc.h中定义
return value	uint32_t 类型数值，数值范围：0~(1 << 15)	电平状态：高低(1/0)

13.3.12 csi_pin_irq_mode

csi_error_t csi_pin_irq_mode(pin_name_e ePinName, csi_exi_grp_e eExiGrp, csi_gpio_irq_mode_e eTrgEdge)

13.3.12.1 功能描述

配置pin中断模式

13.3.12.2 参数/返回值说明

1. 参数

   ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

   eExiGrp：外部中断组，枚举定义详见csi_exi_grp_e。

   eTrgEdge：中断触发边沿模式，枚举定义详见csi_gpio_irq_mode_e。

2. 返回值

   CSI_OK：配置成功

   CSI_ERROR：配置失败

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义
eExiGrp	typedef enum { EXI_GRP0 = 0, EXI_GRP1, EXI_GRP2, EXI_GRP3, EXI_GRP4, EXI_GRP5, EXI_GRP6, EXI_GRP7, EXI_GRP8, EXI_GRP9, EXI_GRP10, EXI_GRP11, EXI_GRP12, EXI_GRP13, EXI_GRP14, EXI_GRP15, EXI_GRP16, EXI_GRP17, EXI_GRP18, EXI_GRP19, }csi_exi_grp_e;	系统有16(0~15)个中断组和4(16~19)个扩展中断组，共20个。配置外部中断时，默认建议用0~15中断组（PA00:GRP0->PA015:GRP15），PB时和PA配置类似。资源不够时（不同端口的相同pin number都需要配置中断，如PA00和PB00都需要中断），再使用扩展中断组 在gpio.h中定义。

eTrgEdge	typedef enum {
	GPIO_IRQ_RISING_EDGE = 0, //rising edge
	GPIO_IRQ_FALLING_EDGE, //falling edge
	GPIO_IRQ_BOTH_EDGE, //both edge
	} csi_gpio_irq_mode_e;
		外部中断触发边沿有三种模式：上升沿、下降沿和双边(上升/下降)沿

csi_error_t	csi_error_t 中定义值	在common.h中定义

13.3.13 csi_pin_irq_enable

csi_error_t csi_pin_irq_enable(pin_name_e ePinName, csi_exi_grp_e eExiGrp, bool bEnable)

13.3.13.1 功能描述

使能pin中断

13.3.13.2 参数/返回值说明

1. 参数

   ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

   eExiGrp：外部中断组，枚举定义详见csi_exi_grp_e。

   bEnable：使能/禁止中断（ENABLE/DISABLE）

2. 返回值

   CSI_OK：成功。

   CSI_ERROR：失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明表	在soc.h中定义
eExiGrp	csi_exi_grp_e 中枚举值	在gpio.h中定义。
bEnable	bool类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

13.3.14 csi_pin_toggle

void csi_pin_toggle(pin_name_e ePinName)

13.3.14.1 功能描述

翻转pin输出电平状态

13.3.14.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

2. 返回值

无返回值。

3. 参数说明

参数	说明	枚举定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

13.3.15 csi_pin_set_high

void csi_pin_set_high(pin_name_e ePinName)

13.3.15.1 功能描述

设置pin输出电平状态为高

13.3.15.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

2. 返回值

无返回值。

3. 参数说明

参数	说明	枚举定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

13.3.16 csi_pin_set_low

void csi_pin_set_low(pin_name_e ePinName)

13.3.16.1 功能描述

设置设置pin输出电平状态为低

13.3.16.2 参数/返回值说明

1. 参数

ePinName：pin脚名字，即pin name，枚举定义详见pin_name_e。

2. 返回值

无返回值。

3. 参数说明

参数	说明	枚举定义位置
ePinName	pin_name_e 中枚举值，请参阅13.3.1.2参数说明	在soc.h中定义

13.3.17 csi_exi_set_evtrg

csi_error_t csi_exi_set_evtrg(csi_exi_trgout_e eTrgOut, csi_exi_trgsrc_e eExiTrgSrc, uint8_t byTrgPrd)

13.3.17.1 功能描述

配置EXI(外部中断)事件触发输出

13.3.17.2 参数/返回值说明

1. 参数

eTrgOut：输出通道选择，枚举定义详见csi_exi_trgout_e。

eExiTrgSrc：EXI触发源，枚举定义详见csi_exi_trgsrc_e。

byTrgPrd：EXI事件触发计数周期

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举定义位置
byTrgOut	typedef enum { EXI_TRGOUT0 = 0, EXI_TRGOUT1, EXI_TRGOUT2, EXI_TRGOUT3, EXI_TRGOUT4, EXI_TRGOUT5, }csi_exi_trgout_e;	共有6个(EXI_TRGOUT0~5)触发输出通道 在gpio.h中定义。
eExiTrgSrc	typedef enum { TRGSRC_EXI0 = 0, TRGSRC_EXI1, TRGSRC_EXI2, TRGSRC_EXI3, TRGSRC_EXI4, TRGSRC_EXI5, TRGSRC_EXI6, TRGSRC_EXI7, TRGSRC_EXI8, TRGSRC_EXI9, TRGSRC_EXI10, TRGSRC_EXI11, TRGSRC_EXI12, TRGSRC_EXI13, TRGSRC_EXI14, TRGSRC_EXI15, TRGSRC_EXI16, TRGSRC_EXI17, TRGSRC_EXI18, TRGSRC_EXI19, }csi_exi_trgsrc_e;	EXI事件触发源共20个(EXI0~EXI19)，和外部中断组相对应。 在gpio.h中定义。
byTrgPrd	uint8_t 类型数值，数值范围：0~15	EXI触发次数大于该设置值时产生触发输出(该值为1是，需要2次触发才会产生一次触发输出)； 该值默认为0
csi_error_t	csi_error_t 中定义值	在common.h中定义

13.3.18 csi_exi_soft_evtrg

void csi_exi_soft_evtrg(csi_exi_trgout_e eTrgOut)

13.3.18.1 功能描述

EXI(外部中断)事件软件触发

13.3.18.2 参数/返回值说明

1. 参数

eTrgOut：输出通道选择，枚举定义详见csi_exi_trgout_e。

2. 返回值：无。

3. 参数说明

参数	说明	概述及其枚举定义位置
byTrgOut	typedef enum { EXI_TRGOUT0 = 0, EXI_TRGOUT1, EXI_TRGOUT2, EXI_TRGOUT3, EXI_TRGOUT4, EXI_TRGOUT5, }csi_exi_trgout_e;	共有6个(EXI_TRGOUT0~5)触发输出通道 在gpio.h中定义。

13.3.19 csi_exi_flt_enable

void csi_exi_flt_enable(csi_exi_flt_ckdiv_e eCkDiv, bool bEnable)

13.3.19.1 功能描述

EXI 通道的数字滤波器使能

13.3.19.2 参数/返回值说明

1. 参数

eCkDiv：数字滤波clk分频，枚举定义详见csi_exi_flt_ckdiv_e。滤波时间和IMO_FSEL选择有关， IMO_FSEL有4种频率值，具体的配置API接口请参阅时钟章节csi_error_t csi_imosc_enable(uint8_t byFre)接口函数。

bEnable：使能/禁止滤波（ENABLE/DISABLE）

2. 返回值：无

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
eCkDiv	数字滤波clk分频	在gpio.h中定义
bEnable	bool类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

14 GPIO PORT

14.1 概述

ATP32F171有四个GPIO端口，GPIOA0(Port A0) 、GPIOA1(Port A1) 、GPIOB0(Port B0)和GPIOC0(Port C0)；CSI接口GPIO的PORT设计中，提供IO输入和输出的相关配置。配置方面包括I/O方向(输入/输出)，输入/输出模式，上拉/下拉和中断等。操作方面提供GIO翻转，输出高低电平等。API接口函数对Port A0/A1/B0/C0进行操作（可以是整个Port或者Port中的某几个引脚）。

14.2 API列表

Table 14-1 GPIO PORT CSI接口函数

API	说明	函数位置
csi_gpio_port_dir	设置gpio port 方向(输入/输出)	gpio.c
csi_gpio_port_pull_mode	设置gpio port 上拉/下拉	
csi_gpio_port_input_filter	使能/禁止gpio port 输入滤波功能	
csi_gpio_port_output_mode	设置gpio port 输出模式	
csi_gpio_port_input_mode	设置gpio port 输入模式	
csi_gpio_port_write	设置gpio port 引脚输出电平状态	
csi_gpio_port_read	读取gpio port 引脚输入电平状态	
csi_gpio_port_irq_mode	设置gpio port 中断模式	
csi_gpio_port_irq_enable	使能gpio port 引脚中断	
csi_gpio_port_toggle	翻转gpio port 引脚输出电平状态	
csi_gpio_port_set_high	设置gpio port 引脚输出电平状态为高	
csi_gpio_port_set_low	设置gpio port 引脚输出电平状态为低	

14.3 API详细说明

14.3.1 csi_gpio_port_dir

*csi_error_t csi_gpio_port_dir(csp_gpio_t *ptGpioBase, uint32_t wPinMask, csi_gpio_dir_e eDir)*

14.3.1.1 功能描述

配置gpio port方向(输入/输出模式)

14.3.1.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/A1/B0/C0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

eDir：方向，枚举定义详见csi_gpio_dir_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptGpioBase	csp_gpio_t 类型指针：GPIOA0/ GPIOA1/GPIOB0/ GPIOC0，指向对应外设基地址	四个Port，定义两个对应的结构体指针GPIOA0/ GPIOA1/GPIOB0/ GPIOC0，分别指向Port A0/A1/B0/C0基地址。 GPIOA0/ GPIOA1/GPIOB0/ GPIOC0在devices.c中定义 csp_gpio_t在csp_gpio.h中定义
wPinMask	uint32_t（unsigned int）类型，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

eDir	`typedef enum {`
	`  GPIO_DIR_GPD = 0,`
	`  GPIO_DIR_INPUT, //GPIO as input`
	`  GPIO_DIR_OUTPUT //GPIO as output`
	`} csi_gpio_dir_e;`
		三种模式：高阻、输入和输出；默认为高阻态
		在gpio.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

14.3.2 csi_gpio_port_pull_mode

*csi_error_t csi_gpio_port_pull_mode(csp_gpio_t *ptGpioBase, uint32_t wPinMask, csi_gpio_pull_mode_e eMode)*

14.3.2.1 功能描述

配置gpio port上拉/下拉模式

14.3.2.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

ePullMode：上/下拉模式，枚举定义详见csi_gpio_pull_mode_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t（unsigned int）类型，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

ePullMode	typedef enum {
	GPIO_PULLNONE = 0, //Pull none
	GPIO_PULLUP, //Pull up
	GPIO_PULLDOWN, //Pull down
	} csi_gpio_pull_mode_e;
		三种模式：上拉、下拉和禁止上下拉；默认禁止上下拉。
		在gpio.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

14.3.3 csi_gpio_port_input_filter

*void csi_gpio_port_input_filter(csp_gpio_t *ptGpioBase, uint32_t wPinMask, bool bEnable)*

14.3.3.1 功能描述

使能/禁止gpio port 输入滤波功能，输入信号有30ns滤波。

14.3.3.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

bEnable：使能/禁止，ENABLE/DISABLE。

2. 返回值：无。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3
bEnable	bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

14.3.4 csi_gpio_port_output_mode

*csi_error_t csi_gpio_port_output_mode(csp_gpio_t *ptGpioBase, uint32_t wPinMask, csi_gpio_output_mode_e eOutMode)*

14.3.4.1 功能描述

设置gpio port输出模式。

14.3.4.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

eOutMode：输出模式，枚举定义详见csi_gpio_output_mode_e。

2. 返回值

CSI_OK：设置成功。

CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

eOutMode	typedef enum {
	GPIO_PUSH_PULL = 0, //push-pull
	GPIO_OPEN_DRAIN, //open drain
	} csi_gpio_output_mode_e;
		两种模式：推挽输出、开漏输出；默认为推挽输出。
		在gpio.h中定义。

csi_error_t	csi_error_t 中定义值	在common.h中定义

14.3.5 csi_gpio_port_input_mode

*csi_error_t csi_gpio_port_input_mode(csp_gpio_t *ptGpioBase, uint32_t wPinMask, csi_gpio_input_mode_e eInputMode)*

14.3.5.1 功能描述

设置 gpio port 输入模式。

14.3.5.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

eInputMode：输出模式，枚举定义详见csi_gpio_input_mode_e。

2. 返回值

CSI_OK：设置成功。

CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

eInputMode	typedef enum {
	GPIO_INPUT_CMOS = 0U, //cmos
	GPIO_INPUT_TTL1, //ttl1
	GPIO_INPUT_TTL2 //ttl2
	}csi_gpio_input_mode_e;
		三种模式：推CMOS、TTL1和TTL2；
		在gpio.h中定义。

csi_error_t	csi_error_t 中定义值	在common.h中定义

14.3.6 csi_gpio_port_write

*void csi_gpio_port_write(csp_gpio_t *ptGpioBase, uint32_t wPinMask, csi_gpio_pin_state_e ePinVal)*

14.3.6.1 功能描述

设置gpio port引脚输出电平状态

14.3.6.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

ePinVal：电平状态，枚举定义详见csi_gpio_pin_state_e

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

ePinVal	typedef enum {
	GPIO_PIN_LOW = 0, //GPIO low level
	GPIO_PIN_HIGH, //GPIO high level
	} csi_gpio_pin_state_e;
		两种状态：高电平、低电平
		在gpio.h中定义。

14.3.7 csi_gpio_port_read

*uint32_t csi_gpio_port_read(csp_gpio_t *ptGpioBase, uint32_t wPinMask)*

14.3.7.1 功能描述

读取gpio port指定引脚输入电平状态

14.3.7.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

2. 返回值

根据指定bit位掩码，得到对应的引脚状态。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3
return value	uint32_t 类型数值，范围：0~0xffff	引脚电平状态：高、低(1/0)

14.3.8 csi_gpio_port_irq_mode

*csi_error_t csi_gpio_port_irq_mode(csp_gpio_t *ptGpioBase, uint32_t wPinMask, csi_gpio_irq_mode_e eTrgEdge)*

14.3.8.1 功能描述

配置gpio port中断模式。

14.3.8.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

eTrgEdge：中断触发边沿模式，枚举定义详见csi_gpio_irq_mode_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

eTrgEdge	typedef enum {
	GPIO_IRQ_RISING_EDGE = 0, //rising edge
	GPIO_IRQ_FALLING_EDGE, //falling edge
	GPIO_IRQ_BOTH_EDGE, //both edge
	} csi_gpio_irq_mode_e;
		中断触发边沿有三种模式：上升沿、下降沿、双边(上升/下降)沿

return value	csi_error_t 中定义值	在common.h中定义

14.3.9 csi_gpio_port_irq_enable

*void csi_gpio_port_irq_enable(csp_gpio_t *ptGpioBase, uint32_t wPinMask, bool bEnable)*

14.3.9.1 功能描述

使能gpio port引脚中断

14.3.9.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

bEnable：使能/禁止中断，ENABLE/DISABLE。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位(pin)。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3
bEnable	bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

14.3.10 csi_gpio_port_toggle

*void csi_gpio_port_toggle(csp_gpio_t *ptGpioBase, uint32_t wPinMask)*

14.3.10.1 功能描述

翻转gpio port引脚电平状态

14.3.10.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

14.3.11 csi_gpio_port_set_high

*void csi_gpio_port_set_high(csp_gpio_t *ptGpioBase, uint32_t wPinMask)*

14.3.11.1 功能描述

设置gpio port引脚输出电平状态为高

14.3.11.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

14.3.12 csi_gpio_port_set_low

*void csi_gpio_port_set_low(csp_gpio_t *ptGpioBase, uint32_t wPinMask)*

14.3.12.1 功能描述

设置gpio port引脚输出电平状态为低

14.3.12.2 参数/返回值说明

1. 参数

ptGpioBase：gpio port寄存器结构体指针，指向具体端口(Port A0/B0)基地址，结构体定义详见csp_gpio_t。

wPinMask：bit位掩码，指定需要设置的bit位，如：0x00ff代表，设置pin0~pin7。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptGpioBase	csp_gpio_t 类型指针，请参阅14.3.1.2参数说明	在devices.c中定义
wPinMask	uint32_t 类型数值，范围：1 ~0xffff	指定需要设置的bit位。 如：0x000f：设置pin0~pin3 如：0x0009：设置pin0和pin3

15 BT

15.1 概述

基本型定时器(Basic Timer)，是一个16位递增计数器。支持自动重载功能。可提供定时、计数和简单PWM波形输出。APT CSI接口提供了BT的定时和PWM输出相关配置和操作。

15.2 API列表

Table 15-1 时钟CSI接口函数

API	说明	函数位置
csi_bt_timer_init	定时功能初始化	pin.c
csi_bt_start	启动BT	
csi_bt_stop	停止BT	
csi_bt_pwm_init	PWM输出功能初始化	
csi_bt_get_remaining_value	获取BT剩余计数值(距离定时中断)	
csi_bt_get_load_value	获取BT的Load寄存器值	
csi_bt_is_running	检测BT是否正在工作	
csi_bt_count_mode	设置BT计数器工作模式	
csi_bt_int_enable	使能BT中断	
csi_bt_pwm_duty_cycle_updata	更新BT的PWM输出占空比	
csi_bt_pwm_updata	更新BT的PWM输出周期和占空比	
csi_bt_prdr_cmp_updata	更新BT的PRDR和CMP寄存器值	
csi_bt_set_sync	配置BT外部同步触发输入	
csi_bt_rearm_sync	设置同步触发模式自动REARM	
csi_bt_set_evtrg	配置BT事件触发输出	
csi_bt_evtrg_soft	BT软件产生一次事件触发输出	

15.3 API详细说明

15.3.1 csi_bt_timer_init

*csi_error_t csi_bt_timer_init(csp_bt_t *ptBtBase, uint32_t wTimeOut)*

15.3.1.1 功能描述

定时功能初始化，默认使用计数器周期结束中断(PEND)

15.3.1.2 参数/返回值说明

1. 参数

   ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

   wTimeOut：计数器溢出时间，即定时时间，单位ms。

2. 返回值

   CSI_OK：初始化成功。

   CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3，指向对应BT的基地址	四个BT，定义两个对应的结构体指针BT0~BT3 分别指向BT0~3对应的基地址。 BT0~BT3在devices.c中定义 csp_bt_t在csp_bt.h中定义
wTimeOut	uint32_t 类型数值，单位：us	定时时间，单位us
csi_error_t	csi_error_t 中定义值	在common.h中定义

15.3.2 csi_bt_start

*void csi_bt_start(csp_bt_t *ptBtBase)*

15.3.2.1 功能描述

启动BT

15.3.2.2 参数/返回值说明

1. 参数

   ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	枚举定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义

15.3.3 csi_bt_stop

*void csi_bt_stop(csp_bt_t *ptBtBase)*

15.3.3.1 功能描述

停止BT

15.3.3.2 参数/返回值说明

1. 参数

   ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	枚举定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2中参数说明	在devices.c中定义

15.3.4 si_bt_pwm_init

*csi_error_t csi_bt_pwm_init(csp_bt_t *ptBtBase, csi_bt_pwm_config_t *ptBtPwmCfg)*

15.3.4.1 功能描述

PWM输出初始化

15.3.4.2 参数/返回值说明

1. 参数

ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

ptBtPwmCfg：PWM输出配置结构体指针，结构体定义详见csi_bt_pwm_config_t。

2. 返回值

CSI_OK：设置成功。

CSI_ERROR：设置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义

ptBtPwmCfg	typedef struct {
	uint8_t byIdleLevel; //PWM idel level
	uint8_t byStartLevel; //PWM start Level
	uint8_t byInt; //PWM interrupt source
	uint8_t byDutyCycle; //PWM duty cycle
	uint32_t wFreq; //PWM frequency
	} csi_bt_pwm_config_t;
		初始化配置中包含PWM输出空闲电平、起始电平、中断源、占空比和频率
		在bt.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

15.3.5 csi_bt_get_remaining_value

*uint32_t csi_bt_get_remaining_value(csp_bt_t *ptBtBase)*

15.3.5.1 功能描述

获取BT定时剩余计数值

15.3.5.2 参数/返回值说明

1. 参数

ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

2. 返回值

定时剩余计数值，若要得到剩余时间需要换算(t = 1/Tclk * 剩余计数值)。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义
return value	uint32_t 类型数值，count的剩余计数值	count的剩余计数值

15.3.6 csi_bt_get_load_value

*uint32_t csi_bt_get_load_value(csp_bt_t *ptBtBase)*

15.3.6.1 功能描述

获取BT的Load寄存器值(PRDR寄存器值)。

15.3.6.2 参数/返回值说明

1. 参数

ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

2. 返回值

计数器加载值，若要得到加载时间(定时时间)需换算(t = 1/Tclk * 加载值)。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义
return value	uint32_t 类型数值，count的加载值	count的加载值

15.3.7 csi_bt_is_running

*bool csi_bt_is_running(csp_bt_t *ptBtBase)*

15.3.7.1 功能描述

检测BT工作状态

15.3.7.2 参数/返回值说明

1. 参数

ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

2. 返回值

ture：正在工作。

false：停止工作。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义
return value	Bool 类型数值，ture/false(1/0)	ture：数值为1(真) false：数值为0(假)

15.3.8 csi_bt_count_mode

*void csi_bt_count_mode(csp_bt_t *ptBtBase, csi_bt_cntmode_e eCntMode)*

15.3.8.1 功能描述

设置BT计数工作模式

15.3.8.2 参数/返回值说明

1. 参数

ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

eCntMode：计数工作模式，枚举定义详见csi_bt_cntmode_e。

2. 返回值

无返回值

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义

eCntMode	typedef enum{
	BT_CNT_ONEPULSE = 0, //one pulse
	BT_CNT_CONTINUOUS //continuous
	}csi_bt_cntmode_e;
		计数工作模式有两种，连续计数模式、单次触发模式
		默认为连续计数模式

15.3.9 csi_bt_int_enable

*void csi_bt_int_enable(csp_bt_t *ptBtBase, csi_bt_intsrc_e eIntSrc, bool bEnable)*

15.3.9.1 功能描述

使能BT中断

15.3.9.2 参数/返回值说明

1. 参数

ptBtBase：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

eIntSrc：BT中断源，枚举定义详见csi_bt_intsrc_e。

bEnable：使能/禁止中断，ENABLE/DISABLE。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义

eIntSrc	typedef enum
	{
	BT_INTSRC_NONE = (0x00ul << 0), //NONE interrupt
	BT_INTSRC_PEND = (0x01ul << 0), //PEND interrupt
	BT_INTSRC_CMP = (0x01ul << 1), //CMP interrupt
	BT_INTSRC_OVF = (0x01ul << 2), //OVF interrupt
	BT_INTSRC_EVTRG = (0x01ul << 3) //EVTRG interrupt
	}csi_bt_intsrc_e;
		有4个中断源：周期结束、比较匹配、计数溢出和触发输出事件
		在bt.h中定义

bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

15.3.10 csi_bt_pwm_duty_cycle_updata

*void csi_bt_pwm_duty_cycle_updata(csp_bt_t *ptBtBase, uint8_t byDutyCycle)*

15.3.10.1 功能描述

更新PWM输出占空比

15.3.10.2 参数/返回值说明

1. 参数

   ptBtBase ：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

   byDutyCycle：PWM输出占空比(0 < byDutyCycle < 100)。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义
byDutyCycle	uint8_t 类型数值，范围：(>0 && < 100)	PWM占空比，(>0 && < 100)

15.3.11 csi_bt_pwm_updata

*void csi_bt_pwm_updata(csp_bt_t *ptBtBase, uint32_t wFreq, uint8_t byDutyCycle)*

15.3.11.1 功能描述

更新PWM输出频率和占空比

15.3.11.2 参数/返回值说明

1. 参数

   ptBtBase ：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

   wFreq：PWM输出频率，单位Hz

   byDutyCycle：PWM输出占空比(0 < byDutyCycle < 100)。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义
wFreq	uint32_t 类型数值，单位Hz	PWM频率，单位：Hz
byDutyCycle	uint8_t 类型数值，范围：(>0 && < 100)	PWM占空比，(>0 && < 100)

15.3.12 csi_bt_prdr_cmp_updata

*void csi_bt_prdr_cmp_updata(csp_bt_t *ptBtBase, uint16_t hwPrdr, uint16_t hwCmp)*

15.3.12.1 功能描述

更新PRDR和CMP寄存器值

15.3.12.2 参数/返回值说明

1. 参数

ptBtBase ：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

hwPrdr：加载到PRDR寄存器值，即周期结束值。

hwCmp：加载到CMP寄存器值，即比较匹配值。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义
hwPrdr	uint16_t 类型数值，范围：0~0xffff	PRDR加载值，周期结束寄存器
hwCmp	uint16_t 类型数值，范围：0~0xffff	CMP加载值，比较匹配寄存器

15.3.13 csi_bt_set_sync

*void csi_bt_set_sync(csp_bt_t *ptBtBase,csi_bt_trgin_e eTrgin, csi_bt_trgmode_e eTrgMode, bool bAutoRearm)*

15.3.13.1 功能描述

配置BT外部同步触发输入

15.3.13.2 参数/返回值说明

1. 参数

ptBtBase ：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

eTrgin：同步触发输入，枚举定义详见csi_bt_trgin_e。

eTrgMode：同步触发输入模式，枚举定义详见csi_bt_trgmode_e。

bAutoRearm：自动REARM禁止，禁止/允许后续触发。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义
eTrgin	typedef enum{ BT_TRG_SYNCIN0 = 0, BT_TRG_SYNCIN1, BT_TRG_SYNCIN2 }csi_bt_trgin_e;	BT同步触发输入有两个端口： SYNCIN0触发BT的启动， SYNCIN1触发BT的计数值增加一拍 SYNCIN2触发BT的停止 在bt.h中定义

eTrgMode	typedef enum{
	BT_TRG_CONTINU = 0, //continuous trg mode
	BT_TRG_ONCE //once trg mode
	}csi_bt_trgmode_e;
		两种触发模式：连续触发、一次性触发
		在bt.h中定义

bAutoRearm	bool类型数值，ENABLE/DISABLE	一次性触发模式下有效 ENABLE：清除当初通道状态，并允许新的触发 DISABLE：无效 在common.h中定义

15.3.14 csi_bt_rearm_sync

*void csi_bt_rearm_sync(csp_bt_t *ptBtBase,csi_bt_trgin_e eTrgin)*

15.3.14.1 功能描述

使能硬件自动REARM；计数周期结束时，自动REARM

15.3.14.2 参数/返回值说明

1. 参数

ptBtBase ：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

eTrgin：同步触发输入，枚举定义详见csi_bt_trgin_e

2. 返回值

无返回值。

3. 参数说明

参数	说明	枚举定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义

eTrgin	typedef enum{
	BT_TRGIN_SYNCEN0 = 0, //sync evtrr input0
	BT_TRGIN_SYNCEN1 //sync evtrg input1
	}csi_bt_trgin_e;
		BT同步触发输入有两个端口： SYNCIN0触发BT的启动，SYNCIN1触发BT的计数值增加一拍
		在bt.h中定义

15.3.15 csi_bt_set_evtrg

*csi_error_t csi_bt_set_evtrg(csp_bt_t *ptBtBase, csi_bt_trgout_e eTrgOut, csi_bt_trgsrc_e eTrgSrc)*

15.3.15.1 功能描述

配置BT事件触发输出

15.3.15.2 参数/返回值说明

1. 参数

ptBtBase ：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

eTrgOut：输出通道选择，枚举定义详见csi_bt_trgout_e。

eTrgSrc：BT触发源，枚举定义详见csi_bt_trgsrc_e。

2. 返回值

CSI_OK：配置成功。

CSI_ERROR：配置失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义

eTrgOut	typedef enum { BT_TRGOUT = 0, }csi_bt_trgout_e;	只有一个触发输出通道 BT_TRGOUT

eExiTrgSrc	typedef enum
	{
	BT_TRGSRC_DIS = 0, //none trigger
	BT_TRGSRC_PEND, //PEND as trigger event
	BT_TRGSRC_CMP, //CMP as trigger event
	BT_TRGSRC_OVF //CMP as trigger event
	}csi_bt_trgsrc_e;
		BT事件触发源有三个：
		PEND，周期结束
		CMP，比较匹配、
		OVF，计数器计数溢出
		在bt.h中定义。

csi_error_t	csi_error_t 中定义值	在common.h中定义

15.3.16 csi_bt_evtrg_soft

*void csi_bt_evtrg_soft(csp_bt_t *ptBtBase)*

15.3.16.1 功能描述

软件产生一次事件触发输出

15.3.16.2 参数/返回值说明

1. 参数

   ptBtBase ：BT寄存器结构体指针，指向BT基地址，结构体定义详见csp_bt_t。

2. 返回值：无

3. 参数/返回值说明

参数	说明	概述及其枚举/结构体定义位置
ptBtBase	csp_bt_t 类型指针，BT0~BT3；请参阅15.3.1.2参数说明	在devices.c中定义

16 OPA

16.1 概述

为了方便客户使用，加快客户上手速度，ATP CSI接口OPA的设计中，提供简单方便的配置及其操作。配置方面包括OPA的工作模式配置。操作方面OPA的启动和停止。

16.2 API列表

Table 16-1 OPA CSI 接口函数

API	说明	函数位置
csi_opa_start	启动OPA模块	opa.c
csi_opa_stop	停止OPA模块	
csi_opa_init	OPA模块初始化	

16.3 API详细说明

16.3.1 csi_opa_start

*void csi_opa_start(csp_opa_t *ptOpaBase)*

16.3.1.1 功能描述

启动OPA模块。

16.3.1.2 参数/返回值说明

1. 参数

* ptOpaBase：OPA 寄存器结构体地址。

2. 返回值

无返回值。

3. 结构体/枚举说明表

结构体	结构体/枚举说明	定义位置

csp_opa_t	`typedef struct`
	`{`
	`  __IOM uint32_t  CR;  //OPA 控制寄存器`
	`} csp_opa_t;`
		csp_opa.h

16.3.2 csi_opa_stop

*void csi_opa_stop(csp_opa_t *ptOpaBase)*

16.3.2.1 功能描述

停止OPA模块。

16.3.2.2 参数/返回值说明

1. 参数

* ptOpaBase：opa 寄存器结构体地址。

2. 返回值

无返回值。

3. 结构体/枚举说明表

参数/返回值	说明	定义位置
ptOpaBase	请参阅16.3.1.2中参数说明	csp_opa.h

16.3.3 csi_opa_init

*csi_error_t csi_opa_init(csp_opa_t *ptOpaBase, csi_opa_config_t *ptOpaCfg)*

16.3.3.1 功能描述

OPA从机初始化配置函数

16.3.3.2 参数/返回值说明

1. 参数

* ptOpaBase：OPA 寄存器结构体地址。

* ptOpaCfg：OPA配置结构体指针。

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败

3. 结构体/枚举说明表

结构体/枚举	说明	定义位置
ptOpaBase	请参阅16.3.1.2中参数说明	csp_opa.h

csi_opa_config_t	`typedef struct {`
	`  uint8_t  byMode;  //模式配置`
	`  uint8_t  byInternalGain;  //内部增益倍数`
	`  uint8_t  byOpaNegativeInput;  //Opa 负端输入使能`
	`} csi_opa_config_t;`
		opa.h

csi_error_t	csi_error_t中定义值	common.h

17 GPTA

17.1 概述

通用定时器（General Purpose Timer）作为MCU的关键外设，可以提供多种时基计数和波形产生功能。通过灵活的PWM输出，可以适用于各种复杂多变的应用。GPTA内部包含一个16位的定时/计数模块，支持2种工作模式(捕捉模式和波形发生器模式)。

17.2 API列表

Table 16-1 GPTA CSI接口函数

API	说明	函数位置
gpta0_initen_irqhandler	定时器0中断回调函数	
csi_gpta_capture_init	定时器捕获模式参数设置	
csi_gpta_wave_init	定时器波形模式参数基础设置（计数模式，周期，占空比等）	
csi_gpta_channel_cmpload_config	定时器CMPA.CMPB载入时机设置	
csi_gpta_channel_config	设置通道PWM1、PWM2的波形	
csi_gpta_channel_aqload_config	通道PWM1、PWM2的波形载入时机设置	
csi_gpta_global_config	全局载入控制配置	
csi_gpta_gldcfg	全局载入对象使能或禁止	
csi_gpta_global_sw	软件产生一次全局载入	
csi_gpta_global_rearm	单次模式下全局载入重使能	
csi_gpta_start	定时器开始计数	
csi_gpta_stop	定时器停止计数	
csi_gpta_set_start_mode	定时器开始位模式选择（普通开始/开始的同时触发同步事件0）	
csi_gpta_set_os_mode	定时器模式选择（单次模式或连续模式）	
csi_gpta_set_stop_st	波形输出被停止时，输出端口的缺省状态	
csi_gpta_get_prdr	读取定时器周期寄存器	
csi_gpta_change_ch_duty	改变比较值寄存器	
csi_gpta_debug_enable	调试使能控制	
csi_gpta_evtrg_enable	使能或禁止事件	
csi_gpta_onetimesoftware_output	一次性软件强制输出控制	
csi_gpta_loading_method_aqcsf	AQCSF寄存器载入时机设置	

csi_gpta_continuous_software_waveform	连续软件强制输出控制	
csi_gpta_int_enable	中断使能控制器	
csi_gpta_set_sync	同步事件配置	
csi_gpta_set_extsync_chnl	同步事件用于事件触发输出	
csi_gpta_set_sync_filter	同步事件滤波器控制	
csi_gpta_rearm_sync	单次模式下同步事件重使能	
csi_gpta_set_evtrg	事件输出功能配置	
csi_gpta_set_evcntinit	事件输出计数功能配置	
csi_gpta_reglk_config	链接寄存功能设置	

17.3 API详细说明

17.3.1 gpta0_irqhandler_pro

*__attribute__((weak)) void gpta0_irqhandler_pro(csp_gpta_t *ptGptaBase)*

17.3.1.1 功能描述

定时器0中断回调函数。

17.3.1.2 参数/返回值说明

1. 参数

   ptGptaBase： 每个功能模块都有对应的ptGptaBase，枚举定义详见csp_gpta_t

2. 返回值

   无

3. 参数说明表

参数	说明	概述及其变量位置
ptGptaBase	csp_gpta_t结构体，用于进行寄存器操作。	定时器模块所有寄存器的定义 在csp_gpta.h中定义

17.3.2 略

17.3.3 gpta_capture_init

*csi_error_t csi_gpta_capture_init(csp_gpta_t *ptGptaBase, csi_gpta_captureconfig_t *ptGptaPwmCfg)*

17.3.3.1 功能描述

定时器工作在捕获模式下的参数定义。

17.3.3.2 参数/返回值说明

1. 参数

   ptGptaBase: 定义详见csp_gpta_t

   ptGptaPwmCfg: 定义详见csi_gpta_captureconfig_t

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	枚举变量位置
ptGptaBase	csp_gpta_t中定义	在csp_gpta.h中定义

ptGptaPwmCfg	typedef struct csi_gpta_captureconfig csi_gpta_captureconfig_t;
	struct csi_gpta_captureconfig {
	uint8_t byWorkmod; //Count or capture
	uint8_t byCountingMode; //csi_gpta_cntmd_e
	uint8_t byOneshotMode; //Single or continuous
	uint8_t byStartSrc ;
	uint8_t byPscld;
	uint8_t byDutyCycle; //TIMER PWM OUTPUT duty cycle
	uint8_t byCaptureCapLden;
	uint8_t byCaptureRearm;
	uint8_t byCaptureCapmd;
	uint8_t byCaptureStopWrap;
	uint8_t byCaptureLdaret;
	uint8_t byCaptureLdbret;
	// uint8_t byCaptureLdcret;
	// uint8_t byCaptureLddret;
	uint32_t wInt;
	uint8_t byBurst;
	uint8_t byCgsrc;
	uint8_t byCgflt;
	};
		在gpta.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.4 csi_gpta_wave_init

*csi_error_t csi_gpta_wave_init(csp_gpta_t *ptGptaBase, csi_gpta_pwmconfig_t *ptGptaPwmCfg)*

17.3.4.1 功能描述

定时器工作在波形模式下的定义

17.3.4.2 参数/返回值说明

1. 参数

   ptGptaBase：定义详见csp_gpta_t

   ptGptaPwmCfg：定义详见csi_gpta_pwmconfig_t。

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

ptGptaPwmCfg	typedef struct csi_gpta_pwmconfig csi_gpta_pwmconfig_t;
	struct csi_gpta_pwmconfig {
	uint8_t byWorkmod; //Count or capture
	uint8_t byCountingMode; //csi_gpta_cntmd_e
	uint8_t byOneshotMode; //Single or continuous
	uint8_t byStartSrc ;
	uint8_t byPscld;
	uint8_t byDutyCycle; //TIMER PWM OUTPUT duty cycle
	uint32_t wFreq; //TIMER PWM OUTPUT frequency
	uint32_t wInt;
	uint8_t byBurst;
	uint8_t byCgsrc;
	uint8_t byCgflt;
	uint8_t byCks;
	};
		在gpta.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.5 csi_gpta_channel_cmpload_config

*csi_error_t csi_gpta_channel_cmpload_config(csp_gpta_t *ptGptaBase, csp_gpta_cmpdata_ldmd_e tld, csp_gpta_ldamd_e tldamd ,csi_gpta_camp_e channel)*

17.3.5.1 功能描述

定时器CMPA.CMPB载入时机设置

17.3.5.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   tld： 定义详见 csp_gpta_cmpdata_ldmd_e

   tldamd： 定义详见 csp_gpta_ldamd_e

channel： 定义详见 csi_gpta_camp_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
tld	typedef enum { GPTA_CMPLD_SHDW = 0, GPTA_CMPLD_IMM }csp_gpta_cmpdata_ldmd_e;	在csp_gpta.h中定义
tldamd	typedef enum{ GPTA_LDCMP_NEVER = 0, GPTA_LDCMP_ZRO, GPTA_LDCMP_PRD, GPTA_LDCMP_LD_SYNC = 4, }csp_gpta_ldamd_e;	在csp_gpta.h中定义

channel	typedef enum{
	GPTA_CAMPA=1,
	GPTA_CAMPB,
	// GPTA_CAMPC,
	// GPTA_CAMPD
	}csi_gpta_camp_e;
		在gpta.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.6 csi_gpta_channel_config

*csi_error_t csi_gpta_channel_config(csp_gpta_t *ptGptaBase, csi_gpta_pwmchannel_config_t *tPwmCfg, csi_gpta_channel_e channel)*

17.3.6.1 功能描述

设置通道参数

17.3.6.2 参数/返回值说明

1. 参数

ptGptaBase：定义详见csp_gpta_t

tPwmCfg：定义详见 csi_gpta_pwmchannel_config_t

channel：定义详见csi_gpta_channel_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

tPwmCfg	typedef struct csi_gpta_pwmchannel_config csi_gpta_pwmchannel_config_t;
	struct csi_gpta_pwmchannel_config {
	uint8_t byActionZro; //
	uint8_t byActionPrd; //
	uint8_t byActionCau; //
	uint8_t byActionCad; //
	uint8_t byActionCbu; //
	uint8_t byActionCbd; //
	uint8_t byActionT1u; //
	uint8_t byActionT1d; //
	uint8_t byActionT2u; //
	uint8_t byActionT2d; //
	uint8_t byChoiceCasel;
	uint8_t byChoiceCbsel;
	};
		在gpta.h中定义

channel	typedef enum{ GPTA_CHANNEL_1=1, GPTA_CHANNEL_2 }csi_gpta_channel_e;	在gpta.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.7 csi_gpta_channel_aqload_config

*csi_error_t csi_gpta_channel_aqload_config(csp_gpta_t *ptGptaBase, csp_gpta_ld_e tld, csp_gpta_ldamd_e tldamd ,csi_gpta_channel_e channel)*

17.3.7.1 功能描述

通道PWM1、PWM2的波形载入时机设置

17.3.7.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见csp_gpta_t

tld： 定义详见csp_gpta_ld_e

tldamd： 定义详见csp_gpta_ldamd_e

channel： 定义详见csi_gpta_channel_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义
tld	typedef enum{ GPTA_LD_IMM = 0, GPTA_LD_SHDW }csp_gpta_ld_e;	在csp_gpta.h中定义
tldamd	typedef enum{ GPTA_LDCMP_NEVER = 0, GPTA_LDCMP_ZRO, GPTA_LDCMP_PRD, GPTA_LDCMP_LD_SYNC = 4, }csp_gpta_ldamd_e;	
channel	typedef enum{ GPTA_CHANNEL_1=1, GPTA_CHANNEL_2 }csi_gpta_channel_e;	在gpta.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.8 csi_gpta_global_config

*csi_error_t csi_gpta_global_config(csp_gpta_t *ptGptaBase,csi_gpta_Global_load_control_config_t *Global)*

17.3.8.1 功能描述

设置全局载入控制器参数

17.3.8.2 参数/返回值说明

1. 参数

ptGptaBase：定义详见csp_gpta_t

Global：定义详见 csi_gpta_Global_load_control_config_t

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

Global	typedef struct csi_gpta_Global_load_control_config csi_gpta_Global_load_control_config_t;
	struct csi_gpta_Global_load_control_config{
	bool bGlden;
	bool bOstmd;
	uint8_t bGldprd;
	uint8_t byGldmd;
	};
		在gpta.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.9 csi_gpta_gldcfg

*csi_error_t csi_gpta_gldcfg(csp_gpta_t *ptGptaBase ,csi_gpta_Global_load_gldcfg Glo,bool bENABLE)*

17.3.9.1 功能描述

全局载入对象使能或禁止

17.3.9.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见 csp_gpta_t

Glo： 定义详见 csi_gpta_Global_load_gldcfg

bENABLE: ENABLE/DISABLE

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义
Global	typedef enum { byprdr_A=0, bycmpa_A, bycmpb_A, byaqcra_A=8, byaqcrb_A, byaqcsf_A=12, }csi_gpta_Global_load_gldcfg;	
csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.10 csi_gpta_global_sw

*csi_error_t csi_gpta_global_sw(csp_gpta_t *ptGptaBase)*

17.3.10.1 功能描述

软件触发全局载入

17.3.10.2 参数/返回值说明

1. 参数

   ptGptaBase：定义详见csp_gpta_t

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.11 csi_gpta_global_rearm

*csi_error_t csi_gpta_global_rearm(csp_gpta_t *ptGptaBase)*

17.3.11.1 功能描述

单次模式下全局载入重使能

17.3.11.2 参数/返回值说明

1. 参数

   ptGptaBase：定义详见csp_gpta_t

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.12 csi_gpta_start

*csi_error_t csi_gpta_start(csp_gpta_t *ptgptaBase)*

17.3.12.1 功能描述

定时器开始计数

17.3.12.2 参数/返回值说明

1. 参数

   ptGptaBase：定义详见csp_gpta_t

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

17.3.13 csi_gpta_swstop

*void csi_gpta_stop(csp_gpta_t *ptgptaBase)*

17.3.13.1 功能描述

定时器停止计数

17.3.13.2 参数/返回值说明

1. 参数

   ptGptaBase：定义详见csp_gpta_t

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

17.3.14 csi_gpta_set_start_mode

*void csi_gpta_set_start_mode(csp_gpta_t *ptgptaBase, csi_gpta_stmd_e eMode)*

17.3.14.1 功能描述

定时器开始位模式选择（普通开始/开始的同时触发同步事件0）

17.3.14.2 参数/返回值说明

1. 参数

   ptGptaBase：定义详见csp_gpta_t

   eMode： 定义详见 csi_gpta_stmd_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

eMode	`typedef enum {`
	`  GPTA_SW = 0,`
	`  GPTA_SYNC`
	`}csi_gpta_stmd_e;`
		在gpta.h中定义

17.3.15 csi_gpta_set_os_mode

*void csi_gpta_set_os_mode(csp_gpta_t *ptgptaBase, csi_gpta_opmd_e eMode)*

17.3.15.1 功能描述

定时器工作模式设定（单次或连续）

17.3.15.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   eMode： 定义详见 csi_gpta_opmd_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

eMode	`typedef enum{`
	`  GPTA_OP_CONT = 0,`
	`  GPTA_OP_OT,`
	`}csi_gpta_opmd_e;`
		在gpta.h中定义

17.3.16 csi_gpta_set_stop_st

*void csi_gpta_set_stop_st(csp_gpta_t *ptgptaBase, csp_gpta_stpst_e eSt)*

17.3.16.1 功能描述

波形输出被停止时，输出端口的缺省状态

17.3.16.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   eSt： 定义详见 csp_gpta_stpst_e

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
eSt	typedef enum { GPTA_STPST_HZ = 0, GPTA_STPST_LOW } csp_gpta_stpst_e;	在gpta.h中定义

17.3.17 csi_gpta_get_prdr

*uint16_t csi_gpta_get_prdr(csp_gpta_t *ptgptaBase)*

17.3.17.1 功能描述

获得周期寄存器值

17.3.17.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见 csp_gpta_t

2. 返回值

返回周期寄存器值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
return	返回周期寄存器值	

17.3.18 csi_gpta_change_ch_duty

*csi_error_t csi_gpta_change_ch_duty(csp_gpta_t *ptGptaBase, csi_gpta_chtype_e eCh, uint32_t wActiveTime)*

17.3.18.1 功能描述

改变比较值寄存器

17.3.18.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   eCh： 定义详见 csi_gpta_chtype_e

   wActiveTime：周期寄存器的x%

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

eCh	typedef enum{
	GPTA_CAMPA=1,
	GPTA_CAMPB,
	// GPTA_CAMPC,
	// GPTA_CAMPD
	}csi_gpta_camp_e;
		在gpta.h中定义

wActiveTime	周期寄存器的x%	

17.3.19 csi_gpta_debug_enable

*void csi_gpta_debug_enable(csp_gpta_t *ptGptaBase, bool bEnable)*

17.3.19.1 功能描述

调试使能控制

17.3.19.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   bEnable： 使能或禁止

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
bEnable	ENABLE/DISABLE	在common.h中定义

17.3.20 csi_gpta_evtrg_enable

*csi_error_t csi_gpta_evtrg_enable(csp_gpta_t *ptGptaBase, csi_gpta_trgout_e byCh, bool bEnable)*

17.3.20.1 功能描述

事件输出使能或禁止

17.3.20.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   byCh： 定义详见csi_gpta_trgout_e

   bEnable： ENABLE/DISABLE

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

byCh	*/
	typedef enum{
	GPTA_TRG_OUT0 = 0, //trigger out0
	GPTA_TRG_OUT1, //trigger out1
	// GPTA_TRG_OUT2, //trigger out2
	// GPTA_TRG_OUT3 //trigger out3
	}csi_gpta_trgout_e;
		在gpta.h中定义

bEnable	ENABLE/DISABLE	在common.h中定义
csi_error_t	csi_error_t中定义	在common.h中定义

17.3.21 csi_gpta_onetimesoftware_output

*csi_error_t csi_gpta_onetimesoftware_output(csp_gpta_t *ptGptaBase, uint16_t byCh,csp_gpta_action_e bEnable)*

17.3.21.1 功能描述

一次性软件强制输出控制

17.3.21.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   byCh： GPTA_OSTSFA/GPTA_OSTSFB

   bEnable： 定义详见csp_gpta_action_e

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
byCh	///AQOSF #define GPTA_OSTSFA_POS (0) #define GPTA_OSTSFB_POS (4) #define GPTA_OSTSFA (1) #define GPTA_ACTA_POS (1) #define GPTA_ACTA_MSK (0x3 << GPTA_ACTA_POS) #define GPTA_OSTSFB (0x1 << 4) #define GPTA_ACTB_POS (5) #define GPTA_ACTB_MSK (0x3 << GPTA_ACTB_POS) #define GPTA_AQCSF_LDTIME_POS (16) #define GPTA_AQCSF_LDTIME_MSK (0x3 << GPTA_AQCSF_LDTIME_POS)	在csp_gpta.h中定义
bEnable	typedef enum { GPTA_NA = 0, GPTA_LO, GPTA_HI, GPTA_TG } csp_gpta_action_e;	在csp_gpta.h中定义
csi_error_t	csi_error_t中定义	在common.h中定义

17.3.22 csi_gpta_loading_method_aqcsf

*void csi_gpta_loading_method_aqcsf(csp_gpta_t *ptGptaBase, csp_gpta_aqosf_e bEnable)*

17.3.22.1 功能描述

AQCSF寄存器从Shadow载入到Active的控制

17.3.22.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   csp_gpta_aqosf_e：定义详见csp_gpta_aqosf_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
csp_gpta_aqosf_e	typedef enum { GPTA_AQCSF_NOW=0, GPTA_AQCSF_ZRO, GPTA_AQCSF_PRD, GPTA_AQCSF_ZRO_PRD }csp_gpta_aqosf_e;	在gpta.h中定义

17.3.23 csi_gpta_continuous_software_waveform

*csi_error_t csi_gpta_continuous_software_waveform(csp_gpta_t *ptGptaBase, csi_gpta_channel_e byCh, csp_gpta_aqcsf_e bEnable)*

17.3.23.1 功能描述

连续软件强制输出控制

17.3.23.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   byCh： 定义详见csi_gpta_channel_e

   bEnable 定义详见csp_gpta_aqcsf_e

2. 返回值

   CSI_OK：设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义
csi_gpta_channel_e	typedef enum{ GPTA_CHANNEL_1=1, GPTA_CHANNEL_2 }csi_gpta_channel_e;	在gpta.h中定义
csp_gpta_aqcsf_e	typedef enum { GPTA_AQCSF_NONE=0, GPTA_AQCSF_L, GPTA_AQCSF_H, GPTA_AQCSF_NONE1 }csp_gpta_aqcsf_e;	在gpta.h中定义
csi_error_t	csi_error_t中定义	在common.h中定义

17.3.24 csi_gpta_int_enable

*csi_error_t csi_gpta_int_enable(csp_gpta_t *ptGptaBase, csp_gpta_int_e eInt, bool bEnable)*

17.3.24.1 功能描述

中断使能

17.3.24.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见 csp_gpta_t

eInt： 定义详见csp_gpta_int_e

bEnable ENABLE/DISABLE

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置

ptGptaBase	csp_gpta_t中说明	在csp_gpta.h中定义

csp_gpta_int_e	typedef enum {
	GPTA_INT_TRGEV0 = 0x1,
	GPTA_INT_TRGEV1 = 0x2,
	// GPTA_INT_TRGEV2 = 0x4,
	// GPTA_INT_TRGEV3 = 0x8,
	GPTA_INT_CAPLD0 = 0x1 << 4,
	GPTA_INT_CAPLD1 = 0x1 << 5,
	// GPTA_INT_CAPLD2 = 0x1 << 6,
	// GPTA_INT_CAPLD3 = 0x1 << 7,
	GPTA_INT_CAU = 0x1 << 8,
	GPTA_INT_CAD = 0x1 << 9,
	GPTA_INT_CBU = 0x1 << 10,
	GPTA_INT_CBD = 0x1 << 11,
	GPTA_INT_PEND = 0x1 << 16
	}csp_gpta_int_e;	

bEnable	ENABLE/DISABLE	在common.h中定义
csi_error_t	csi_error_t中定义	在common.h中定义

17.3.25 csi_gpta_set_sync

*void csi_gpta_set_sync(csp_gpta_t *ptGptaBase, csi_gpta_trgin_e eTrgIn, csi_gpta_trgmode_e eTrgMode, csi_gpta_arearm_e eAutoRearm)*

17.3.25.1 功能描述

同步事件配置

17.3.25.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见 csp_gpta_t

eTrgIn： 定义详见csi_gpta_trgin_e

eTrgMode： 定义详见csi_gpta_trgmode_e

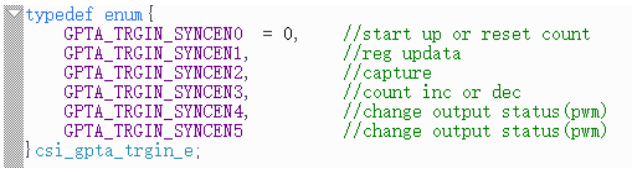
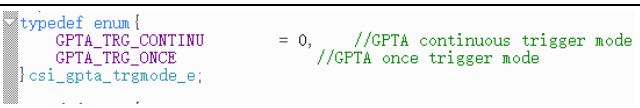
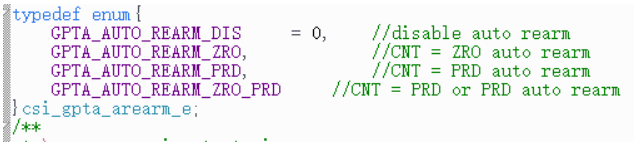
eAutoRearm： 定义详见csi_gpta_arearm_e

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义

eTrgIn		
eTrgMode		
eAutoRearm		

17.3.26 csi_gpta_set_extsync_chnl

*csi_error_t csi_ept_set_sync2evtrg (csp_gpta_t *ptGptaBase, csi_gpta_trgin_e eTrgIn, uint8_t byTrgChx)*

17.3.26.1 功能描述

事件输出选择（同步事件中选择）

17.3.26.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   eTrgIn： 定义详见csi_gpta_trgin_e

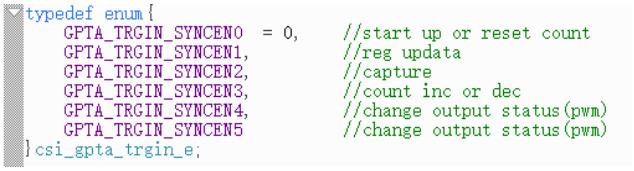
   byTrgChx： 0/1

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义
eTrgIn		
byTrgChx	0/1	

csi_error_t	csi_error_t中定义	在common.h中定义

17.3.27 csi_gpta_set_sync_filter

*csi_error_t csi_gpta_set_sync_filter(csp_gpta_t *ptGptaBase, csi_gpta_filter_config_t *ptFilter)*

17.3.27.1 功能描述

同步事件计数器配置

17.3.27.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见 csp_gpta_t

ptFilter： 定义详见csi_gpta_filter_config_t

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义

ptFilter	typedef struct {
	uint8_t byFiltSrc; //filter input signal source
	uint8_t byWinInv; //window inversion
	uint8_t byWinAlign; //window alignment
	uint8_t byWinCross; //window cross
	uint16_t byWinOffset; //window offset
	uint16_t byWinWidth; //window width
	} csi_gpta_filter_config_t;	

csi_error_t	csi_error_t中定义	在common.h中定义

17.3.28 csi_gpta_rearm_sync

*void csi_gpta_rearm_sync(csp_gpta_t *ptGptaBase,csi_gpta_trgin_e eTrgin)*

17.3.28.1 功能描述

单次模式下同步事件重使能

17.3.28.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   eTrgin： 定义详见csi_gpta_trgin_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义

eTrgin	typedef enum{
	GPTA_TRGIN_SYNCEN0 = 0, //start up or reset count
	GPTA_TRGIN_SYNCEN1, //reg updata
	GPTA_TRGIN_SYNCEN2, //capture
	GPTA_TRGIN_SYNCEN3, //count inc or dec
	GPTA_TRGIN_SYNCEN4, //change output status(pwm)
	GPTA_TRGIN_SYNCEN5 //change output status(pwm)
	}csi_gpta_trgin_e;	

17.3.29 csi_gpta_set_evtrg

*csi_error_t csi_gpta_set_evtrg(csp_gpta_t *ptGptaBase, csi_gpta_trgout_e byTrgOut, csp_gpta_trgsrc0_e eTrgSrc)*

17.3.29.1 功能描述

事件输出功能配置

17.3.29.2 参数/返回值说明

1. 参数

   ptGptaBase： 定义详见 csp_gpta_t

   byTrgOut： 定义详见csi_gpta_trgout_e

   eTrgSrc： 定义详见csp_gpta_trgsrc0_e

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义

byTrgOut	typedef enum{
	GPTA_TRG_OUT0 = 0, //trigger out0
	GPTA_TRG_OUT1, //trigger out1
	// GPTA_TRG_OUT2, //trigger out2
	// GPTA_TRG_OUT3 //trigger out3
	}csi_gpta_trgout_e;	

eTrgSrc	typedef enum{ GPTA_TRG01_DIS = 0, GPTA_TRG01_ZRO, GPTA_TRG01_PRD, GPTA_TRG01_ZRO_PRD, GPTA_TRG01_CMPA_R, GPTA_TRG01_CMPA_F, GPTA_TRG01_CMPB_R, GPTA_TRG01_CMPB_F, GPTA_TRG01_SYNC = 0xc, GPTA_TRG01_PE0, GPTA_TRG01_PE1, GPTA_TRG01_PE2 }csp_gpta_trgsrc0_e;	在csp_gpta.h中定义
csi_error_t	csi_error_t中定义	在common.h中定义

17.3.30 csi_gpta_set_evcntinit

*csi_error_t csi_gpta_set_evcntinit(csp_gpta_t *ptGptaBase, uint8_t byCntChx, uint8_t byCntVal, uint8_t byCntInitVal)*

17.3.30.1 功能描述

事件输出计数功能配置

17.3.30.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见 csp_gpta_t

byCntChx： 事件通道选择0~1

byCntVal： 事件周期值

byCntInitVal： 事件周期初始

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在gpta.h中定义
byCntChx	事件通道选择（0~1）	
byCntVal	事件周期值	
byCntInitVal	事件周期初始化值	

17.3.31 csi_gpta_reglk_config

*csi_error_t csi_gpta_reglk_config(csp_gpta_t *ptGptaBase,csi_gpta_feglk_config_t *Global)*

17.3.31.1 功能描述

链接寄存功能设置

17.3.31.2 参数/返回值说明

1. 参数

ptGptaBase： 定义详见 csp_gpta_t

Global： 定义详见 csi_gpta_feglk_config_t

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptGptaBase	csp_gpta_t中说明	在csp_ gpta.h中定义

Global	typedef struct { uint8_t byPrdr; uint8_t byCmpa; uint8_t byCmpb; uint8_t byGld2; uint8_t byRssr; uint8_t byEmslclr; uint8_t byEmhlclr; uint8_t byEmicr; uint8_t byEmfrcr; uint8_t byAqosf; uint8_t byAqcsf; } csi_gpta_feglk_config_t;	在gpta.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

18 CMP

18.1 概述

模拟比较器通过比较两个模拟输入电压量，输出一个数字标志用以表示两个输入量的幅值大小，是模拟电路和数字电路的的一种转换接口。模拟比较器在数模混合电路中作为非常重要的一种构建单元，可以提供独立于程序运行的模拟功能。

ATP CSI 接口 CMP 的设计中，提供丰富的配置及其操作。包含基本配置功能、数字滤波配置功能及窗口滤波控制功能等。

18.2 API列表

Table 18-1 CMP CSI接口函数

API	说明	函数位置
csi_cmp_int_enable	使能CMP中断	cmp.c
csi_cmp_init	初始化CMP基本功能	
csi_cmp_start	开始CMP功能	
csi_cmp_stop	停止CMP功能	
csi_cmp_fltcr_config	配置CMP数字滤波功能	
csi_cmp_wfcr_config	配置CMP窗口滤波功能	
csi_cmp_set_evtrg	事件触发边沿选择	
csi_cmp_inpcr_step	CMP负向输入步进配置	
csi_cmp_sync_step	CMP同步输入步进配置	
csi_cmp_set_sync	CMP同步步进输入触发模式配置	
csi_cmp_trgcr_ad_enable	CMP触发ADC采集的硬件使能控制	
csi_cmp_get_out	获取输出状态	
csi_cmp_int_clear	清除中断状态	
csi_cmp_get_misr	获取中断状态	

18.3 API详细说明

18.3.1 csi_cmp_int_enable

*void csi_cmp_int_enable(csp_cmp_t *ptCmpBase, csi_cmp_intsrc_e eIntSrc,bool bEnable,uint8_t byIdx)*

18.3.1.1 功能描述

使能CMP中断

18.3.1.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

eIntSrc：中断模式，枚举定义详见csi_cmp_intsrc_e。

bEnable：启用中断或禁止中断。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值：无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
eIntSrc	typedef enum { CMP_INTSRC_NONE = (0x00ul << 0), CMP_INTSRC_EDGEDET0 = (0x01ul << 0), CMP_INTSRC_EDGEDET1 = (0x01ul << 1), CMP_INTSRC_EDGEDET2 = (0x01ul << 2), CMP_INTSRC_EDGEDET3 = (0x01ul << 3), CMP_INTSRC_EDGEDET4 = (0x01ul << 4), CMP_INTSRC_EDGEDET5 = (0x01ul << 5), CMP_INTSRC_CMP1_NMAX = (0x01ul << 6), }csi_cmp_intsrc_e;	CMP的中断类型 csi_cmp_intsrc_e在cmp.h中定义
bEnable	bool类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义
byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP

18.3.2 csi_cmp_init

*csi_error_t csi_cmp_init(csp_cmp_t *ptCmpBase,csi_cmp_config_t *ptCmpCfg,uint8_t byIdx)*

18.3.2.1 功能描述

初始化CMP基本功能

18.3.2.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

ptCmpCfg：CMP基本配置结构体指针，结构体定义详见csi_cmp_config_t。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptSioBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义

ptCmpCfg	typedef struct
	{
	uint8_t byNsel; //N- pin
	uint8_t byPsel; //P+ pin
	uint8_t byNhystpol;
	uint8_t byPhystpol;
	uint8_t byVolSel;
	uint8_t byPolarity;
	uint8_t byCpoSel;
	uint8_t byEveSel;
	uint32_t wInt;
	}csi_cmp_config_t;
		在cmp.h中定义

byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP
csi_error_t	csi_error_t 中定义值	在common.h中定义

18.3.3 csi_cmp_start

*void csi_cmp_start(csp_cmp_t *ptCmpBase,uint8_t byIdx)*

18.3.3.1 功能描述

开始CMP功能

18.3.3.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值：无

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP

18.3.4 csi_cmp_stop

*void csi_cmp_stop(csp_cmp_t *ptCmpBase,uint8_t byIdx)*

18.3.4.1 功能描述

停止CMP功能

18.3.4.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值：无

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP

18.3.5 csi_cmp_fltcr_config

*csi_error_t csi_cmp_fltcr_config(csp_cmp_t *ptCmpBase,csi_cmp_fltcr_config_t *ptCmpFltcrCfg,uint8_t byIdx)*

18.3.5.1 功能描述

配置CMP数字滤波功能

18.3.5.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

ptCmpFltcrCfg：CMP数字滤波配置结构体指针，结构体定义详见csi_cmp_fltcr_config_t。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptSioBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
ptCmpFltcrCfg	typedef struct { uint8_t byCksrc; uint8_t byDivn; uint8_t byDivm; }csi_cmp_fltcr_config_t;	在cmp.h中定义

byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP
csi_error_t	csi_error_t 中定义值	在common.h中定义

18.3.6 csi_cmp_wfcr_config

*csi_error_t csi_cmp_wfcr_config(csp_cmp_t *ptCmpBase,csi_cmp_wfcr_config_t *ptCmpWfcrCfg,uint8_t byIdx)*

18.3.6.1 功能描述

配置CMP窗口滤波功能

18.3.6.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

ptCmpWfcrCfg：CMP窗口滤波配置结构体指针，结构体定义详见csi_cmp_wfcr_config_t。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数说明

参数	说明	概述及其枚举/结构体定义位置
ptSioBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型csp_cmp_t在csp_cmp.h中定义
ptCmpWfcrCfg	typedef struct { uint8_t byHls; uint8_t byMskMod; uint8_t byTrgSel; uint8_t byClkDiv; uint8_t byDcnt; uint8_t byRev; uint16_t hwWcnt; }csi_cmp_wfcr_config_t;	在cmp.h中定义

byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP
csi_error_t	csi_error_t 中定义值	在common.h中定义

18.3.7 csi_cmp_set_evtrg

*void csi_cmp_set_evtrg(csp_cmp_t *ptCmpBase,csi_eve_sel_e eEveSel, uint8_t byIdx)*

18.3.7.1 功能描述

事件触发边沿选择

18.3.7.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

eEveSel：事件触发边沿选择，枚举定义详见csi_eve_sel_e。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
eEveSel	typedef enum { CMP_EVE_DOWN = 0x00, CMP_EVE_UP, CMP_EVE_DOWN_UP, CMP_EVE_DOWN_UP1 }csi_eve_sel_e;	CMP同步触发边沿选择 csi_eve_sel_e在cmp.h中定义
byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP

18.3.8 csi_cmp_inpcr_step

*vvoid csi_cmp_inpcr_step(csp_cmp_t *ptCmpBase ,uint16_t hwNselMax,uint8_t byStepDiv)*

18.3.8.1 功能描述

CMP负向输入步进配置

18.3.8.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

hwNselMax：比较器步进最终负向输入选择。

byStepDiv：比较器1负向输入步进模式的时钟分频系数。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
hwNselMax	uint16_t 类型数据 0~126	比较器步进最终负向输入选择: VOL_REF/126 * hwNselMax
byStepDiv	uint8_t 类型数据，比较器负向输入步进模式的时钟分频系数	步进幅度等于： byStepDiv +1个PCLK

18.3.9 csi_cmp_sync_step

*void csi_cmp_sync_step(csp_cmp_t *ptCmpBase ,uint16_t hwNselMax,uint8_t byStepDiv)*

18.3.9.1 功能描述

CMP同步输入步进配置

18.3.9.2 参数/返回值说明

1. 参数

   ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

   hwNselMax：比较器步进最终负向输入选择。

   byStepDiv：比较器1负向输入步进模式的时钟分频系数。

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
hwNselMax	uint16_t 类型数据 0~126	比较器步进最终负向输入选择: VOL_REF/126 * hwNselMax
byStepDiv	uint8_t 类型数据，比较器负向输入步进模式的时钟分频系数	步进幅度等于： byStepDiv +1个PCLK

18.3.10 csi_cmp_set_sync

*void csi_cmp_set_sync(csp_cmp_t *ptCmpBase,csi_ostmd_e eOstMode, bool bEnableRearm,bool bEnableArearm)*

18.3.10.1 功能描述

CMP同步步进输入触发模式配置

18.3.10.2 参数/返回值说明

1. 参数

   ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

eOstMode：触发模式，枚举定义详见csi_ostmd_e。

bEnableRearm：在一次性同步触发模式下，软件重置当前通道状态控制位。

bEnableArearm：启用或禁止REARM控制位。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
eOstMode	typedef enum { CMP_OSTMD_CONTINUOUS = 0, CMP_OSTMD_ONCE }csi_ostmd_e;	CMP触发模式 csi_ostmd_e在cmp.h中定义
bEnableRearm	bool类型数值，ENBALE/DISABLE	ENBALE：清除当前通道状态，并允许新的触发 DISABLE：无效
bEnableArearm	bool类型数值，ENBALE/DISABLE	ENBALE：负向输入已经达到NSEL_MAX，自动REARM DISABLE:禁止硬件自动REARM

18.3.11 csi_cmp_trgcr_ad_enable

*void csi_cmp_trgcr_ad_enable(csp_cmp_t *ptCmpBase ,csi_ad_trgx_e eAdTrgx,bool bEnable)*

18.3.11.1 功能描述

CMP触发ADC采集的硬件使能控制

18.3.11.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

eAdTrgx：CMPx的ADC转换硬件触发使能控制，枚举定义详见csi_ad_trgx_e。

bEnable：启用或禁止CMP触发ADC。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
eAdTrgx	typedef enum { CMP_AD_TRG0 = (1<<0), CMP_AD_TRG1 = (1<<1), CMP_AD_TRG2 = (1<<2), CMP_AD_TRG3 = (1<<3), CMP_AD_TRG4 = (1<<4), CMP_AD_TRG5 = (1<<5) }csi_ad_trgx_e;	CMP的触发ADC使能控制 csi_ad_trgx_e在cmp.h中定义
bEnable	bool类型数值，ENBALE/DISABLE	ENBALE：使能CMP触发ADC DISABLE：禁止CMP触发ADC 在common.h中定义

18.3.12 csi_cmp_get_out

*uint8_t csi_cmp_get_out(csp_cmp_t *ptCmpBase,uint8_t byIdx)*

18.3.12.1 功能描述

获取输出状态

18.3.12.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

byIdx： CMP id号,可选择cmp0~cmp5。

2. 返回值

输出状态,0：低电平；1：高电平。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
byIdx	uint8_t 类型数据 CMP 选择，CMP0 ~ CMP5	选择对应的CMP
return value	uint8_t 类型数据，具体返回值如下 返回值,0：低电平；1：高电平。	输出状态

18.3.13 csi_cmp_int_clear

*void csi_cmp_int_clear(csp_cmp_t *ptCmpBase,csi_cmp_intsrc_e eIntMode)*

18.3.13.1 功能描述

清除中断状态。

18.3.13.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

eIntMode：中断模式，枚举定义详见csi_cmp_intsrc_e。

2. 返回值

无返回值。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义

eIntMode	typedef enum { CMP_INTSRC_NONE = (0x00ul << 0), CMP_INTSRC_EDGEDET0 = (0x01ul << 0), CMP_INTSRC_EDGEDET1 = (0x01ul << 1), CMP_INTSRC_EDGEDET2 = (0x01ul << 2), CMP_INTSRC_EDGEDET3 = (0x01ul << 3), CMP_INTSRC_EDGEDET4 = (0x01ul << 4), CMP_INTSRC_EDGEDET5 = (0x01ul << 5), CMP_INTSRC_CMP1_NMAX = (0x01ul << 6), }csi_cmp_intsrc_e;	CMP的中断类型 csi_cmp_intsrc_e在cmp.h中定义

18.3.14 csi_cmp_get_misr

*uint32_t csi_cmp_get_misr(csp_cmp_t *ptCmpBase)*

18.3.14.1 功能描述

获取中断状态

18.3.14.2 参数/返回值说明

1. 参数

ptCmpBase：CMP寄存器结构体指针，指向CMP基地址，结构体定义详见csp_cmp_t。

2. 返回值

uint32_t类型的返回数据，返回的为中断状态。

3. 返回值说明

返回值	说明	概述及其枚举定义位置
ptCmpBase	csp_cmp_t 类型指针，指向CMP的基地址	CMP寄存器结构体指针类型 csp_cmp_t在csp_cmp.h中定义
Return value	uint32_t类型的数据，返回中断状态,每位表示不同的中断。 0：中断未发生 1：中断发生	返回中断状态

19 EPT

19.1 概述

增强型通用定时器（Enhanced Purpose Timer）作为MCU的关键外设，可以在各种功率控制应用中发挥关键控制作用。通过灵活的PWM输出，可以适用于各种复杂多变的应用，这些应用包括数字马达控制、开关电源控制、不间断电源控制、变频功率转换控制等。EPT内部包含一个16位的定时/计数模块，支持2种工作模式(捕捉模式和波形发生器模式)。

下图为EPT框图的一部分，用于配合api函数说明。

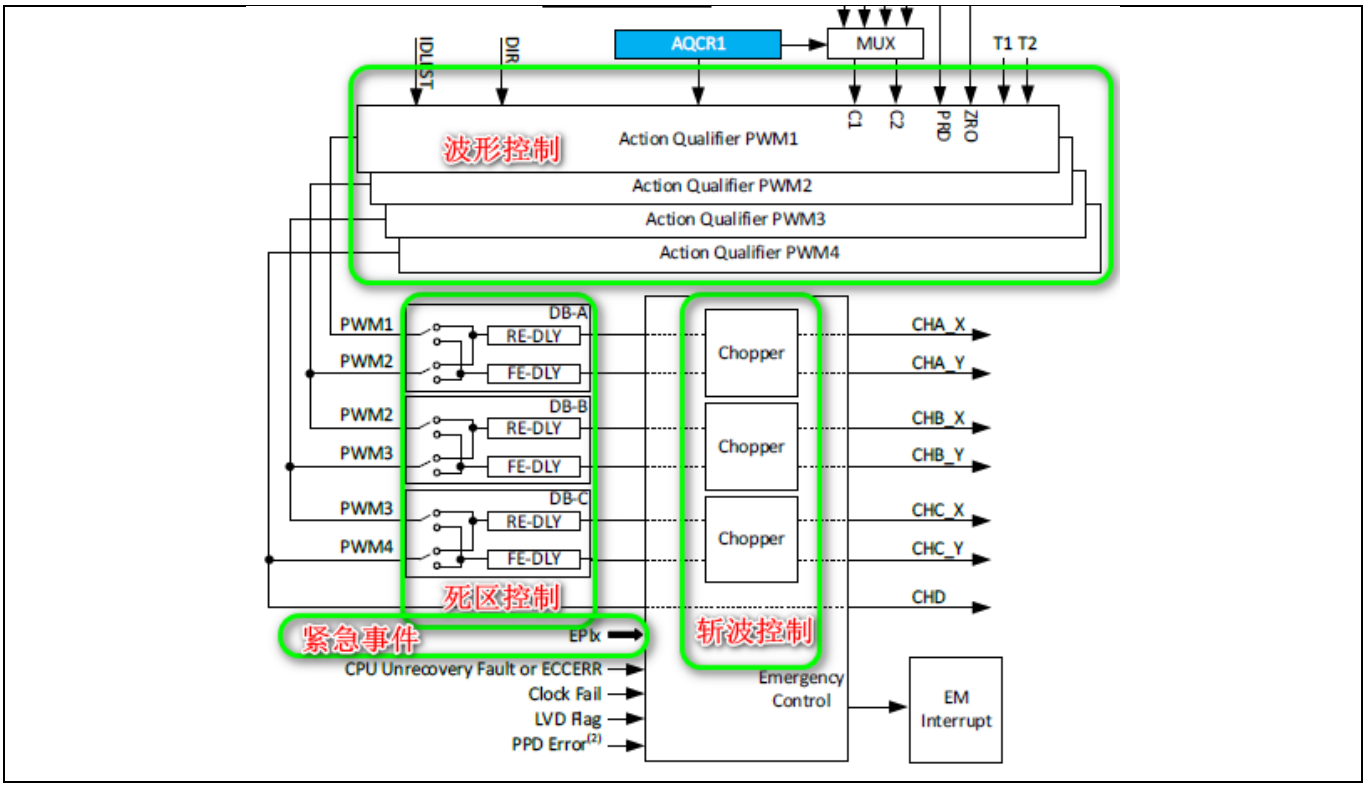


Figure 18-1 波形控制

19.2 API列表

Table 18-1 EPT CSI接口函数

API	说明	函数位置
ept_initen_irqhandler	定时器中断回调函数	
csi_ept_capture_init	定时器捕获模式参数设置	
csi_ept_wave_init	定时器波形模式参数基础设置（计数模式，周期，占空比等）	

csi_ept_channel_config	设置通道PWM1、PWM2、PWM3、PWM4的波形	
csi_ept_channel_cmpload_config	定时器CMPA.CMPB.CMPC.CMPD载入时机设置	
csi_ept_channel_aqload_config	通道PWM1、PWM2、PWM3、PWM4的波形载入时机设置	
csi_ept_dbload_config	死区配置载入控制	
csi_ept_dz_config	死区功能基本配置	
csi_ept_channelmode_config	死区通道配置（CH_A, CH_B, CH_C）	
csi_ept_chopper_config	斩波输出控制	
csi_ept_chopper_enable	通道斩波输出使能	
csi_ept_emergency_config	紧急状态输入基本参数设置	
csi_ept_emergency_pinout	设置紧急事件到来时，CHA/B/C/D_X/Y处的状态。	
csi_ept_ gload_config	全局载入控制配置	
csi_ept_gldcfg	全局载入对象使能或禁止	
csi_ept_ gload_sw	软件产生一次全局载入	
csi_ept_ gload_rearm	单次模式下全局载入重使能	
csi_ept_start	定时器开始计数	
csi_ept_stop	定时器停止计数	
csi_ept_set_start_mode	定时器开始位模式选择（普通开始/开始的同时触发同步事件0）	
csi_ept_set_os_mode	定时器模式选择（单次模式或连续模式）	
csi_ept_set_stop_st	波形输出被停止时，输出端口的缺省状态	
csi_ept_get_prdr	读取定时器周期寄存器	
csi_ept_change_ch_duty	改变比较值寄存器	
csi_ept_force_em	紧急状态软件触发	
csi_ept_get_hdlck_st	获取紧急硬锁止状态寄存器	
csi_ept_clr_hdlck	清除紧急硬锁止状态寄存器	
csi_ept_get_sftlck_st	获取紧急软锁止状态寄存器	
csp_ept_clr_sftlck	紧急软锁止状态清除寄存器	
csi_ept_debug_enable	使能或禁止debug模式	
csi_ept_emergency_int_enable	紧急状态中断使能或禁止	
csi_ept_evtrg_enable	使能或禁止事件	
csi_ept_onetime_software_output	一次性软件波形控制设置	
csi_ept_aqcsfload_config	AQCSF寄存器载入时机设置	
csi_ept_continuous_software_output	持续性软件波形控制设置	
csi_ept_int_enable	中断使能控制器	
csi_ept_set_sync	同步事件使能	
csi_ept_set_sync2evtrg	同步事件用于事件触发输出	
csi_ept_set_sync_filter	同步事件滤波器控制	
csi_ept_rearm_sync	在单次模式下同步事件重新使能	

csi_ept_set_evtrg	事件触发控制设置	
csi_ept_set_evcntinit	事件输出计数功能配置	
csi_ept_reglk_config	链接寄存功能设置	

19.3 API详细说明

19.3.1 ept_initen_irqhandler

*__attribute__((weak)) void ept_initen_irqhandler(csp_ept_t *ptEptBase)*

19.3.1.1 功能描述

定时器中断回调函数。

19.3.1.2 参数/返回值说明

1. 参数

ptEptBase： 每个功能模块都有对应的ptEptBase，枚举定义详见csp_ept_t

2. 返回值

无

3. 参数说明表

参数	说明	概述及其变量位置
ptEptBase	csp_ept_t结构体，用于进行寄存器操作。	定时器模块所有寄存器的定义 在csp_ ept.h中定义

19.3.2 csi_ept_capture_init

*csi_error_t csi_ept_capture_init(csp_ept_t *ptEptBase, csi_ept_captureconfig_t *pteptPwmCfg)*

19.3.2.1 功能描述

捕获模式下定时器基本参数设定

19.3.2.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见csp_ept_t

pteptPwmCfg： 定义详见csi_ept_captureconfig_t

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

pteptPwmCfg	typedef struct csi_ept_captureconfig csi_ept_captureconfig_t;
	struct csi_ept_captureconfig {
	uint8_t byWorkmod; //Count or capture
	uint8_t byCountingMode; //csi_ept_cntmd_e
	uint8_t byOneshotMode; //Single or continuous
	uint8_t byStartSrc ;
	uint8_t byPscld;
	uint8_t byDutyCycle; //TIMER PWM OUTPUT duty cycle
	uint8_t byCaptureCapLden;
	uint8_t byCaptureRearm;
	uint8_t byCaptureCapmd;
	uint8_t byCaptureStopWrap;
	uint8_t byCaptureLdaret;
	uint8_t byCaptureLdbret;
	uint8_t byCaptureLdcret;
	uint8_t byCaptureLddret;
	uint32_t wInt;
	uint8_t byBurst;
	uint8_t byCgsrc;
	uint8_t byCgflt;
	};
		在ept.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.3 csi_ept_wave_init

*csi_error_t csi_ept_wave_init(csp_ept_t *ptEptBase, csi_ept_pwmconfig_t *pteptPwmCfg)*

19.3.3.1 功能描述

定时器波形模式参数基础设置（计数模式，周期，占空比等）

19.3.3.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见csp_ept_t

pteptPwmCfg： 定义详见csi_ept_pwmconfig_t

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

pteptPwmCfg	typedef struct csi_ept_pwmconfig csi_ept_pwmconfig_t;
	struct csi_ept_pwmconfig {
	uint8_t byWorkmod; //Count or capture
	uint8_t byCountingMode; //csi_ept_cntmd_e
	uint8_t byOneshotMode; //Single or continuous
	uint8_t byStartSrc ;
	uint8_t byPscld;
	uint8_t byDutyCycle; //TIMER PWM OUTPUT duty cycle
	uint32_t wFreq; //TIMER PWM OUTPUT frequency
	uint32_t wInt;
	uint8_t byBurst;
	uint8_t byCgsrc;
	uint8_t byCgflt;
	};
		在ept.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.4 csi_ept_channel_config

*csi_error_t csi_ept_channel_config(csp_ept_t *ptEptBase, csi_ept_pwmchannel_config_t *tPwmCfg, csi_ept_channel_e channel)*

19.3.4.1 功能描述

设置通道PWM1、PWM2、PWM3、PWM4的波形

19.3.4.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见csp_ept_t

tPwmCfg： 定义详见csi_ept_pwmchannel_config_t

channel： 定义详见 csi_ept_channel_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

tPwmCfg	typedef struct csi_ept_pwmchannel_config csi_ept_pwmchannel_config_t;
	struct csi_ept_pwmchannel_config {
	uint8_t byActionZro; //
	uint8_t byActionPrd; //
	uint8_t byActionCau; //
	uint8_t byActionCad; //
	uint8_t byActionCbu; //
	uint8_t byActionCbd; //
	uint8_t byActionT1u; //
	uint8_t byActionT1d; //
	uint8_t byActionT2u; //
	uint8_t byActionT2d; //
	uint8_t byChoiceCasel;
	uint8_t byChoiceCbsel;
	};	

channel	typedef enum { EPT_CHANNEL_A=1, EPT_CHANNEL_B, EPT_CHANNEL_C, EPT_CHANNEL_D }csi_ept_channel_e;	
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.5 csi_ept_channel_cmpload_config

*csi_error_t csi_ept_channel_cmpload_config(csp_ept_t *ptEptBase, csp_ept_cmpdata_ldmd_e tld, csp_ept_ldamd_e tldamd ,csi_ept_camp_e channel)*

19.3.5.1 功能描述

定时器CMPA.CMPB.CMPC.CMPD载入时机设置

19.3.5.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

tld： 定义详见 csp_ept_cmpdata_ldmd_e

tldamd： 定义详见 csp_ept_ldamd_e

channel： 定义详见 csi_ept_camp_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
tld	typedef enum { EPT_CMPLD_SHDW = 0, EPT_CMPLD_IMM }csp_ept_cmpdata_ldmd_e;	在csp_ept.h中定义
tldamd	typedef enum{ EPT_LDCMP_NEVER=0, EPT_LDCMP_ZRO, EPT_LDCMP_PRD, EPT_LDCMP_LD_SYNC=4, }csp_ept_ldamd_e;	
channel	typedef enum{ EPT_CAMPA=1, EPT_CAMPB, EPT_CAMPC, EPT_CAMPD }csi_ept_camp_e;	
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.6 csi_ept_channel_aqload_config

*csi_error_t csi_ept_channel_aqload_config(csp_ept_t *ptEptBase, csp_ept_ld_e tld, csp_ept_ldamd_e tldamd ,csi_ept_channel_e channel)*

19.3.6.1 功能描述

通道PWM1、PWM2、PWM3、PWM4的波形载入时机设置

19.3.6.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见csp_ept_t

tld： 定义详见csp_ept_ld_e

tldamd： 定义详见csp_ept_ldamd_e

channel： 定义详见csi_ept_channel_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
tld	typedef enum{ EPT_LD_IMM = 0, EPT_LD_SHDW }csp_ept_ld_e;	在csp_ept.h中定义
tldamd	typedef enum{ EPT_LDCMP_NEVER=0, EPT_LDCMP_ZRO, EPT_LDCMP_PRD, EPT_LDCMP_LD_SYNC=4, }csp_ept_ldamd_e;	
channel	typedef enum{ EPT_CHANNEL_1=1, EPT_CHANNEL_2, EPT_CHANNEL_3, EPT_CHANNEL_4 }csi_ept_channel_e;	在ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.7 csi_ept_dbload_config

*csi_error_t csi_ept_dbload_config(csp_ept_t *ptEptBase, csi_ept_dbdldr_e byVal,csp_ept_shdw_e byWod,csp_ept_lddb_e byWod2)*

19.3.7.1 功能描述

死区配置载入控制

19.3.7.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

byVal： 定义详见 csi_ept_dbdldr_e

byWod： 定义详见 csp_ept_shdw_e

byWod2： 定义详见 csp_ept_lddb_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
byVal	typedef enum{ DBCR =0, DBDTR, DBDTF, DCKPSC, }csi_ept_dbdldr_e;	
byWod	typedef enum{ EPT_SHDW_IMMEDIATE =0, EPT_SHDW_SHADOW }csp_ept_shdw_e;	在csp_ept.h中定义
byWod2	typedef enum{ EPT_LD_NEVER = 0, EPT_LD_ZRO, EPT_LD_PRD, EPT_LD_ZRO_PRD }csp_ept_lddb_e;	
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.8 csi_ept_dz_config

*csi_error_t csi_ept_dz_config(csp_ept_t *ptEptBase, csi_ept_deadzone_config_t *tCfg)*

19.3.8.1 功能描述

死区功能基本配置

19.3.8.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

tCfg： 定义详见 csi_ept_deadzone_config_t

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

tCfg	typedef struct csi_ept_deadzone_config csi_ept_deadzone_config_t;
	struct csi_ept_deadzone_config
	{
	uint8_t byChxOuselS1S0 ; //
	uint8_t byChxPolarityS3S2 ; //
	uint8_t byChxInselS5S4 ; //
	uint8_t byChxOutSwapS8S7 ; //
	uint8_t byDcksel;
	uint8_t byChaDedb;
	uint8_t byChbDedb;
	uint8_t byChcDedb;
	uint16_t hwDpsc; //
	uint16_t hwRisingEdgereGister ; //
	uint16_t hwFallingEdgereGister; //
	};	

csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.9 csi_ept_channelmode_config

*csi_error_t csi_ept_channelmode_config(csp_ept_t *ptEptBase,csi_ept_deadzone_config_t *tCfg,csi_ept_channel_e byCh)*

19.3.9.1 功能描述

死区通道配置（CHANNEL_A, CHANNEL_B, CHANNEL_C）

19.3.9.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

tCfg： 定义详见 csi_ept_deadzone_config_t

byCh： 定义详见csi_ept_channel_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

tCfg	typedef struct csi_ept_deadzone_config csi_ept_deadzone_config_t;
	struct csi_ept_deadzone_config
	{
	uint8_t byChxOuselS1S0 ; //
	uint8_t byChxPolarityS3S2 ; //
	uint8_t byChxInselS5S4 ; //
	uint8_t byChxOutSwapS8S7 ; //
	uint8_t byDcksel;
	uint8_t byChaDedb;
	uint8_t byChbDedb;
	uint8_t byChcDedb;
	uint16_t hwDpsc; //
	uint16_t hwRisingEdgereGister ; //
	uint16_t hwFallingEdgereGister; //
	};
		在ept.h中定义

byCh	typedef enum{ EPT_CHANNEL_A=1, EPT_CHANNEL_B, EPT_CHANNEL_C, EPT_CHANNEL_D }csi_ept_channel_e;	在ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.10 csi_ept_chopper_config

*csi_error_t csi_ept_chopper_config(csp_ept_t *ptEptBase, csi_ept_Chopper_config_t *tCfg)*

19.3.10.1 功能描述

斩波输出控制

19.3.10.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

tCfg： 定义详见 csi_ept_Chopper_config_t

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

tCfg	typedef struct csi_ept_Chopper_config csi_ept_Chopper_config_t;
	struct csi_ept_Chopper_config
	{
	uint8_t byOspwth;
	uint8_t byCdiv;
	uint8_t byCduty;
	uint8_t byCasel;
	uint8_t chen;
	};
		在ept.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.11 csi_ept_chopper_enable

*csi_error_t csi_ept_chopper_enable(csp_ept_t *ptEptBase, csi_ept_chx_e byCh, bool bEn)*

19.3.11.1 功能描述

通道斩波输出使能

19.3.11.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

byCh： 定义详见 csi_ept_chx_e

bEn： ENABLE/DISABLE

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

tCfg	typedef enum{ EPTCHAX = 0x1, EPTCHAY, EPTCHBX, EPTCHBY, EPTCHCX, EPTCHCY }csi_ept_chx_e;	在ept.h中定义
bEn	ENABLE/DISABLE	在csp_common.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.12 csi_ept_emergency_config

*csi_error_t csi_ept_emergency_config (csp_ept_t *ptEptBase, csi_ept_emergency_config_t *tCfg)*

19.3.12.1 功能描述

紧急状态输入基本参数设置

19.3.12.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

tCfg： 定义详见 csi_ept_emergency_config_t

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

tCfg	typedef struct csi_ept_emergency_config csi_ept_emergency_config_t;
	struct csi_ept_emergency_config
	{
	uint8_t byEpxInt ;
	uint8_t byPolEbix;
	uint8_t byEpx;
	uint8_t byEpxLckmd;
	uint8_t byFltpace0;
	uint8_t byFltpace1;
	uint8_t byOrl0;
	uint8_t byOrl1;
	};
		在ept.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.13 csi_ept_emergency_pinout

*csi_error_t csi_ept_emergency_pinout(csp_ept_t *ptEptBase,csi_ept_osrchx_e byCh ,csp_ept_emout_e byCh2)*

19.3.13.1 功能描述

设置紧急事件到来时，CHA/B/C/D_X/Y处的状态

19.3.13.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

byCh： 定义详见 csi_ept_osrchx_e

byCh2： 定义详见 csp_et_emout_e

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
byCh	typedef enum{ EMCOAX =0, EMCOBX, EMCOCX, EMCOD, EMCOAY, EMCOBY, EMCOCY }csi_ept_osrchx_e;	在ept.h中定义
byCh2	typedef enum { EM_OUT_HZ, EM_OUT_H, EM_OUT_L, EM_OUT_NONE }csp_ept_emout_e;	在ept.h中定义
csi_error_t	csi_error_t 中定义值	common.h

19.3.14 csi_ept_gload_config

*csi_error_t csi_ept_ gload _config(csp_ept_t *ptEptBase,csi_ept_Global_load_control_config_t *Global)*

19.3.14.1 功能描述

全局载入控制配置

19.3.14.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

Global： 定义详见 csi_ept_Global_load_control_config_t

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

Global	typedef struct csi_ept_Global_load_control_config csi_ept_Global_load_control_config_t;
	struct csi_ept_Global_load_control_config{
	bool bGlden;
	bool bOstmd;
	uint8_t bGldprd;
	uint8_t byGldmd;
	};
		在ept.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.15 csi_ept_gldcfg

*csi_error_t csi_ept_gldcfg(csp_ept_t *ptEptBase ,csi_ept_Global_load_gldcfg Glo,bool bENABLE)*

19.3.15.1 功能描述

全局载入对象使能或禁止

19.3.15.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   Glo： 定义详见 csi_ept_Global_load_gldcfg

   bENABLE: ENABLE/DISABLE

2. 返回值

   CSI_OK： 设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
Global	typedef enum { byprdr_B=0, bycmpa_B, bycmpb_B, byaqcra_B, byaqcrb_B, byaqcsf_B, }csi_gptb_Global_load_gldcfg;	在ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.16 csi_ept_gload_sw

*csi_error_t csi_ept_gload_sw(csp_ept_t *ptEptBase)*

19.3.16.1 功能描述

软件产生一次全局载入

19.3.16.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.17 csi_ept_gload_rearm

*csi_error_t csi_ept_gload_rearm(csp_ept_t *ptEptBase)*

19.3.17.1 功能描述

单次模式下全局载入重使能

19.3.17.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.18 csi_ept_start

*csi_error_t csi_ept_start(csp_ept_t *pteptBase)*

19.3.18.1 功能描述

定时器开始计数

19.3.18.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.19 csi_ept_swstop

*void csi_ept_stop(csp_ept_t * ptEptBase)*

19.3.19.1 功能描述

定时器停止计数

19.3.19.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置

ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

19.3.20 csi_ept_set_start_mode

*void csi_ept_set_start_mode(csp_ept_t * ptEptBase, csi_ept_stmd_e eMode)*

19.3.20.1 功能描述

定时器开始位模式选择（普通开始/开始的同时触发同步事件0）

19.3.20.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   eMode： 定义详见 csi_ept_stmd_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

eMode	`typedef enum {`
	`  EPT_SW = 0,`
	`  EPT_SYNC`
	`}csi_ept_stmd_e;`	

19.3.21 csi_ept_set_os_mode

*void csi_ept_set_os_mode(csp_ept_t * ptEptBase, csi_ept_opmd_e eMode)*

19.3.21.1 功能描述

定时器模式选择（单次模式或连续模式）

19.3.21.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   eMode： 定义详见 csi_ept_opmd_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
eMode	typedef enum { EPT_OP_CONT = 0, EPT_OP_OT, }csi_ept_opmd_e;	在ept.h中定义

19.3.22 csi_ept_set_stop_st

*void csi_ept_set_stop_st(csp_ept_t * ptEptBase, csp_ept_stpst_e eSt)*

19.3.22.1 功能描述

波形输出被停止时，输出端口的缺省状态

19.3.22.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

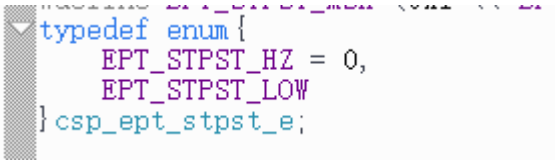
   eSt： 定义详见 csp_ept_stpst_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回	说明	概述及其变量位置

值		
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
eSt		

19.3.23 csi_ept_get_prdr

*uint16_t csi_ept_get_prdr(csp_ept_t * ptEptBase)*

19.3.23.1 功能描述

读取定时器周期寄存器

19.3.23.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

2. 返回值

   返回周期值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
uint16_t	返回周期值	

19.3.24 csi_ept_change_ch_duty

*csi_error_t csi_ept_change_ch_duty(csp_ept_t *ptEptBase, csi_ept_chtype_e eCh, uint32_t wActiveTime)*

19.3.24.1 功能描述

改变比较值寄存器

19.3.24.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   eCh： 定义详见 csi_ept_chtype_e

   wActiveTime： 周期值的X%

2. 返回值

   CSI_OK： 设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
eCh	typedef enum { EPT_CH_A = 0, EPT_CH_B, EPT_CH_C, EPT_CH_D }csi_ept_chtype_e;	在ept.h中定义
wActiveTime	周期值的X%	
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.25 csi_ept_force_em

*void csi_ept_force_em(csp_ept_t *ptEptBase, csp_ept_ep_e byEbi)*

19.3.25.1 功能描述

紧急状态软件触发

19.3.25.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   byEbi： 定义详见 csp_ept_ep_e

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
byEbi	typedef enum { EP0 = 0, EP1, EP2, EP3, EP4, EP5, EP6, EP7, } csp_ept_ep_e;	在csp_ept.h中定义

19.3.26 csi_ept_get_hdlck_st

*uint8_t csi_ept_get_hdlck_st(csp_ept_t *ptEptBase)*

19.3.26.1 功能描述

获取紧急硬锁止状态寄存器

19.3.26.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

2. 返回值

返回紧急硬锁止状态寄存器值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
uint8_t	返回紧急硬锁止状态寄存器值	

19.3.27 csi_ept_clr_hdlck

*void csi_ept_clr_hdlck(csp_ept_t *ptEptBase, csp_ept_ep_e eEbi)*

19.3.27.1 功能描述

清除紧急硬锁止状态寄存器

19.3.27.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

eEbi 定义详见 csp_ept_ep_e

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
eEbi	typedef enum{ EP0 = 0, EP1, EP2, EP3, EP4, EP5, EP6, EP7, }csp_ept_ep_e;	在csp_ept.h中定义

19.3.28 csi_ept_get_sftlck_st

*uint8_t csi_ept_get_sftlck_st(csp_ept_t *ptEptBase)*

19.3.28.1 功能描述

获取紧急软锁止状态寄存器

19.3.28.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

2. 返回值

   返回紧急软锁止状态值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
uint8_t	返回紧急软锁止状态值	

19.3.29 csp_ept_clr_sftlck

*void csp_ept_clr_sftlck(csp_ept_t *ptEptBase, csp_ept_ep_e eEpi)*

19.3.29.1 功能描述

紧急软锁止状态清除寄存器

19.3.29.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   eEpi： 定义详见 csp_ept_ep_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

eEpi	typedef enum{ EP0 = 0, EP1, EP2, EP3, EP4, EP5, EP6, EP7, }csp_ept_ep_e;	在csp_ept.h中定义

19.3.30 csi_ept_debug_enable

*void csi_ept_debug_enable(csp_ept_t *ptEptBase, bool bEnable)*

19.3.30.1 功能描述

使能或禁止debug模式

19.3.30.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

bEnable： ENABLE/DISABLE

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
bEnable	ENABLE/DISABLE	在common.h中定义

19.3.31 csi_ept_emergency_int_enable

*void csi_ept_emergency_int_enable(csp_ept_t *ptEptBase, csp_ept_emint_e eEbi)*

19.3.31.1 功能描述

紧急状态中断使能或禁止

19.3.31.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

eEbi： 定义详见 csp_ept_emint_e

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
eEbi	typedef enum { EPT_INT_EP0 = 0x1, EPT_INT_EP1 = 0x1 << 1, EPT_INT_EP2 = 0x1 << 2, EPT_INT_EP3 = 0x1 << 3, EPT_INT_EP4 = 0x1 << 4, EPT_INT_EP5 = 0x1 << 5, EPT_INT_EP6 = 0x1 << 6, EPT_INT_EP7 = 0x1 << 7, EPT_INT_CPUF= 0x1 << 8, EPT_INT_MEMF= 0x1 << 9, EPT_INT_EOMF= 0x1 << 10 }csp_ept_emint_e;	在csp_ept.h中定义

19.3.32 csi_ept_evtrg_enable

*csi_error_t csi_ept_evtrg_enable(csp_ept_t *ptEptBase, uint8_t byCh, bool bEnable)*

19.3.32.1 功能描述

使能或禁止事件

19.3.32.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

byCh： 0/1/2/3

bEnable ENABLE/DISABLE

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
byCh	0/1/2/3	
bEnable	ENABLE/DISABLE	
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.33 csi_ept_onetime_software_output

*csi_error_t csi_ept_onetime_software_output(csp_ept_t *ptEptBase, uint16_t byCh, csp_ept_action_e bEnable)*

19.3.33.1 功能描述

一次性软件波形控制设置

19.3.33.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

byCh： EPT_OSTSFA/EPT_OSTSFB/EPT_OSTSFC/EPT_OSTSFD

bEnable ENABLE/DISABLE

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
byCh	EPT_OSTSFA/EPT_OSTSFB/EPT_OSTSFC/EPT_OSTSFD	
bEnable	typedef enum { NA = 0, LO, HI, TG }csp_ept_action_e;	在csp_ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.34 csi_ept_aqcsfload_config

*void csi_ept_aqcsfload_config (csp_ept_t *ptEptBase, csp_ept_aqosf_e bEnable)*

19.3.34.1 功能描述

AQCSF寄存器载入时机设置

19.3.34.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

bEnable 定义详见 csp_ept_aqosf_e

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
bEnable	typedef enum { EPT_AQCSF_NOW=0, EPT_AQCSF_ZRO, EPT_AQCSF_PRD, EPT_AQCSF_ZRO_PRD }csp_ept_aqosf_e;	在csp_ept.h中定义

19.3.35 csi_ept_continuous_software_output

*csi_error_t csi_ept_continuous_software_output(csp_ept_t *ptEptBase, csi_ept_channel_e byCh, csp_ept_aqcsf_e bEnable)*

19.3.35.1 功能描述

持续性软件波形控制设置

19.3.35.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

byCh 定义详见 csi_ept_channel_e

bEnable 定义详见 csp_ept_aqcsf_e

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
byCh	`typedef enum{ EPT_CHANNEL_A=1, EPT_CHANNEL_B, EPT_CHANNEL_C, EPT_CHANNEL_D }csi_ept_channel_e;`	在ept.h中定义
bEnable	`typedef enum { EM_AQCSF_NONE=0, EM_AQCSF_L, EM_AQCSF_H, EM_AQCSF_NONE1 }csp_ept_aqcsf_e;`	在csp_ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.36 csi_ept_int_enable

*csi_error_t csi_ept_int_enable(csp_ept_t *ptEptBase, csp_ept_int_e eInt, bool bEnable)*

19.3.36.1 功能描述

中断使能控制器

19.3.36.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

eInt： 定义详见 csp_ept_int_e

bEnable ENABLE/DISABLE

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
eInt	typedef enum { EPT_INT_TRGEV0 = 0x1, EPT_INT_TRGEV1 = 0x2, EPT_INT_TRGEV2 = 0x4, EPT_INT_TRGEV3 = 0x8, EPT_INT_CAPLD0 = 0x1 << 4, EPT_INT_CAPLD1 = 0x1 << 5, EPT_INT_CAPLD2 = 0x1 << 6, EPT_INT_CAPLD3 = 0x1 << 7, EPT_INT_CAU = 0x1 << 8, EPT_INT_CAD = 0x1 << 9, EPT_INT_CBU = 0x1 << 10, EPT_INT_CBD = 0x1 << 11, EPT_INT_CCU = 0x1 << 12, EPT_INT_CCD = 0x1 << 13, EPT_INT_CDU = 0x1 << 14, EPT_INT_CDD = 0x1 << 15, EPT_INT_PEND = 0x1 << 16 }csp_ept_int_e;	在csp_ept.h中定义
bEnable	ENABLE/DISABLE	
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.37 csi_ept_set_sync

*void csi_ept_set_sync(csp_ept_t *ptEptBase, csi_ept_trgin_e eTrgIn, csi_ept_trgmode_e eTrgMode, csi_ept_arearm_e eAutoRearm)*

19.3.37.1 功能描述

同步事件使能

19.3.37.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

eTrgIn： 定义详见 csi_ept_trgin_e

eTrgMode： 定义详见 csi_ept_trgmode_e

eAutoRearm： 定义详见 csi_ept_arearm_e

2. 返回值

无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

eTrgIn	typedef enum{
	EPT_TRG_SYNCEN0 = 0, //start up or reset count
	EPT_TRG_SYNCEN1, //reg updata
	EPT_TRG_SYNCEN2, //capture
	EPT_TRG_SYNCEN3, //count inc or dec
	EPT_TRG_SYNCEN4, //change output status(pwm)
	EPT_TRG_SYNCEN5 //change output status(pwm)
	}csi_ept_trgin_e;
		在ept.h中定义

eTrgMode	typedef enum{
	EPT_TRG_CONTINU = 0, //EPT continuous trigger mode
	EPT_TRG_ONCE //EPT once trigger mode
	}csi_ept_trgmode_e;	

eAutoRearm	typedef enum{
	EPT_AUTO_REARM_DIS = 0, //disable auto rearm
	EPT_AUTO_REARM_ZRO, //CNT = ZRO auto rearm
	EPT_AUTO_REARM_PRD, //CNT = PRD auto rearm
	EPT_AUTO_REARM_ZRO_PRD //CNT = PRD or PRD auto rearm
	}csi_ept_arearm_e;	

19.3.38 csi_ept_set_sync2evtrg

*csi_error_t csi_ept_set_sync2evtrg(csp_ept_t *ptEptBase, csi_ept_trgin_e eTrgIn, uint8_t byTrgChx)*

19.3.38.1 功能描述

同步事件用于事件触发输出

19.3.38.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   eTrgIn： 选择用于事件的同步源

   byTrgChx： 选择通道号

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

eTrgIn	typedef enum{
	EPT_TRGIN_SYNCEN0 = 0, //start up or reset count
	EPT_TRGIN_SYNCEN1, //reg updata
	EPT_TRGIN_SYNCEN2, //capture
	EPT_TRGIN_SYNCEN3, //count inc or dec
	EPT_TRGIN_SYNCEN4, //change output status(pwm)
	EPT_TRGIN_SYNCEN5 //change output status(pwm)
	}csi_ept_trgin_e;
		在ept.h中定义

byTrgChx	0/1	

19.3.39 csi_ept_set_sync_filter

*csi_error_t csi_ept_set_sync_filter(csp_ept_t *ptEptBase, csi_ept_filter_config_t *ptFilter)*

19.3.39.1 功能描述

同步事件滤波器控制

19.3.39.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   ptFilter： 定义详见 csi_ept_filter_config_t

2. 返回值

   CSI_OK： 设置成功

   CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

ptFilter	typedef struct {
	uint8_t byFiltSrc; //filter input signal source
	uint8_t byWinInv; //window inversion
	uint8_t byWinAlign; //window alignment
	uint8_t byWinCross; //window cross
	uint16_t byWinOffset; //window offset
	uint16_t byWinWidth; //window width
	} csi_ept_filter_config_t;
		在ept.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.40 csi_ept_rearm_sync

*void csi_ept_rearm_sync(csp_ept_t *ptEptBase,csi_ept_trgin_e eTrgin)*

19.3.40.1 功能描述

在单次模式下同步事件重新使能

19.3.40.2 参数/返回值说明

1. 参数

   ptEptBase： 定义详见 csp_ept_t

   eTrgin： 定义详见 csi_ept_trgin_e

2. 返回值

   无返回值

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

eTrgin	typedef enum{
	EPT_TRGIN_SYNCEN0 = 0, //start up or reset count
	EPT_TRGIN_SYNCEN1, //reg updata
	EPT_TRGIN_SYNCEN2, //capture
	EPT_TRGIN_SYNCEN3, //count inc or dec
	EPT_TRGIN_SYNCEN4, //change output status(pwm)
	EPT_TRGIN_SYNCEN5 //change output status(pwm)
	}csi_ept_trgin_e;
		在ept.h中定义

19.3.41 csi_ept_set_evtrg

*csi_error_t csi_ept_set_evtrg(csp_ept_t *ptEptBase, csi_ept_trgout_e eTrgOut, csi_ept_trgsrc_e eTrgSrc)*

19.3.41.1 功能描述

事件触发控制设置

19.3.41.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

eTrgOut： 定义详见 csi_ept_trgout_e

eTrgSrc 定义详见 csi_ept_trgsrc_e

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEptBase	csp_ept_t中说明	在csp_ept.h中定义

eTrgOut	typedef enum{
	EPT_TRGOUT0 = 0, //trigger out0
	EPT_TRGOUT1, //trigger out1
	EPT_TRGOUT2, //trigger out2
	EPT_TRGOUT3 //trigger out3
	}csi_ept_trgout_e;	

eTrgSrc	typedef enum { EPT_TRGSRC_DIS = 0, EPT_TRGSRC_ZRO, EPT_TRGSRC_PRD, EPT_TRGSRC_ZRO_PRD, EPT_TRGSRC_CAU, EPT_TRGSRC_CAD, EPT_TRGSRC_CBU, EPT_TRGSRC_CBD, EPT_TRGSRC_CCU, EPT_TRGSRC_CCD, EPT_TRGSRC_CDU, EPT_TRGSRC_CDD, EPT_TRGSRC_EX, EPT_TRGSRC_PE0, EPT_TRGSRC_PE1, EPT_TRGSRC_PE2, EPT_TRGSRC_PEND }csi_ept_trgsrc_e;	在ept.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.42 csi_ept_set_evcntinit

*csi_error_t csi_ept_set_evcntinit(csp_ept_t *ptEptBase, uint8_t byCntChx, uint8_t byCntVal, uint8_t byCntInitVal)*

19.3.42.1 功能描述

事件输出计数功能配置

19.3.42.2 参数/返回值说明

1. 参数

ptEptBase： 定义详见 csp_ept_t

byCntChx： 事件通道选择0~3

byCntVal 事件周期值

byCntInitVal 事件周期初始

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置

ptEptBase	csp_ept_t中说明	在csp_ept.h中定义
byCntChx	EPT_TRG_OUT0~ EPT_TRG_OUT3	
byCntVal	事件周期值	
byCntInitVal	事件周期初始	
csi_error_t	csi_error_t 中定义值	在common.h中定义

19.3.43 csi_ept_reglk_config

*csi_error_t csi_ept_reglk_config(csp_ept_t *pteptBase,csi_ept_feglk_config_t *Global)*

19.3.43.1 功能描述

链接寄存功能设置

19.3.43.2 参数/返回值说明

1. 参数

pteptBase： 定义详见 csp_gptb_t

Global： 定义详见 csi_gptb_feglk_config_t

2. 返回值

CSI_OK： 设置成功

CSI_ERROR：设置失败

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
pteptBase	csp_gptb_t中说明	在csp_ gptb.h中定义
Global	typedef struct { uint8_t byPrdr; uint8_t byCmpa; uint8_t byCmpb; uint8_t byGld2; uint8_t byRssr; uint8_t byEmslclr; uint8_t byEmhlclr; uint8_t byEmicr; uint8_t byEmfrcr; uint8_t byAqosf; uint8_t byAqcsf; } csi_gptb_feglk_config_t;	在gptb.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

20 UART

20.1 概述

UART是一个简单通用的异步串行接收和发送接口，支持8位的数据通信，支持校验位，每次发送都以一个停止位结束。APT CSI接口提供了UART包括发送、接收等相关配置和操作。

20.2 API列表

Table20-1 UART CSI接口函数

API	说明	函数位置
csi_uart_init	初始化UART	uart.c
csi_uart_start	开启UART收发功能	
csi_uart_stop	关闭UART收发功能	
csi_uart_set_buffer	配置串口接收数据缓存(buffer)	
csi_uart_putc	发送一个字符	
csi_uart_getc	接收一个字符	
csi_uart_send	发送数据	
csi_uart_receive	接收数据	
csi_uart_get_msg	获取UART接收/发送是否完成消息	
csi_uart_clr_msg	清除UART接收/发送消息(设置为空闲)	

20.3 API详细说明

20.3.1 csi_uart_init

*csi_error_t csi_uart_init(csp_uart_t *ptUartBase, csi_uart_config_t *ptUartCfg)*

20.3.1.1 功能描述

初始化 UART

20.3.1.2 参数/返回值说明

1. 参数

ptUartBase：UART 寄存器结构体指针，指向 UART 基地址，结构体定义详见 csp_uart_t。

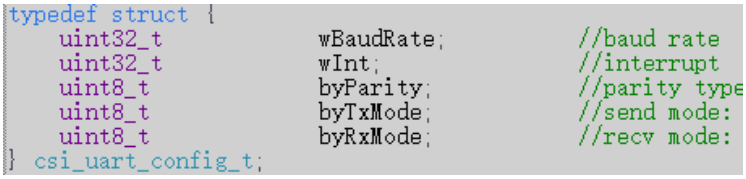
ptUartCfg：UART 配置结构体指针，结构体定义详见 csi_uart_config_t。

2. 返回值

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，UART0/UART1/UART2，指向对应UART的基地址	系统有一个个UART，即UART0，定义了对应的结构体指UART0，指向对应UART的基地址。 在devices.c中定义
ptUartCfg		初始化配置参数： wBaudRate：波特率 wInter：中断源选择 byParity：校验 byTxMode：发送模式 byRxMode：接收模式 在uart.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

20.3.2 csi_uart_start

*csi_error_t csi_uart_start(csp_uart_t *ptUartBase, csi_uart_func_e eFunc)*

20.3.2.1 功能描述

开启(使能)UART 收发功能

20.3.2.2 参数/返回值说明

1.参数

ptUartBase：UART 寄存器结构体指针，指向 UART 基地址，结构体定义详见 csp_uart_t。

eFunc：UART 的 RX/TX 使能，可以全部使能，也可以单独对 RX/TX 中某一个使能，枚举定义详见 csi_uart_func_e。

2.返回值

CSI_OK：成功。

CSI_ERROR：失败。

3.参数说明

参数/返回值	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义

eFunc	`typedef enum{`
	`  UART_FUNC_RX = 0, //uart only support rx`
	`  UART_FUNC_TX, //uart only support tx`
	`  UART_FUNC_RX_TX //uart support rx and tx`
	`}csi_uart_func_e;`
		UART_FUNC_RX：使能RX
		UART_FUNC_TX：使能TX
		UART_FUNC_RX_TX：使能RX和TX
		在uart.h钟定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

20.3.3 csi_uart_stop

*csi_error_t csi_uart_stop(csp_uart_t *ptUartBase, csi_uart_func_e eFunc)*

20.3.3.1 功能描述

关闭(禁止)UART 收发功能

20.3.3.2 参数/返回值说明

1.参数

ptUartBase：UART 寄存器结构体指针，指向 UART 基地址，结构体定义详见 csp_uart_t。

eFunc：UART 的 RX/TX 禁止，可以全部禁止，也可以单独对 RX/TX 中某一个禁止，枚举定义详见 csi_uart_func_e。

2.返回值

CSI_OK：成功。

CSI_ERROR：失败。

3.参数/返回值说明

参数参数/返回值	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义

eFunc	typedef enum{
	UART_FUNC_RX = 0, //uart only support rx
	UART_FUNC_TX, //uart only support tx
	UART_FUNC_RX_TX //uart support rx and tx
	}csi_uart_func_e;
		UART_FUNC_RX：禁止RX
		UART_FUNC_TX：禁止TX
		UART_FUNC_RX_TX：禁止RX和TX
		在uart.h钟定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

20.3.4 csi_uart_set_buffer

*void csi_uart_set_buffer(csp_uart_t *ptUartBase, ringbuffer_t *ptRingbuf, uint8_t *pbyRdBuf, uint16_t hwLen)*

20.3.4.1 功能描述

配置串口接收数据缓存(buffer)，中断接收模式时调用

20.3.4.2 参数/返回值说明

1. 参数

ptUartBase：UART 寄存器结构体指针，指向 UART 基地址，结构体定义详见 csp_uart_t。

ptRingbuf：循环 buf(ringbuf)结构体指针，结构体定义详见 ringbuffer_t。

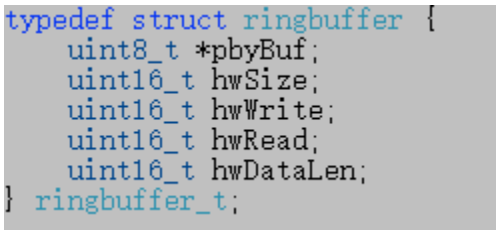
pbyRdBuf：串口接收数据缓存(buffer)数组指针，指向 buffer 首地址。

hwLen：接收缓存大小(接收数据数组长度)，用户定义。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义
ptRingbuf		参数说明： pbyBuf：buf指针，指向缓存 hwSize：循环buf大小 hwWrite：写入数据长度 hwRead：读取数据长度 hwDataLen：数据长度 在ringbuf.h中定义
pbyRdBuf	uint8_t 类型指针，指向接收数据缓存区首地址	指向接收数据缓存(接收数据数组)，赋值给循环buf的pbyBuf
hwLen	uint16_t 类型数据，缓存大小，即接收数组长度	赋值给循环buf的hwSize

20.3.5 csi_uart_putc

*void csi_uart_putc(csp_uart_t *ptUartBase, uint8_t byData)*

20.3.5.1 功能描述

UART 发送一个字符。

20.3.5.2 参数/返回值说明

1. 参数

ptUartBase：UART 寄存器结构体指针，指向 UART 基地址，结构体定义详见 csp_uart_t。

byData：要发送的字符

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义
byData	uint8_t 类型数据，要发送的字符	阻塞方式发送一个字符

20.3.6 csi_uart_getc

*uint8_t csi_uart_getc(csp_uart_t *ptUartBase)*

20.3.6.1 功能描述

UART接收一个字符

20.3.6.2 参数/返回值说明

1. 参数

ptUartBase：UART寄存器结构体指针，指向UART基地址，结构体定义详见csp_uart_t。

2. 返回值

接收到的字符。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义
return value	uint8_t 类型数据，接收到的字符	阻塞方式接收一个字符

20.3.7 csi_uart_send

*int32_t csi_uart_send(csp_uart_t *ptUartBase, const void *pData, uint16_t hwSize)*

20.3.7.1 功能描述

发送数据，UART发送有两种模式(轮训/中断)，初始化时用户可配置。

20.3.7.2 参数/返回值说明

1. 参数

ptUartBase：UART寄存器结构体指针，指向UART基地址，结构体定义详见csp_uart_t。

pData：指向要发送数据buffer首地址。

hwSize：要发送数据长度。

2. 返回值

已发送数据长度/ CSI_OK/ CSI_ERROR。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义
pData	void 类型指针，指向发送数据缓存首地址	发送数据缓存类型为uint8_t，调用时需强制转换下(void *)
hwSize	uint16_t 类型数据，要发送数据的长度	
return value	轮训模式：发送完成的数据长度 中断模式：CSI_OK/CSI_ERROR	UART发送数据两种模式返回数据不一样，具体如下： 轮训模式，返回发送数据长度 中断模式：返回发送成功/失败

20.3.8 csi_uart_receive

*uint16_t csi_uart_receive(csp_uart_t *ptUartBase, void *pData, uint16_t hwSize, uint32_t wTimeOut)*

20.3.8.1 功能描述

获取UART接收到的数据，UART接收有三种模式(轮训/中断1/中断2)，中断模式1为接收用户指定长度数据；中断模式2为动态接收一串字符，字符长度为动态(不固定)；接收模式初始化时用户可配置。

20.3.8.2 参数/返回值说明

1. 参数

ptUartBase：UART寄存器结构体指针，指向UART基地址，结构体定义详见csp_uart_t。

pData：指向用户获取数据buffer首地址，即把接收到的数据读取到用户定义的buffer中。

hwSize：要获取的数据长度，轮训模式和中断模式1时参数有意义；中断模式2忽略此参数。

wTimeOut：获取UART串口数据超时处理，轮训模式时参数有意义；另外两种模式忽略此参数。

2. 返回值

获取到的数据长度。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义
pData	void 类型指针，指向发送数据缓存首地址	获取数据缓存类型为uint8_t，调用时需强制转换下(void *)
hwSize	uint16_t 类型数据，要获取数据的长度	用户指定的数据长度
return value	uint16_t 类型数值，获取到的数据长度	实际返回的数据长度

20.3.9 csi_uart_get_msg

*bool csi_uart_get_msg(csp_uart_t *ptUartBase, csi_uart_wkmode_e eWkMode, bool bClrEn)*

20.3.9.1 功能描述

获取 UART 接收/发送数据是否完毕消息

20.3.9.2 参数/返回值说明

1. 参数

   ptUartBase：UART 寄存器结构体指针，指向 UART 基地址，结构体定义详见 csp_uart_t。

   eWkMode：UART 工作状态，接收/发送，枚举定义详见 csi_uart_wkmode_e。

   bClrEn：ENABLE/DISABLE,，获取到消息时，是否清除接收/发送状态（设置为空闲），使用时一般使能此选项。

2. 返回值

   布尔类型变量，true/ false(1/0)，true 表示接收/发送完毕，反之亦然。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义

eWkMode	typedef enum{
	UART_SEND = 0, //uart send
	UART_RECV = 1, //uart receive
	}csi_uart_wkmode_e;
		发送/接收两种模式
		在uart.h中定义

bClrEn	布尔类型，ENABLE/DISABLE	是否清除UART接收/发送状态（设置为空闲）
返回值	布尔类型，true/false(真/假)	true：1 false：0

20.3.10 csi_uart_clr_msg

*void csi_uart_clr_msg(csp_uart_t *ptUartBase, csi_uart_wkmode_e eWkMode)*

20.3.10.1 功能描述

清除 UART 接收/发送消息，即设置为空闲。

20.3.10.2 参数/返回值说明

1. 参数

   ptUartBase：UART 寄存器结构体指针，指向 UART 基地址，结构体定义详见 csp_uart_t。

   eWkMode：UART 工作状态，接收/发送，枚举定义详见 csi_uart_wkmode_e。

2. 返回值：无。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptUartBase	csp_uart_t 类型指针，请参阅20.3.1.2参数说明	在devices.c中定义

eWkMode	typedef enum{
	UART_SEND = 0, //uart send
	UART_RECV = 1, //uart receive
	}csi_uart_wkmode_e;
		发送/接收两种模式
		在uart.h中定义

21 USART

21.1 概述

USART 是一个简单通用的同步异步串行接收和发送接口，支持 5 到 9 位的数据通信，支持校验位，每次发送都以 1/1.5/2 个停止位结束。APT CSI 接口提供了 USART 包括发送、接收等相关配置和操作。

21.2 API列表

Table21-1 TC1 CSI接口函数

API	说明	函数位置
csi_usart_init	初始化USART	usart.c
csi_usart_int_enable	使能/禁止USART中断	
csi_usart_start	开启USART收发功能	
csi_usart_stop	关闭USART收发功能	
csi_usart_set_buffer	配置串口接收数据缓存(buffer)	
csi_usart_putc	发送一个字符	
csi_usart_getc	接收一个字符	
csi_usart_send	发送数据	
csi_usart_receive	接收数据	
csi_usart_get_msg	获取USART接收/发送完成信息，并清除/保留状态	
csi_usart_clr_msg	清除USART接收/发送完成状态信息	

21.2.1 API详细说明

21.2.2 csi_usart_init

*csi_error_t csi_usart_init(csp_usart_t *ptUsartBase, csi_usart_config_t *ptUsartCfg)*

21.2.2.1 功能描述

初始化USART

21.2.2.2 参数/返回值说明

1. 参数

   ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

   ptUsartCfg：USART配置结构体指针，结构体定义详见csi_usart_config_t。

2. 返回值

   CSI_OK：初始化成功。

   CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，USART0，指向对应USART的基地址	系统有一个USART，即USART0，定义了结构体指针USART0，指向对应USART的基地址。 在csp_usart.h中定义

ptUsartCfg	typedef struct {
	uint32_t wBaudRate; //baud rate
	uint32_t wInt; //interrupt
	uint8_t byParity; //parity type
	uint8_t byDatabit; //data bits
	uint8_t byStopbit; //stop bits
	uint8_t byClkSrc; //clk source
	uint8_t byMode; //usart mode, sync/async
	uint8_t byTxMode; //send mode: polling/interrupt
	uint8_t byRxMode; //recv mode: polling/interrupt0/interrupt1
	} csi_usart_config_t;
		初始化配置参数：
		wBaudRate：波特率
		wInter：中断源选择
		byParity：校验
		byDatabit：数据位数
		byStopbit：停止位数
		byClkSrc：时钟源选择
		byTxMode：发送模式
		byRxMode：接收模式
		在usart.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

21.2.3 csi_usart_int_enable

*void csi_usart_int_enable(csp_usart_t *ptUsartBase, csi_usart_intsrc_e eIntSrc, bool bEnable)*

21.2.3.1 功能描述

使能/禁止USART中断

21.2.3.2 参数/返回值说明

1. 参数

ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

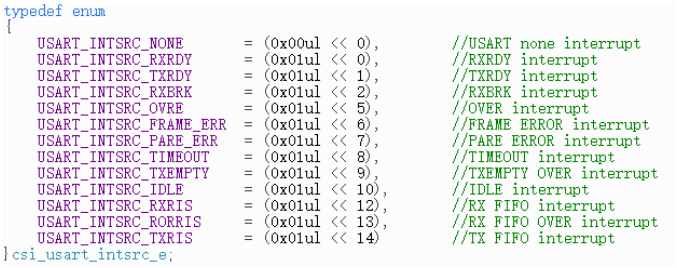
eIntSrc：USART中断源结构体，结构体定义详见csi_usart_intsrc_e。

bEnable：使能/禁止中断。

2. 返回值

无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，USART0，指向对应USART的基地址	系统有一个USART，即USART0，定义了结构体指USART0，指向对应USART的基地址。在csp_usart.h中定义
eIntSrc		初始化配置参数： USART_INTSRC_NONE：无中断 USART_INTSRC_RXRDY：接收端待机 USART_INTSRC_TXRDY：发送端待机 USART_INTSRC_RXBRK：接收端Break USART_INTSRC_OVRE：溢出错误 USART_INTSRC_FRAME_ERR：帧错误 USART_INTSRC_PARE_ERR：校验错 USART_INTSRC_TIMEOUT：超时 USART_INTSRC_TXEMPTY：发送缓冲空闲 USART_INTSRC_IDLE：空闲 USART_INTSRC_RXRIS：接收FIFO USART_INTSRC_RORRIS：接收FIFO溢出

		USART_INTSRC_TXRIS：发送FIFO 在usart.h中定义
bEnable	bool类型，使能/禁止中断	

21.2.4 csi_usart_start

*csi_error_t csi_usart_start(csp_usart_t *ptUsartBase, csi_usart_func_e eFunc)*

21.2.4.1 功能描述

开启(使能)USART收发功能

21.2.4.2 参数/返回值说明

1. 参数

ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

eFunc：UART的RX/TX使能，可以全部使能，也可以单独对RX/TX中某一个使能，枚举详见csi_usart_func_e。

2. 返回值

CSI_OK：成功。

CSI_ERROR：失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义

eFunc	typedef enum{
	UART_FUNC_RX = 0, //uart only support rx
	UART_FUNC_TX, //uart only support tx
	UART_FUNC_RX_TX //uart support rx and tx
	}csi_uart_func_e;
		USART_FUNC_RX：使能RX
		USART_FUNC_TX：使能TX
		USART_FUNC_RX_TX：使能RX和TX
		在usart.h钟定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

21.2.5 csi_usart_stop

*csi_error_t csi_usart_stop(csp_usart_t *ptUsartBase, csi_usart_func_e eFunc)*

21.2.5.1 功能描述

关闭(禁止)USART收发功能

21.2.5.2 参数/返回值说明

1. 参数

ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

eFunc：UART的RX/TX使能，可以全部使能，也可以单独对RX/TX中某一个使能，枚举详见csi_usart_func_e。

2. 返回值

CSI_OK：成功。

CSI_ERROR：失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义

eFunc	typedef enum{
	UART_FUNC_RX = 0, //uart only support rx
	UART_FUNC_TX, //uart only support tx
	UART_FUNC_RX_TX //uart support rx and tx
	}csi_uart_func_e;
		USART_FUNC_RX：禁止RX
		USART_FUNC_TX：禁止TX
		USART_FUNC_RX_TX：禁止RX和TX
		在usart.h钟定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

21.2.6 csi_usart_set_buffer

*void csi_usart_set_buffer(csp_usart_t *ptUsartBase, ringbuffer_t *ptRingbuf, uint8_t *pbyRdBuf, uint16_t hwLen)*

21.2.6.1 功能描述

配置串口接收数据缓存(buffer)，中断接收模式时调用

21.2.6.2 参数/返回值说明

1. 参数

ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

ptRingbuf：循环buf(ringbuf)结构体指针，结构体定义详见ringbuffer_t。

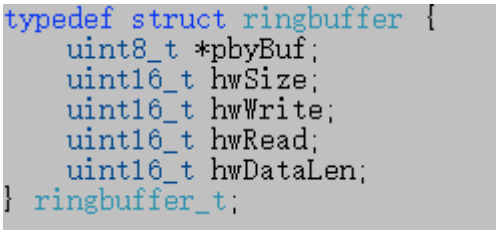
pbyRdBuf：串口接收数据缓存(buffer)数组指针，指向buffer首地址。

hwLen：接收缓存大小(接收数据数组长度)，用户定义。

2. 返回值

无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义
ptRingbuf		参数说明： pbyBuf：buf指针，指向缓存 hwSize：循环buf大小 hwWrite：写入数据长度 hwRead：读取数据长度 hwDataLen：数据长度 在ringbuf.h中定义
pbyRdBuf	uint8_t 类型指针，指向接收数据缓存区首地址	指向接收数据缓存(接收数据数组)，赋值给循环buf的pbyBuf
hwLen	uint16_t 类型数据，缓存大小，即接收数组长度	赋值给循环buf的hwSize

21.2.7 csi_usart_putc

*void csi_usart_putc(csp_usart_t *ptUsartBase, uint8_t byData)*

21.2.7.1 功能描述

USART发送一个字符。

21.2.7.2 参数/返回值说明

1. 参数

   ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

   byData：要发送的字符

2. 返回值

   无返回值。

3. 参数说明

参数	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义
byData	uint8_t 类型数据，要发送的字符	阻塞方式发送一个字符

21.2.8 csi_usart_getc

*uint8_t csi_usart_getc(csp_usart_t *ptUsartBase)*

21.2.8.1 功能描述

USART接收一个字符

21.2.8.2 参数/返回值说明

1. 参数

   ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

2. 返回值

   接收到的字符。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义
return value	uint8_t 类型数据，接收到的字符	阻塞方式接收一个字符

21.2.9 csi_usart_send

*int16_t csi_usart_send(csp_usart_t *ptUsartBase, const void *pData, uint16_t hwSize)*

21.2.9.1 功能描述

发送数据，USART发送有两种模式(轮训/中断)，初始化时用户可配置。

21.2.9.2 参数/返回值说明

1. 参数

ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

pData：指向要发送数据buffer首地址。

hwSize：要发送数据长度。

2. 返回值

已发送数据长度/ CSI_OK/ CSI_ERROR。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义
pData	void 类型指针，指向发送数据缓存首地址	发送数据缓存类型为uint8_t，调用时需强制转换下(void *)
hwSize	uint16_t 类型数据，要发送数据的长度	
return value	轮训模式：发送完成的数据长度 中断模式：CSI_OK/CSI_ERROR	USART发送数据两种模式返回数据不一样，具体如下： 轮训模式，返回发送数据长度 中断模式：返回发送成功/失败

21.2.10 csi_usart_receive

*uint16_t csi_usart_receive(csp_usart_t *ptUsartBase, void *pData, uint16_t hwSize, uint32_t wTimeOut)*

21.2.10.1 功能描述

获取USART接收到的数据，USART接收有三种模式(轮训/中断1/中断2)，中断模式1为接收用户指定长度数据；中断模式2为动态接收一串字符，字符长度为动态(不固定)；接收模式初始化时用户可配置。

21.2.10.2 参数/返回值说明

1. 参数

   ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

   pData：指向用户获取数据buffer首地址，即把接收到的数据读取到用户定义的buffer中。

   hwSize：要获取的数据长度，轮训模式和中断模式1时参数有意义；中断模式2忽略此参数。

   wTimeOut：获取USART串口数据超时处理，轮训模式时参数有意义；另外两种模式忽略此参数。

2. 返回值

   获取到的数据长度。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义
pData	void 类型指针，指向发送数据缓存首地址	获取数据缓存类型为uint8_t，调用时需强制转换下(void *)
hwSize	uint16_t 类型数据，要获取数据的长度	用户指定的数据长度
return value	uint16_t 类型数值，获取到的数据长度	实际返回的数据长度

21.2.11 csi_usart_get_msg

*bool csi_usart_get_msg(csp_usart_t *ptUsartBase, csi_usart_wkmode_e eWkMode, bool bClrEn)*

21.2.11.1 功能描述

获取USART接收和发送完成状态，并且清除/保留该状态。

21.2.11.2 参数/返回值说明

1. 参数

ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

eWkMode：USART工作模式，即USART工作于接收还是发送模式，枚举定义详见csi_usart_wkmode_e。

bClrEn：清除/保留发送/接收完成状态。

2. 返回值

bool类型，true/false

3. 参数说明

参数	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义

eWkMode	`typedef enum{`
	`  USART_SEND = 0, //usart send`
	`  USART_RECV = 1, //usart receive`
	`}csi_usart_wkmode_e;`
		USART工作模式：
		USART_SEND：发送
		USART_RECV：接收
		在usart.h中定义

bClrEn	清除/保留USART接收/发送完成状态。	ENABLE：清除状态 DISABLE：保留状态
Return value	接收/发送完成状态返回true，反之，false	

21.2.12 csi_usart_clr_msg

*void csi_usart_clr_msg(csp_usart_t *ptUsartBase, csi_usart_wkmode_e eMode)*

21.2.12.1 功能描述

清除USART接收/发送完成状态，即接收/发送状态设置为空闲。

21.2.12.2 参数/返回值说明

4. 参数

ptUsartBase：USART寄存器结构体指针，指向USART基地址，结构体定义详见csp_usart_t。

eWkMode：USART工作模式，即USART工作于接收还是发送模式，枚举定义详见csi_usart_wkmode_e。

5. 返回值

无返回值。

6. 参数说明

参数	说明	概述及其结构体定义位置
ptUsartBase	csp_usart_t 类型指针，请参阅21.3.1.2参数说明	在csp_usart.h中定义

eWkMode	`typedef enum{`
	`  USART_SEND = 0, //usart send`
	`  USART_RECV = 1, //usart receive`
	`}csi_usart_wkmode_e;`
		USART工作模式：
		USART_SEND：发送
		USART_RECV：接收
		在usart.h中定义

22 TC0

22.1 概述

通用定时器 0 (TC0)是一个 16 位的定时/计数模块，定时器 0 有 3 个输入输出通道，2 种工作模式(捕捉模式和波形发生器模式)，用来实现各种功能，例如频率测量，事件计数，时长测量，脉冲发生，产生延时，脉冲宽度调制(PWM)等。该文档接口说明提供了 TC0 所有模式相关配置和操作

22.2 API列表

Table22-1 TC0 CSI接口函数

API	说明	函数位置
csi_tc0_capture_init	捕获功能初始化	tc0.c
csi_tc0_pwm_init	pwm输出功能初始化	
csi_tc0_clk_enable	TC0模块时钟使能	
csi_tc0_counter_clk_enable	TC0计数时钟使能	
csi_tc0_int_enable	TC0中断使能	
csi_tc0_get_channel_number	获取TC0的通道号	
csi_tc0_set_internal_clksrc	选择TC0的内部时钟源	
csi_tc0_set_external_clksrc	选择TC0的外部时钟源	
csi_tc0_swtrg	TC0软件启动	
csi_tc0_swrst	TC0软件复位	
csi_tc0_set_ra	设置TC0寄存器A的值	
csi_tc0_get_ra	获取TC0寄存器A的值	
csi_tc0_set_rb	设置TC0寄存器B的值	
csi_tc0_get_rb	获取TC0寄存器B的值	
csi_tc0_set_rc	设置TC0寄存器C的值	
csi_tc0_get_rc	获取TC0寄存器C的值	
csi_tc0_get_sr	获取TC0状态寄存器的值	

22.3 API详细说明

22.3.1 csi_tc0_capture_init

*csi_error_t csi_tc0_capture_init(csp_tc0_t *ptTc0Base, csi_tc0_capture_config_t *ptTc0CapCfg)*

22.3.1.1 功能描述

捕获功能初始化配置

22.3.1.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

ptTc0CapCfg：捕获功能配置结构体指针，结构体定义详见csi_tc0_capture_config_t。

2. 返回值

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	TC0有三个通道。 TC0_0,TC0_1,TC0_2 分别指向TC0三个通道对应的基地址。 TC0_0~ TC0_2在devices.c中定义 csp_tc0_t在csp_tc0.h中定义

ptTc0CapCfg	typedef struct {
	uint8_t byIntClkSrc; //tc0 internal clk source select
	uint8_t byExtClkSrc; //tc0 external clk source select
	uint8_t byClkSel; //tc0 clk select,internal clk(div) or external clk
	uint8_t byWorkmod; //capture or output
	uint8_t byClkInvert; //The rising or falling edge count increases by 1
	uint8_t byClkBurst; //burst clk set
	uint8_t byLoadbStopCnt; //when load RB, stop counter
	uint8_t byLoadbStopClk; //when load RB, stop clk
	uint8_t byExtTrgEdge; //external trigger edge
	uint8_t byExtTrgSrc; //external trigger source
	uint8_t byLoadaEventSel; //load RA event select
	uint8_t byLoadbEventSel; //load RB event select
	uint8_t byCpcTrg; //compare with RC,if equal, whether trg start
	uint8_t byInt; //interrupt
	}csi_tc0_capture_config_t;
		该结构体为捕获模式提供了丰富的配置：
		1：内外部时钟源的选择
		2：内外部时钟的选择
		3：工作模式的选择
		4：时钟是否反向
		5：群脉冲配置
		6：装载寄存器RB停止计数器或者时钟
		7：外部触发源和边沿的选择
		8：装载RA,RB的事件选择
		9：RC比较匹配是否触发启动，中断选择。
		详见tc0.h

csi_error_t	csi_error_t 中定义值	在common.h中定义

22.3.2 csi_tc0_pwm_init

*csi_error_t csi_tc0_pwm_init(csp_tc0_t *ptTc0Base, csi_tc0_pwm_config_t *ptTc0PwmCfg)*

22.3.2.1 功能描述

pwm功能初始化配置

22.3.2.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

ptTc0PwmCfg：捕获功能配置结构体指针，结构体定义详见csi_tc0_pwm_config_t。

2. 返回值

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明

ptTc0PwmCfg	typedef struct {
	uint8_t byIntClkSrc; //tc0 internal clk source select
	uint8_t byExtClkSrc; //tc0 external clk source select
	uint8_t byClkSel; //tc0 clk select,internal clk(div) or external clk
	uint8_t byWorkmod; //capture or output
	
	uint8_t byClkInvert; //The rising or falling edge count increases by 1
	uint8_t byClkBurst; //burst clk set
	uint8_t byCpcStopCnt; //compare with RC,if equal, whether stop counter
	uint8_t byCpcStopClk; //compare with RC,if equal, whether stop clk
	
	uint8_t byExtTrgEdge; //external trigger edge
	uint8_t byExtTrgSrc; //external trigger source
	uint8_t byEnetrg; //enable external event Whether to trigger as a reset and restart count
	uint8_t byCpcTrg; //compare with RC,if equal, whether start trg
	
	uint32_t wFreq; //frequence:a,b
	uint8_t byDutyCycleA; //duty cycle:a
	uint8_t byDutyCycleB; //duty cycle:b
	uint8_t byInt; //interrupt
	}csi_tc0_pwm_config_t;
		该结构体为pwm输出提供了丰富的配置：
		1：内外部时钟源的选择
		2：内外部时钟的选择
		3：工作模式的选择
		4：时钟是否反向，群脉冲配置
		5：RC比较匹配是否停止计数器或者时钟
		6：外部触发源和边沿的选择
		7：使能外部事件触发计数器复位和重启
		8：RC比较匹配是否触发启动
		9：频率的设置，占空比的设置，中断选择。
		详见tc0.h

csi_error_t	csi_error_t 中定义值	在common.h中定义

22.3.3 csi_tc0_clk_enable

*void csi_tc0_clk_enable(csp_tc0_t *ptTc0Base, bool bEnable)*

22.3.3.1 功能描述

TC0模块时钟使能,读者可能有疑惑，这个函数和csi_clk_enable不是重复了吗，其实不然，csi_clk_enable这个函数是TC0模块的时钟开关，控制着TC0三个通道的总时钟，而csi_tc0_clk_enable是控制具体某一通道时钟的开关。

22.3.3.2 参数/返回值说明

1. 参数

   ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

   bEnable：使能或者禁止时钟，ENABLE/DISABLE。

2. 返回值

   无返回值。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能 DISABLE：禁止 在common.h中定义

22.3.4 csi_tc0_counter_clk_enable

*void csi_tc0_counter_clk_enable(csp_tc0_t *ptTc0Base, bool bEnable)*

22.3.4.1 功能描述

TC0模块某一通道计数器时钟使能。

22.3.4.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

bEnable：使能或者禁止时钟，ENABLE/DISABLE。

2. 返回值

无返回值。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能 DISABLE：禁止 在common.h中定义

22.3.5 csi_tc0_int_enable

*void csi_tc0_int_enable(csp_tc0_t *ptTc0Base, csi_tc0_intsrc_e eIntSrc, bool bEnable)*

22.3.5.1 功能描述

使能或者禁止相关中断，不同中断源可以相或。

22.3.5.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

eIntSrc：TC0中断源，枚举类型定义详见csi_tc0_intsrc_e

bEnable：使能或者禁止中断，ENABLE/DISABLE;

2. 返回值

无返回值。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明

eIntSrc	typedef enum
	{
	TC0_INTSRC_NONE = (0x00ul << 0), // none
	TC0_INTSRC_COVFS = (0x01ul << 0), // Counter overflow
	TC0_INTSRC_LOVRS = (0x01ul << 1), // Load overrun
	TC0_INTSRC_CPAS = (0x01ul << 2), // Compare Register A
	TC0_INTSRC_CPBS = (0x01ul << 3), // Compare Register B
	TC0_INTSRC_CPCS = (0x01ul << 4), // Compare Register C
	TC0_INTSRC_LDRAS = (0x01ul << 5), // Load Register A
	TC0_INTSRC_LDRBS = (0x01ul << 6), // Load Register B
	TC0_INTSRC_ETRGS = (0x01ul << 7) // External Trigger
	}csi_tc0_intsrc_e;
		支持8个中断：
		1：计数值溢出中断
		2：计数值载入溢出中断
		3：比较寄存器A匹配中断
		4：比较寄存器B匹配中断
		5：比较寄存器C匹配中断
		6：载入捕捉寄存器A中断
		7：载入捕捉寄存器B中断
		8：外部触发中断。
		在tc0.h中定义

bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

22.3.6 csi_tc0_get_channel_number

*uint8_t csi_tc0_get_channel_number(csp_tc0_t *ptTc0Base)*

22.3.6.1 功能描述

根据传入的指针，判断是第几通道

22.3.6.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

2. 返回值

返回TC0当前的通道号。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
return value	uint8_t 类型数值，标识通道号	

22.3.7 csi_tc0_set_internal_clksrc

void csi_tc0_set_internal_clksrc(csi_tc0_internal_clksrc_e eClksrc)

22.3.7.1 功能描述

当时钟选择来自内部时，需要具体选择内部哪一个时钟。

22.3.7.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

eClksrc：内部时钟源选择，枚举类型详见csi_tc0_internal_clksrc_e。

2. 返回值

无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
eClksrc	typedef enum{ TC0_PCLK = 0, TC0_HFOSC_48 }csi_tc0_internal_clksrc_e;	内部时钟源选择： 外设时钟PCLK 内部高速时钟HFOSC

22.3.8 csi_tc0_set_external_clksrc

*void csi_tc0_set_external_clksrc(csp_tc0_t *ptTc0Base,csi_tc0_external_clksrc_e eClksrc)*

22.3.8.1 功能描述

当时钟选择来自外部时，需要具体选择外部哪一个时钟。

22.3.8.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

eClksrc：外部时钟源选择，枚举类型详见csi_tc0_external_clksrc_e。

2. 返回值

无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明

eClksrc	typedef enum{
	//tc0 channel0
	TC0_0_SEL_TCLK0 = 0,
	TC0_0_SEL_TIOA1 = 2,
	TC0_0_SEL_TIOA2 = 3,
	//tc0 channel1
	TC0_1_SEL_TCLK1 = 0,
	TC0_1_SEL_TIOA0 = 2,
	TC0_1_SEL_TIOA2 = 3,
	//tc0 channel2
	TC0_2_SEL_TCLK2 = 0,
	TC0_2_SEL_TIOA0 = 2,
	TC0_2_SEL_TIOA1 = 3
	}csi_tc0_external_clksrc_e;
		该枚举类型集成了通道0到通道2的内部时钟源选择。使用时根据实际需要选择

22.3.9 csi_tc0_swtrg

*void csi_tc0_swtrg(csp_tc0_t *ptTc0Base)*

22.3.9.1 功能描述

软件触发启动计数器计数。

22.3.9.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

2. 返回值

无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明

22.3.10 csi_tc0_swrst

*void csi_tc0_swrst(csp_tc0_t *ptTc0Base)*

22.3.10.1功能描述

软件复位定时器。

22.3.10.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

2. 返回值

无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明

22.3.11 csi_tc0_set_ra

*void csi_tc0_set_ra(csp_tc0_t *ptTc0Base,uint16_t hwData)*

22.3.11.1功能描述

往RA寄存器写入数据。

22.3.11.2 参数/返回值说明

1. 参数

   ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

   hwData：需要写入的数据

2. 返回值

   无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
hwData	uint16_t 类型的数据	写入的值

22.3.12 csi_tc0_get_ra

*uint16_t csi_tc0_get_ra(csp_tc0_t *ptTc0Base)*

22.3.12.1功能描述

读取RA寄存器的数据。

22.3.12.2 参数/返回值说明

1. 参数

   ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

2. 返回值

   返回RA寄存器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
return value	uint16_t 类型的数据	返回值

22.3.13 csi_tc0_set_rb

*void csi_tc0_set_rb(csp_tc0_t *ptTc0Base,uint16_t hwData)*

22.3.13.1功能描述

往RB寄存器写入数据。

22.3.13.2 参数/返回值说明

1. 参数

   ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

   hwData：需要写入的数据

2. 返回值

   无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
hwData	uint16_t 类型的数据	写入的值

22.3.14 csi_tc0_get_rb

*uint16_t csi_tc0_get_rb(csp_tc0_t *ptTc0Base)*

22.3.14.1功能描述

读取RB寄存器的数据。

22.3.14.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

2. 返回值

返回RB寄存器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
return value	uint16_t 类型的数据	返回值

22.3.15 csi_tc0_set_rc

*void csi_tc0_set_rc(csp_tc0_t *ptTc0Base,uint16_t hwData)*

22.3.15.1功能描述

往RC寄存器写入数据。

22.3.15.2 参数/返回值说明

1. 参数

   ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

   hwData：需要写入的数据

2. 返回值

   无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
hwData	uint16_t 类型的数据	写入的值

22.3.16 csi_tc0_get_rc

*uint16_t csi_tc0_get_rc(csp_tc0_t *ptTc0Base)*

22.3.16.1功能描述

读取RC寄存器的数据。

22.3.16.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

2. 返回值

返回RC寄存器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
return value	uint16_t 类型的数据	返回值

22.3.17 csi_tc0_get_sr

*uint32_t csi_tc0_get_sr(csp_tc0_t *ptTc0Base)*

22.3.17.1功能描述

读取SR寄存器的数据，用于获取状态。

22.3.17.2 参数/返回值说明

1. 参数

ptTc0Base：TC0寄存器结构体指针，指向TC0基地址，结构体定义详见csp_tc0_t。

2. 返回值

返回SR寄存器的值，包含所有状态值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc0Base	csp_tc0_t 类型指针，指向TC0的基地址	请参阅22.3.1.2参数说明
return value	uint32_t 类型的数据	返回值

23 TC1

23.1 概述

通用定时器 1 (TC1)是一个 32 位的定时/计数模块，TC1 有三种工作模式：比较匹配模式，输入捕捉模式，输出翻转或者称为 PWM 模式。TC1 可以通过特定的管脚输出 PWM 信号，并且 TC1 的时钟源支持从外部引入。

该文档接口说明提供了 TC1 所有模式相关配置和操作。

23.2 API列表

Table23-1 TC1 CSI接口函数

API	说明	函数位置
csi_tc1_capture_init	捕获功能初始化	tc1.c
csi_tc1_pwm_init	pwm输出功能初始化	
csi_tc1_clk_enable	时钟使能	
csi_tc1_int_enable	中断使能	
csi_tc1_swrst	软件复位	
csi_tc1_get_sr	获取状态寄存器	
csi_tc1_set_cnt_size	设置计数器的位数	
csi_tc1_set_prdr	设置周期值	
csi_tc1_get_prdr	获取周期值	
csi_tc1_set_pulr	设置脉冲匹配值	
csi_tc1_get_pulr	获取脉冲匹配值	
csi_tc1_get_cucr	获取上升沿捕捉寄存器的值	
csi_tc1_get_cdcr	获取下降沿捕捉寄存器的值	
csi_tc1_get_cvr	获取当前计数器的值	
csi_tc1_start	启动计数器	
csi_tc1_stop	停止计数器	
csi_tc1_get_clk_divn	获取时钟的分频值n	
csi_tc1_get_clk_divm	获取时钟的分频值m	

23.3 API详细说明

23.3.1 csi_tc1_capture_init

*csi_error_t csi_tc1_capture_init(csp_tc1_t *ptTc1Base, csi_tc1_capture_config_t *ptTc1CapCfg)*

23.3.1.1 功能描述

捕获功能初始化配置，当选择捕获功能的时候，时钟只能选择PCLK。

23.3.1.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

   ptTc1CapCfg：捕获功能配置结构体指针，结构体定义详见csi_tc1_capture_config_t。

2. 返回值

   CSI_OK：初始化成功。

   CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	TC1在devices.c中定义 csp_tc1_t在csp_tc1.h中定义

ptTc0CapCfg	`typedef struct {`
	`  uint8_t byCapSrc;  //tc1 cap source select`
	`  uint8_t byInt;  //interrupt`
	`}csi_tc1_capture_config_t;`
		该结构体可以配置：
		1：选项需要捕获的源
		2：使能对应中断源
		详见tc1.h

csi_error_t	csi_error_t 中定义值	在common.h中定义

23.3.2 csi_tc1_pwm_init

*csi_error_t csi_tc1_pwm_init(csp_tc1_t *ptTc1Base, csi_tc1_pwm_config_t *ptTc1PwmCfg)*

23.3.2.1 功能描述

pwm功能初始化配置，这里注意空闲电平的配置，细心的读者会发现这里有6种配置，这是因为空闲电平与定时器的停止方式有关，也就意味着byIdleLevel这个参数有两层含义，停止方式和空闲电平。使用时请选择符合你应用的配置。

23.3.2.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

   ptTc1PwmCfg：捕获功能配置结构体指针，结构体定义详见csi_tc1_pwm_config_t。

2. 返回值

   CSI_OK：初始化成功。

   CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明

ptTc0PwmCfg	typedef struct {
	uint8_t byIdleLevel; //TC1 pwm idel level
	uint8_t byStartLevel; //TC1 pwm start level
	uint8_t byInt; //TC1 pwm interrupt source
	uint8_t byDutyCycle; //TC1 pwm duty cycle
	uint8_t byClkSrc; //TC1 clock source
	uint32_t wFreq; //TC1 pwm frequency
	}csi_tc1_pwm_config_t;
		该结构体为pwm输出提供了丰富的配置：
		1：空闲电平的配置
		2：起始电平的配置
		3：中断源的配置
		4：占空比的配置
		5：时钟源的选择
		6：pwm频率的设置
		详见tc1.h

csi_error_t	csi_error_t 中定义值	在common.h中定义

23.3.3 csi_tc1_clk_enable

*void csi_tc1_clk_enable(csp_tc1_t *ptTc1Base, bool bEnable)*

23.3.3.1 功能描述

TC1模块时钟使能。

23.3.3.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

bEnable：使能或者禁止时钟，ENABLE/DISABLE。

2. 返回值

无返回值。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能 DISABLE：禁止 在common.h中定义

23.3.4 csi_tc1_int_enable

*void csi_tc1_int_enable(csp_tc1_t *ptTc1Base, csi_tc1_intsrc_e eIntSrc, bool bEnable)*

23.3.4.1 功能描述

使能或者禁止相关中断，不同中断源可以相或。

23.3.4.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

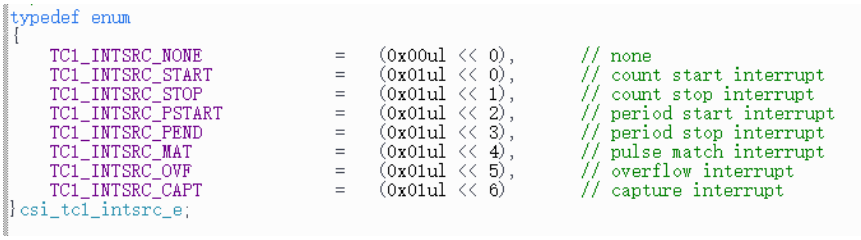
eIntSrc：TC1中断源，枚举类型定义详见csi_tc1_intsrc_e

bEnable：使能或者禁止中断，ENABLE/DISABLE;

2. 返回值

无返回值。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
eIntSrc		支持7个中断： 1：计数启动中断 2：计数停止中断 3：周期开始中断 4：周期结束中断 5：脉冲匹配中断 6：溢出中断 7：捕捉中断 在tc1.h中定义
bEnable	Bool 类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

23.3.5 csi_tc1_swrst

*void csi_tc1_swrst(csp_tc1_t *ptTc1Base)*

23.3.5.1 功能描述

软件复位定时器。

23.3.5.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

无返回值。

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明

23.3.6 csi_tc1_get_sr

*uint32_t csi_tc1_get_sr(csp_tc1_t *ptTc1Base)*

23.3.6.1 功能描述

读取SR寄存器的数据，用于获取状态。

23.3.6.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

返回SR寄存器的值，包含所有状态值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint32_t 类型的数据	返回值

23.3.7 csi_tc1_set_cnt_size

*void csi_tc1_set_cnt_size(csp_tc1_t *ptTc1Base,uint8_t bySize)*

23.3.7.1 功能描述

设置tc1的size大小。

23.3.7.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

bySize：要设置的计数器位数，虽然有八位，但是只会写入第五位，这在程序上会自动处理。

2. 返回值

无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
bySize	uint8_t 类型的数据	写入的值

23.3.8 csi_tc1_set_prdr

*void csi_tc1_set_prdr(csp_tc1_t *ptTc1Base,uint32_t wData)*

23.3.8.1 功能描述

设置周期寄存器的值。

23.3.8.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

   wData：要写入的数据。

2. 返回值

   无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
wData	uint32_t 类型的数据	写入的值

23.3.9 csi_tc1_get_prdr

*uint32_t csi_tc1_get_prdr(csp_tc1_t *ptTc1Base)*

23.3.9.1 功能描述

读取周期寄存器的值。

23.3.9.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

返回周期寄存器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint32_t 类型的数据	返回值

23.3.10 csi_tc1_set_pulr

*void csi_tc1_set_pulr(csp_tc1_t *ptTc1Base,uint32_t wData)*

23.3.10.1功能描述

设置脉冲匹配寄存器的值。

23.3.10.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

   wData：需要写入的数据

2. 返回值

   无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
wData	uint32_t 类型的数据	写入的值

23.3.11 csi_tc1_get_pulr

*uint32_t csi_tc1_get_pulr(csp_tc1_t *ptTc1Base)*

23.3.11.1功能描述

读取脉冲匹配寄存器的值。

23.3.11.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

返回脉冲匹配寄存器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint32_t 类型的数据	返回值

23.3.12 csi_tc1_get_cvr

*uint32_t csi_tc1_get_cvr(csp_tc1_t *ptTc1Base)*

23.3.12.1功能描述

读取当前计数器的值。

23.3.12.2 参数/返回值说明

1. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

返回当前计数器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint32_t 类型的数据	返回值

23.3.13 csi_tc1_get_cucr

*uint32_t csi_tc1_get_cucr(csp_tc1_t *ptTc1Base)*

23.3.13.1功能描述

读取上升沿事件发生时所捕获的计数值

23.3.13.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

   返回当前计数器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint32_t 类型的数据	返回值

23.3.14 csi_tc1_get_cdcr

*uint32_t csi_tc1_get_cdcr(csp_tc1_t *ptTc1Base)*

23.3.14.1功能描述

读取下降沿事件发生时所捕获的计数值

23.3.14.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

   返回当前计数器的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint32_t 类型的数据	返回值

23.3.15 csi_tc1_start

*void csi_tc1_start(csp_tc1_t *ptTc1Base)*

23.3.15.1功能描述

启动计数器

23.3.15.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

   无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明

23.3.16 csi_tc1_stop

*void csi_tc1_stop(csp_tc1_t *ptTc1Base)*

23.3.16.1功能描述

停止计数器

23.3.16.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

   无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明

23.3.17 csi_tc1_get_clk_divn

*uint8_t csi_tc1_get_clk_divn(csp_tc1_t *ptTc1Base)*

23.3.17.1功能描述

获取tc1时钟分频系数n，该分频系数只有4bit宽度

23.3.17.2 参数/返回值说明

1. 参数

   ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

2. 返回值

   返回分频系数n的值

3. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint8_t 类型的数据	返回值

23.3.18 csi_tc1_get_clk_divm

*uint8_t csi_tc1_get_clk_divm(csp_tc1_t *ptTc1Base)*

23.3.18.1功能描述

获取tc1时钟分频系数m，该分频系数只有8bit宽度

23.3.18.2 参数/返回值说明

4. 参数

ptTc1Base：TC1寄存器结构体指针，指向TC1基地址，结构体定义详见csp_tc1_t。

5. 返回值

返回分频系数n的值

6. 参数/返回值说明

参数/返回值	说明	概述及其结构体定义位置
ptTc1Base	csp_tc1_t 类型指针，指向TC1的基地址	请参阅23.3.1.2参数说明
return value	uint8_t 类型的数据	返回值

24 TC2

24.1 概述

TC2 是一个简易定时器，定时器(以下称为 STC)有 2 个功能通道，每个通道可以分别工作在两种模式，定时匹配模式和输入信号捕捉模式。

ATP CSI 接口 TC2 的设计中，提供丰富的配置及其操作。包含基本定时配置功能、通道配置功能等

24.2 API列表

Table 24-1 TC2 CSI接口函数

csi_tc2_start	开始TC2功能	tc2.c
csi_tc2_stop	停止TC2功能	
csi_tc2_int_enable	使能TC2中断	
csi_tc2_get_misr	获取中断状态	
csi_tc2_int_clear	清除中断状态	
csi_tc2_channel_int_enable	使能TC2通道中断	
csi_tc2_get_channel_misr	获取TC2通道中断状态	
csi_tc2_channel_int_clear	清除TC2通道中断状态	
csi_tc2_init	TC2基本功能初始化	
csi_tc2_channel_config	TC2通道配置功能	
csi_tc2_ get_capture0_cmp	获取通道0捕获值	
csi_tc2_get_capture1_cmp	获取通道1捕获值	

24.3 API详细说明

24.3.1 csi_tc2_start

*void csi_tc2_start(csp_tc2_t *ptTc2Base)*

24.3.1.1 功能描述

开始TC2功能

24.3.1.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

2. 返回值：无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义

24.3.2 csi_tc2_stop

*void csi_tc2_stop (csp_tc2_t *ptTc2Base)*

24.3.2.1 功能描述

停止TC2功能

24.3.2.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

2. 返回值：无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型

		csp_tc2_t在csp_tc2.h中定义

24.3.3 csi_tc2_int_enable

*void csi_tc2_int_enable(csp_tc2_t *ptTc2Base, uint8_t byIntSrc,bool bEnable)*

24.3.3.1 功能描述

使能TC2中断

24.3.3.2 参数/返回值说明

1. 参数

   ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

   byIntSrc：中断类型。

   bEnable：启用中断或禁止中断。

2. 返回值：无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义

byIntSrc	中断类型，包含以下几种，可进行相应的组合，参数范围为0 ~ 7
	`typedef enum`
	`{`
	`  TC2_INTSRC_NONE  = (0x00ul),`
	`  TC2_INTSRC_START  = (0x01ul << 0),`
	`  TC2_INTSRC_STOP  = (0x01ul << 1),`
	`  TC2_INTSRC_PEND  = (0x01ul << 2),`
	`}csi_tc2_int_e;`
		TC2的中断类型
		csi_tc2_int_e在tc2.h中定义

bEnable	bool类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

24.3.4 csi_tc2_get_misr

*uint8_t csi_tc2_get_misr(csp_tc2_t *ptTc2Base)*

24.3.4.1 功能描述

获取中断状态

24.3.4.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

2. 返回值：uint8_t类型的返回数据，返回的为中断状态。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义
return value	uint8_t类型的数据，返回中断状态,每位表示不同的中断。 0：中断未发生 1：中断发生	返回中断状态

24.3.5 csi_tc2_int_clear

*void csi_tc2_int_clear(csp_tc2_t *ptTc2Base, csi_tc2_int_e eTc2Int)*

24.3.5.1 功能描述

清除中断状态

24.3.5.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

eTc2Int：中断类型。

2. 返回值：无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义
eTc2Int	中断类型，主要包含以下几种：	TC2的中断类型 csi_tc2_int_e在tc2.h中定义

	typedef enum { TC2_INTSRC_NONE = (0x00ul), TC2_INTSRC_START = (0x01ul << 0), TC2_INTSRC_STOP = (0x01ul << 1), TC2_INTSRC_PEND = (0x01ul << 2), }csi_tc2_int_e;	

24.3.6 csi_tc2_channel_int_enable

*void csi_tc2_channel_int_enable(csp_tc2_t *ptTc2Base, uint16_t hwChannelIntSrc,bool bEnable)*

24.3.6.1 功能描述

使能TC2通道中断

24.3.6.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

hwChannelIntSrc：中断类型。

bEnable：启用中断或禁止中断。

2. 返回值：无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义
hwChannelIntSrc	中断类型，主要包含以下几种，可进行相应的组合。 typedef enum { TC2_CHANNEL_INTSRC_NONE = (0x00ul), TC2_CHANNEL0_INTSRC_RISE = (0x01ul << 0), TC2_CHANNEL1_INTSRC_RISE = (0x01ul << 1), TC2_CHANNEL0_MATCH = (0x01ul << 0), TC2_CHANNEL1_MATCH = (0x01ul << 1), TC2_CHANNEL0_INTSRC_FALL = (0x01ul << 8), TC2_CHANNEL1_INTSRC_FALL = (0x01ul << 9) }csi_tc2_channel_int_e;	TC2的中断类型 csi_tc2_int_e在tc2.h中定义
bEnable	bool类型数值，ENBALE/DISABLE	ENBALE：使能中断 DISABLE：禁止中断 在common.h中定义

24.3.7 csi_tc2_get_channel_misr

*uint16_t csi_tc2_get_channel_misr(csp_tc2_t *ptTc2Base)*

24.3.7.1 功能描述

获取TC2通道中断状态

24.3.7.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

2. 返回值：uint16_t类型的返回数据，返回的为中断状态。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义
return value	uint16_t类型的数据，返回中断状态,每位表示不同的中断。 0：中断未发生 1：中断发生	返回中断状态

24.3.8 csi_tc2_channel_int_clear

*void csi_tc2_channel_int_clear(csp_tc2_t *ptTc2Base, csi_tc2_channel_int_e eTc2ChannelInt)*

24.3.8.1 功能描述

清除TC2通道中断状态

24.3.8.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

eTc2ChannelInt：中断类型。

2. 返回值：无返回值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置

ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义
hwChannelIntSrc	中断类型，主要包含以下几种： typedef enum { TC2_CHANNEL_INTSRC_NONE = (0x00ul), TC2_CHANNEL0_INTSRC_RISE = (0x01ul << 0), TC2_CHANNEL1_INTSRC_RISE = (0x01ul << 1), TC2_CHANNEL0_MATCH = (0x01ul << 0), TC2_CHANNEL1_MATCH = (0x01ul << 1), TC2_CHANNEL0_INTSRC_FALL = (0x01ul << 8), TC2_CHANNEL1_INTSRC_FALL = (0x01ul << 9) }csi_tc2_channel_int_e;	TC2的中断类型 csi_tc2_int_e在tc2.h中定义

24.3.9 csi_tc2_init

*csi_error_t csi_tc2_init(csp_tc2_t *ptTc2Base,csi_tc2_config_t *ptTc2Cfg)*

24.3.9.1 功能描述

TC2基本功能初始化

24.3.9.2 参数/返回值说明

1. 参数

   ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

   ptTc2Cfg：TC2基本配置结构体指针，结构体定义详见csi_tc2_config_t。

2. 返回值：

   CSI_OK：初始化成功。

   CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义

ptTc2Cfg	typedef struct { uint8_t bySingle; uint8_t byStopType; uint8_t byDivn; uint8_t byInt; uint16_t hwDivm; uint16_t hwCnt; }csi_tc2_config_t;	在tc2.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

24.3.10 csi_tc2_channel_config

*csi_error_t csi_tc2_channel_config(csp_tc2_t *ptTc2Base,csi_tc2_channel_config_t *ptTc2ChanCfg,uint8_t byChannel)*

24.3.10.1 功能描述

TC2通道配置功能

24.3.10.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

ptTc2ChanCfg：TC2基本配置结构体指针，结构体定义详见csi_tc2_channel_config_t。

byChannel:通道选择，包含TC2_CHANNEL_IDX0和TC2_CHANNEL_IDX1。

2. 返回值：

CSI_OK：初始化成功。

CSI_ERROR：初始化失败。

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型csp_tc2_t在csp_tc2.h中定义

ptTc2ChanCfg	typedef struct { uint8_t byCmMode; uint8_t byCaptureMode; bool bCaptureEnable; bool bChannelEnable; uint16_t hwInt; uint16_t hwCmp; }csi_tc2_channel_config_t;	在tc2.h中定义
byChannel	通道选择，包含TC2_CHANNEL_IDX0和TC2_CHANNEL_IDX1	通道选择
csi_error_t	csi_error_t 中定义值	在common.h中定义

24.3.11 csi_tc2_get_capture0_cmp

*uint16_t csi_tc2_get_capture0_cmp(csp_tc2_t *ptTc2Base)*

24.3.11.1 功能描述

获取通道0捕获值

24.3.11.2 参数/返回值说明

1. 参数

ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

2. 返回值：uint16_t类型的返回数据，返回的为通道0捕获值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义
return value	uint16_t类型的返回数据，返回的为通道0捕获值。	返回通道0捕获值

24.3.12 csi_tc2_get_capture1_cmp

*uint16_t csi_tc2_ get_capture1_cmp(csp_tc2_t *ptTc2Base)*

24.3.12.1 功能描述

获取通道1捕获值

24.3.12.2 参数/返回值说明

1. 参数

   ptTc2Base：TC2寄存器结构体指针，指向TC2基地址，结构体定义详见csp_tc2_t。

2. 返回值：uint16_t类型的返回数据，返回的为通道1捕获值

3. 参数/返回值说明

参数/返回值	说明	概述及其枚举/结构体定义位置
ptTc2Base	csp_tc2_t 类型指针，指向tc2的基地址	TC2寄存器结构体指针类型 csp_tc2_t在csp_tc2.h中定义
return value	uint16_t类型的返回数据，返回的为通道1捕获值。	返回通道1捕获值

25 EPWM

25.1 概述

此 MCU 内嵌了一个 16 位的 EPWM 模块。该 PWM 模块包含了 3 组 6 个独立的 PWM 通道，每组可以输出两路独立的，或者互补的 PWM 信号，该 PWM 模块还支持和模拟比较器的协同工作用以支持针对电磁加热以及电机的专门应用。

25.2 API列表

Table 25-1 EPWM CSI接口函数

API	说明	函数位置
epwm_irqhandler	定时器中断回调函数	
csi_epwm_wave_init	定时器波形模式参数基础设置（计数模式，周期，占空比等）	
csi_epwm_channel_config	设置通道PWM1、PWM2、PWM3的波形	
csi_epwm_channelmode_config	死区通道配置	
csi_epwm_Chopper_config	斩波输出控制	
csi_epwm_emergency_config	紧急状态输入参数设置	
csi_epwm_int_enable	中断使能控制器	
csi_epwm_evtrg_enable	事件触发控制设置	
csi_epwm_change_camp	比较值更新（赋值）	

25.3 API详细说明

25.3.1 epwm_irqhandler

*__attribute__((weak)) void epwm_irqhandler(csp_epwm_t *ptEpwmBase)*

25.3.1.1 功能描述

定时器中断回调函数。

25.3.1.2 参数/返回值说明

1. 参数

   ptEpwmBase： 每个功能模块都有对应的ptEpwmBase，枚举定义详见csp_epwm_t

2. 返回值

   无

3. 参数说明表

参数	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t结构体，用于进行寄存器操作。	定时器模块所有寄存器的定义 在csp_ epwm.h中定义

25.3.2 csi_epwm_wave_init

*csi_error_t csi_epwm_wave_init(csp_epwm_t *ptEpwmBase, csi_epwm_pwmconfig_t *ptEpwmPwmCfg)*

25.3.2.1 功能描述

定时器波形模式参数基础设置（计数模式，周期，占空比等）

25.3.2.2 参数/返回值说明

1. 参数

   ptEpwmBase： 定义详见csp_epwm_t

   ptEpwmPwmCfg： 定义详见csi_epwm_pwmconfig_t

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义

ptEpwmPwmCfg	typedef struct {
	uint8_t byWorkmod; //SINGLE
	uint8_t byCountingMode; //csi_ept_cntmd_e
	uint8_t byOneshotMode; //Single or continuous
	uint8_t byDutyCycle; //TIMER PWM OUTPUT duty cycle
	uint32_t wFreq0; //TIMER PWM OUTPUT frequency
	uint32_t wFreq1;
	uint32_t wFreq2;
	uint32_t wInt;
	}csi_epwm_pwmconfig_t;
		在epwm.h中定义

csi_error_t	csi_error_t 中定义值	在common.h中定义

25.3.3 csi_epwm_channel_config

*csi_error_t csi_epwm_channel_config(csp_epwm_t *ptEpwmBase, csi_epwm_pwmchannel_config_t *tPwmCfg, csi_epwm_channel_e echannel))*

25.3.3.1 功能描述

设置通道 PWM1、PWM2、PWM3 的波形

25.3.3.2 参数/返回值说明

1. 参数

   ptEpwmBase： 定义详见csp_epwm_t

   tPwmCfg： 定义详见csi_epwm_pwmchannel_config_t

   echannel: 定义详见 csi_epwm_channel_e

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义
tPwmCfg	typedef struct { uint8_t byActionS_X; uint8_t byActionP_X; uint8_t byActionC_X; uint8_t byChoiceB_X; uint8_t byChoiceA_X; uint8_t byActionS_Y; uint8_t byActionP_Y; uint8_t byActionC_Y; uint8_t byChoiceB_Y; uint8_t byChoiceA_Y; }csi_epwm_pwmchannel_config_t;	在epwm.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

25.3.4 csi_epwm_channelmode_config

*csi_error_t csi_epwm_channelmode_config(csp_epwm_t *ptEpwmBase,csi_epwm_deadzone_config_t *tCfg,csi_epwm_channel_e eChannel)*

25.3.4.1 功能描述

死区通道配置

25.3.4.2 参数/返回值说明

1. 参数

   ptEpwmBase： 定义详见csp_epwm_t

   tCfg： 定义详见csi_epwm_deadzone_config_t

   echannel: 定义详见 csi_epwm_channel_e

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义
tCfg	typedef struct { uint8_t byOutsel; uint8_t bySrcsel; uint16_t hwRisingEdgereGister; uint16_t hwFallingEdgereGister; uint8_t hwXpolarity; uint8_t hwYpolarity; }csi_epwm_deadzone_config_t;	在epwm.h中定义
echannel	typedef enum{ EPWM_CHANNEL_0=0, EPWM_CHANNEL_1, EPWM_CHANNEL_2 }csi_epwm_channel_e;	在epwm.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

25.3.5 csi_epwm_Chopper_config

*csi_error_t csi_epwm_Chopper_config(csp_epwm_t *ptEpwmBase, csi_epwm_Chopper_config_t *tCfg ,csi_epwm_channel_e eChannel)*

25.3.5.1 功能描述

斩波输出控制

25.3.5.2 参数/返回值说明

1. 参数

   ptEpwmBase： 定义详见csp_epwm_t

   tCfg： 定义详见csi_epwm_Chopper_config_t

   echannel: 定义详见 csi_epwm_channel_e

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义
tCfg	typedef struct { uint8_t byCfen; uint8_t byOsw; uint8_t byCdiv; uint8_t byDuty; }csi_epwm_Chopper_config_t;	在epwm.h中定义
echannel	typedef enum{ EPWM_CHANNEL_0=0, EPWM_CHANNEL_1, EPWM_CHANNEL_2 }csi_epwm_channel_e;	在epwm.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

25.3.6 csi_epwm_emergency_config

*csi_error_t csi_epwm_emergency_config(csp_epwm_t *ptEpwmBase, csi_epwm_emergency_config_t *tCfg)*

25.3.6.1 功能描述

紧急状态输入参数设置

25.3.6.2 参数/返回值说明

1. 参数

   ptEpwmBase： 定义详见csp_epwm_t

tCfg： 定义详见csi_epwm_emergency_config_t

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义
tCfg	typedef struct { uint8_t EP4LKM; uint8_t EP3LKM; uint8_t EP2LKM; uint8_t EP1LKM; uint8_t EP0LKM; uint8_t CMP4LKM; uint8_t CMP3LKM; uint8_t CMP2LKM; uint8_t CMP1LKM; uint8_t CMP0LKM; uint8_t TRGTDL; uint8_t TRGIVT; uint16_t INC_STEP0; uint16_t INC_STEP1; uint16_t DEC_STEP0; uint16_t DEC_STEP1; uint8_t SL_P2YS; uint8_t SL_P2XS; uint8_t SL_P1YS; uint8_t SL_P1XS; uint8_t SL_P0YS; uint8_t SL_P0XS; uint8_t HL_P2YS; uint8_t HL_P2XS; uint8_t HL_P1YS; uint8_t HL_P1XS; uint8_t HL_P0YS; uint8_t HL_P0XS; bool S_ONE; bool SL_CNTR_INC_EN; bool SL_CNTR_DEC_EN; bool SL_DECA_EN0; bool SL_INCA_EN0; bool SL_DECB_EN0; bool SL_INCB_EN0; bool SL_DECA_EN1; bool SL_INCA_EN1; bool SL_DECB_EN1; bool SL_INCB_EN1; }csi_epwm_emergency_config_t;	在epwm.h中定义
csi_error_t	csi_error_t 中定义值	在common.h中定义

25.3.7 csi_epwm_int_enable

*csi_error_t csi_epwm_int_enable(csp_epwm_t *ptEpwmBase, csp_epwm_int_e eInt, bool bEnable)*

25.3.7.1 功能描述

中断使能控制器

25.3.7.2 参数/返回值说明

1. 参数

ptEpwmBase： 定义详见csp_epwm_t

eInt： 定义详见csp_epwm_int_e

bEnable: ENABLE/DISABLE

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义
eInt	typedef enum{ EPWM_INT_START0 = 0x1, EPWM_INT_STOP0 = 0x1 <<1, EPWM_INT_PEND0 = 0x1 <<2, EPWM_INT_CENTER0 = 0x1 <<3, EPWM_INT_START1 = 0x1 <<4, EPWM_INT_STOP1 = 0x1 <<5, EPWM_INT_PEND1 = 0x1 <<6, EPWM_INT_CENTER1 = 0x1 <<7, EPWM_INT_START2 = 0x1 <<8, EPWM_INT_STOP2 = 0x1 <<9, EPWM_INT_PEND2 = 0x1 <<10, EPWM_INT_CENTER2 = 0x1 <<11, EPWM_INT_CMPAU0 = 0x1 <<12, EPWM_INT_CMPAD0 = 0x1 <<13, EPWM_INT_CMPBU0 = 0x1 <<14, EPWM_INT_CMPBD0 = 0x1 <<15, EPWM_INT_CMPAU1 = 0x1 <<16, EPWM_INT_CMPAD1 = 0x1 <<17, EPWM_INT_CMPBU1 = 0x1 <<18, EPWM_INT_CMPBD1 = 0x1 <<19, EPWM_INT_CMPAU2 = 0x1 <<20, EPWM_INT_CMPAD2 = 0x1 <<21, EPWM_INT_CMPBU2 = 0x1 <<22, EPWM_INT_CMPBD2 = 0x1 <<23, EPWM_INT_SLPA_0_OVF = 0x1 <<24, EPWM_INT_SLPB_0_OVF = 0x1 <<25, EPWM_INT_SLPA_1_OVF = 0x1 <<26, EPWM_INT_SLPB_1_OVF = 0x1 <<27 }csp_epwm_int_e;	在epwm.h中定义
bEnable	ENABLE/DISABLE	
csi_error_t	csi_error_t 中定义值	在common.h中定义

25.3.8 csi_epwm_evtrg_enable

*csi_error_t csi_epwm_evtrg_enable(csp_epwm_t *ptEpwmBase, csi_epwm_evtrg_config_e byCh, csp_epwm_trg_e byVal)*

25.3.8.1 功能描述

事件触发控制设置

25.3.8.2 参数/返回值说明

1. 参数

ptEpwmBase： 定义详见csp_epwm_t

byCh： 定义详见csi_epwm_evtrg_config_e

byVal: 定义详见csp_epwm_trg_e

2. 返回值

CSI_OK：设置成功

CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义
byCh	typedef enum { EVTRG_PWM0_START=0, EVTRG_PWM0_STOP, EVTRG_PWM0_PEND, EVTRG_PWM0_CENTER, EVTRG_PWM1_START, EVTRG_PWM1_STOP, EVTRG_PWM1_PEND, EVTRG_PWM1_CENTER, EVTRG_PWM2_START, EVTRG_PWM2_STOP, EVTRG_PWM2_PEND, EVTRG_PWM2_CENTER, EVTRG_PWM0_CMPAUM, EVTRG_PWM0_CMPADM, EVTRG_PWM0_CMPBUM, EVTRG_PWM0_CMPBDM, EVTRG_PWM1_CMPAUM, EVTRG_PWM1_CMPADM, EVTRG_PWM1_CMPBUM, EVTRG_PWM1_CMPBDM, EVTRG_PWM2_CMPAUM, EVTRG_PWM2_CMPADM, EVTRG_PWM2_CMPBUM, EVTRG_PWM2_CMPBDM }csi_epwm_evtrg_config_e;	在epwm.h中定义
byVal	typedef enum { EPWM_TRG_DIS =0, EPWM_TRG_ADC, EPWM_TRG_STIMER, EPWM_TRG_ADC_STIMER }csp_epwm_trg_e;	
csi_error_t	csi_error_t 中定义值	在common.h中定义

25.3.9 csi_epwm_change_camp

*csi_error_t csi_epwm_change_camp(csp_epwm_t *ptEpwmBase, csi_epwm_camp_e eCh, uint16_t wActiveTime)*

25.3.9.1 功能描述

比较值更新（赋值）

25.3.9.2 参数/返回值说明

1. 参数

   ptEpwmBase： 定义详见csp_epwm_t

   eCh： 定义详见csi_epwm_camp_e

   wActiveTime: 0~周期值

2. 返回值

   CSI_OK：设置成功

   CSI_ERROR：设置失败。

3. 参数/返回值说明表

参数/返回值	说明	概述及其变量位置
ptEpwmBase	csp_epwm_t中说明	在csp_epwm.h中定义
eCh	typedef enum{ EPWM_CAMPA0=0, EPWM_CAMPA1, EPWM_CAMPA2, EPWM_CAMPB0, EPWM_CAMPB1, EPWM_CAMPB2 }csi_epwm_camp_e;	在epwm.h中定义
wActiveTime	0~周期值	
csi_error_t	csi_error_t 中定义值	在common.h中定义