



芯海科技  
CHIPSEA

# CSU32M10 应用笔记

REV 1.0

芯海科技（深圳）股份有限公司

地 址：深圳市南山区蛇口南海大道1079号花园城数码大厦A座9楼

电 话：+(86 755)86169257      传 真：+(86 755)86169057

网 站：www.chipsea.com      邮 编：518067

微信号：芯海科技



## 版本历史

历史版本	修改内容	版本日期
Rev 1.0	首版发布	2019.5.16

## 目 录

版本历史.....	2
目 录.....	3
1 使用 CSU32M10 替换 CSU32P20 方法说明 .....	4
2 芯片应用 .....	5
2.1 模拟比较器应用问题 .....	5
2.2 恒流源.....	5
2.3 EEPROM.....	6
2.4 芯片休眠电流过大问题 .....	9
2.5 使用模拟比较器进行短路保护注意事项 .....	12
3 IDE 与 CS-LINK 使用注意事项.....	13

## 1 使用 CSU32M10 替换 CSU32P20 方法说明

使用 IDE\_5.4.0 版本 IDE 打开以前 CSU32P20 源程序，芯片型号修改为 CSU32M10,会重新生成对应芯片的头文件，汇编工程将头文件替换为 CSU32M10.inc，C 语言工程将原有程序头文件替换为 CSU32M10.h。若原 SysRegDefine.c 文件在该工程的根目录下，芯片型号修改后会自动将原 SysRegDefine.c 文件替换，若原 SysRegDefine.c 在自定义目录下，芯片修改后会在工程根目录下生成新的 SysRegDefine.c 文件，需要用户自行替换。

注意：芯片更换完毕后代码选项恢复为该芯片的默认值，开发者可根据需求重新配置。

## 2 芯片应用

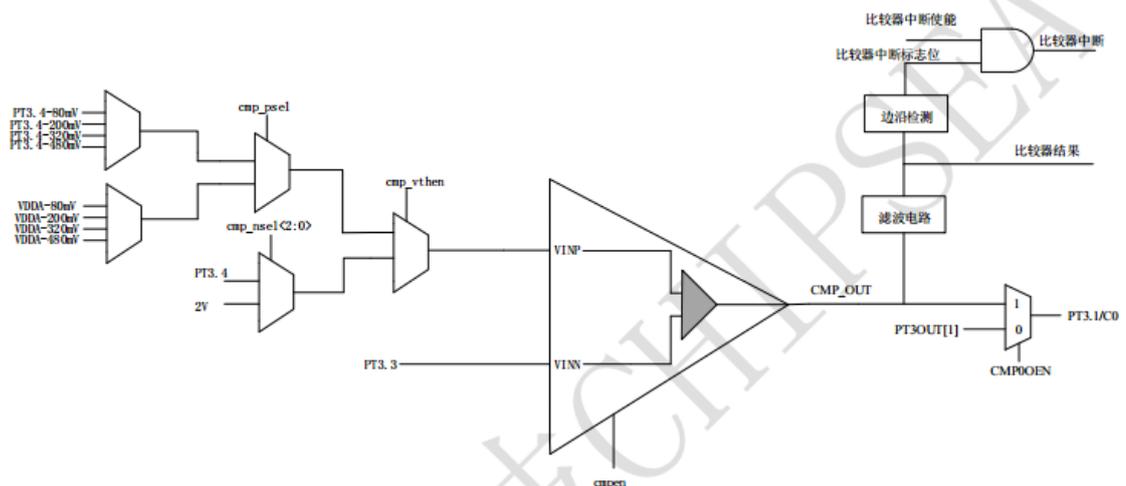
### 2.1 模拟比较器应用问题

CSU32M10 相对 CSU32P20 新增模拟比较器模块，两个模拟输入端 C0P (PT3.4) 和 C0N (PT3.3)，也可以使用比较器内置阈值 80mv、200mv、320mv、480mv 作为比较器的一个输入，C0 (PT3.1) 脚可做为比较器的输出，该输出为数字输出，不需要将该口配置为模拟口。

CSU32M10 可以通过比较器结果变化关闭 PWM 输出，可应用于短路保护功能。

模拟比较器在 SLEEP 模式下不能正常工作，因此，在进入 SLEEP 模式前，需要把模拟比较器使能关闭。

模拟比较器逻辑示意图：



注意：1.使用比较器时，需保证正端与负端电压输入 $>2.4V$ ，否则比较器可能工作异常。

2.模拟比较器的输出滤波计数不允许配置为0个指令周期，否则模拟比较器可能工作异常。

### 2.2 恒流源

CSU32M10 相对 CSU32P20 新增内置 50mA 恒流源模块，通过 PT3.2 口输出，将 CCSCON 寄存器的 CCSOEN 位置 1 即可。

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0						
CCSCON								CCSOEN

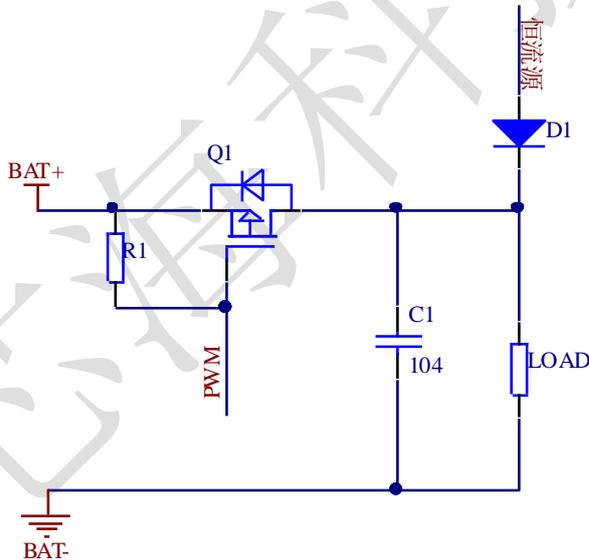
位地址	标识符	功能
7: 1	保留	保留
0	CCSOEN	电流源输出使能控制位 0: 禁止 50mA 电流源从 PT3.2 口输出 1: 使能 50mA 电流源从 PT3.2 口输出

Eg:

```

void main()
{
    PT3PU_2 = 0; //disable pull up
    PT3PD_2 = 0; // disable pull down
    CCSOEN = 1; //enable current source
    while(1);
}
    
```

注意：使用恒流源模块时，若负载非纯阻性负载可能引起恒流源模块震荡，所以负载两端建议并联 0.1uf 以上电容，如下示意图所示：



## 2.3 EEPROM

CSU32M10 相对 CSU32P20 增加 EEPROM 模块，EEPROM 地址范围为 0x2000~0x207F。通过 TBLP 指令可以实现对 EEPROM 进行写操作。每个 EEPROM 地址存储一个 byte 数据，因此在进行读写操作时，仅低 8 位数据有效。

对 EEPROM 进行写操作时，必须解锁写保护。对 EEPROM 写操作进行解锁时，需对 WRPRT 寄存器连续写入 96H、69H、5AH。对其他地址寄存器进行写操作时，WRPRT 寄存器会被清零，解锁自动失效。

### EEPROM 写操作流程

- (1) 配置 {EADRH, EADRL} 写操作地址。
- (2) 建议关闭全局中断使能 GIE。
- (3) 清除 wdt 计数值。
- (4) 向 WRPRT 寄存器写入解锁序列，解锁写保护。
- (5) 配置 WORK 写操作数据。
- (6) 执行指令 TBLP XH 操作。
- (7) 写操作完成后，检查 ISPCON1 寄存器的 CHKRSLT 位，当该位为 0 时，表示写操作校验失败，读出数据与写入数据不一致。当该位为 1 时，表示写操作成功。

### EEPROM 读操作

- (1) 配置 {EADRH, EADRL} 写操作地址。
- (2) 执行指令 MOVP 操作。
- (3) 读取完成。

注意：(1) 进行解锁操作前，需要关闭全局中断

(2) 写解锁成功后，对其他地址寄存器进行写操作，会使 WRPRT 寄存器清 0，导致解锁失败，所以在 C 程序中，解锁成功后不允许对变量赋值，变量赋值需放在解锁前，如下为解锁失败案例：

```
EADRH = address_buf[0]; // write address
EADRL = address_buf[1];
GIE = 0; // disable interrupt
WRPRT = 0x96; // unlock
WRPRT = 0x69;
WRPRT = 0x5A;
while(!WRPRTF);
write_data = 0xAA;
asm ("TBLP 0H"); //write data
asm ("NOP");
asm ("NOP");
if(CHKRSLT);
```

由于在写解锁后对 `write_data` 变量赋值，导致写解锁失败，无法对 EEPROM 写入数据，正确案例如下：

```
EADRH = address_buf[0]; // write address
EADRL = address_buf[1];
GIE = 0; // disable interrupt
write_data = 0xAA;
WRPRT = 0x96; // unlock
WRPRT = 0x69;
WRPRT = 0x5A;
while(!WRPRTF);
asm ("TBLP 0H"); //write data
asm ("NOP");
asm ("NOP");
if(CHKRSLT);
```

(3)建议在解锁操作前，考虑使能看门狗模块，防止芯片在写 EEPROM 过程中出现异常情况无法复位。由于写操作时间较长，建议在解锁前将 wdt 计数清 0。

## 2.4 芯片休眠电流过大问题

CPU 执行睡眠指令后，所有的振荡器停止工作直到出现一个中断事件唤醒 CPU。在睡眠模式下的功耗大约 1uA。

为了保证 CPU 在睡眠模式下的功耗最小，在执行睡眠指令之前，需要保证所有的输入口是接到 VDD 或 VSS 电平。ADC 模块在 SLEEP 模式下必须关闭，并且模拟输入通道不能配置为 1/8VDD（SRADCON2 寄存器的 CHS[3:0]位不能配置为 0101）。

休眠前需保证设定为数字输入的 IO 口不允许悬空，需使能内部上拉电阻或者下拉电阻，否则待机功耗较大。

Eg:

```
void csu32m10_enter_sleep(void)
{
    //PT1 io config
    PT1 = 0;
    PT1EN = 0xF7; // set PT1.3 input mode
    PT1PU = 0x08; // PT1.3 pull up
    PT1CON = 0x14; //enable PT1.3 INT1
    PT1PD = 0;

    //PT3 io config
    PT3 = 0;
    PT3EN = 0x3F;
    PT3PU = 0;
    PT3CON = 0;
    PT3PD = 0;

    //PT5 io config
    PT5 = 0;
```

```
PT5EN = 0x03;
PT5PU = 0;
PT5CON = 0;
PT5PD = 0;

CST_WDT = 1; // disable wdt clk
TM0CON = 0; // disable timer0
TM2CON = 0; // disable timer2
TM3CON = 0; // disable timer3
SRADCON1 = 0; // disable ADC

wakeup_check:
while(!PT1_3);
E1IF = 0; //clr int1 flag
E1IE = 1; // enable int1
GIE = 1;
asm("sleep"); // enter sleep mode

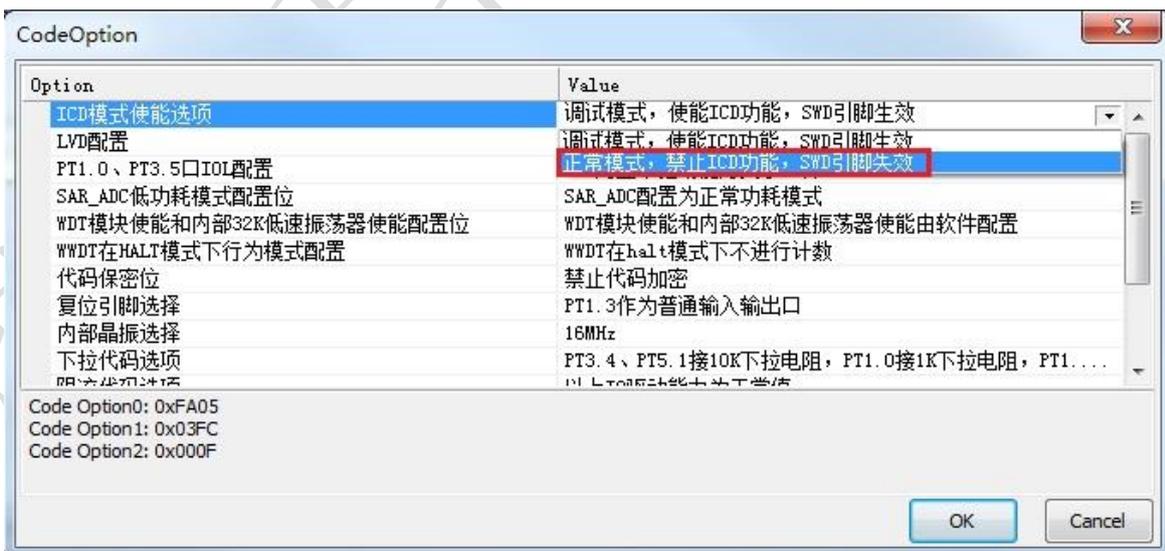
delay_us();
E1IE = 0;
goto wakeup_check;
}
```

注意：若需进入休眠模式，在进行程序烧录时，需在烧录代码选项里禁止 ICD 功能，否则可能功耗异常，如图所示：

CS-Write 代码选项配置：

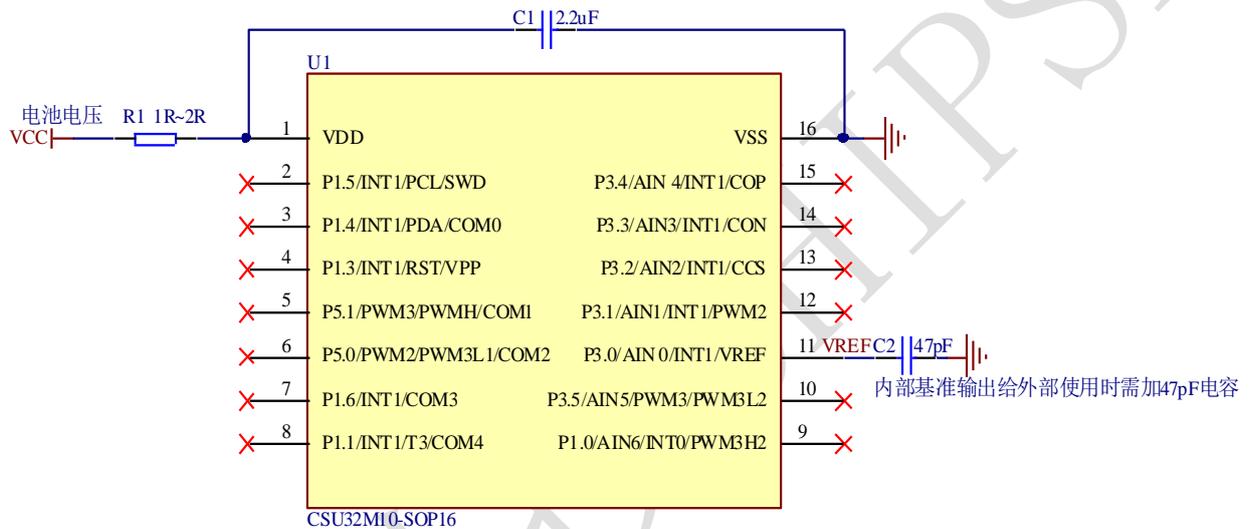


IDE 代码选项配置:



## 2.5 使用模拟比较器进行短路保护注意事项

CSU32M10 内部集成模拟比较器，可使用比较器结果输出触发 PWM 信号关断功能做电子烟负载短路保护，通常保护时间 5~20us（与比较器配置滤波时间有关），使用该模块时需注意以下问题：芯片 VDD 与电源电压输入端需串联 1~2Ω 电阻，滤波电容建议使用 2.2uF 以上，若输入端无串联电阻，会引起芯片 VDD 电压震荡，导致芯片工作异常，短路保护响应时间变长。参考电路如下图所示：



### 3 IDE 与 CS-LINK 使用注意事项

1、代码选项配置 ICD 模式使能或禁止，均可进行 ICD 调试，区别在于：

使能：非调试状态时 PT1.5 仍为调试口，IO、中断等功能无法正常使用，且 sleep 时高速时钟无法关闭。

禁止：非调试状态时 PT1.5 功能正常，sleep 时高速时钟正常关闭，若需要芯片进入 sleep 模式，代码选项应禁止 ICD 调试功能。

2、如果调试板上 PT1.3 口未做复位口或没有上拉，禁止代码选项配置 PT1.3 口做复位口。

3、代码选项配置，【代码保密位】配置为【使能代码加密】，将无法进行 ICD 调试，无法进行烧录。

4、调试状态下，halt、sleep 状态下调试停止时，无法进行修改 PC 操作，即跳到光标等功能失效。

5、WWDT 寄存器在调试停止状态时无法在窗口修改 WWDT 相关寄存器。

6、在较低指令周期下（小于 500KHz），无法进行修改 PC 操作，即跳到光标等功能失效。

7、调试板上建议对调试口 PT1.5 做电路隔离，在做调试口时不接其他电路。

8、CS\_Link 已在调试口加上拉电阻，调试板上禁止在调试口上接上下拉电阻。

9、不允许在 tblp、movp 的下一指令处设置断点。

10、 执行 clrf f 指令时，对地址 f 执行读写操作，因此 f 地址满足断点条件时 clrf 会触发读或写操作断点。

11、 数据断点地址不能配置为 work 寄存器。

12、 call stack 功能只对函数调用有效，双击 call stack 窗口信息即可跳转到各级函数调用入口。若双击 call stack 信息光标跳转到非函数调用入口时，则该条 call stack 信息为中断引起，该条 call stack 信息无效。

- 13、 ICD 调试与 ICE 调试区别：ICD 无法在 IDE 窗口修改系统寄存器值（SFR 地址 00-08h，但不包括 INTE/INTF 寄存器，INTE 中 GIE 不能修改），而 ICE 可以修改。
- 14、 在数据断点地址为 sram 时，在调试停止时，下一条指令满足 sram 地址区的数据断点时（即光标停在代码行），debug run 不会触发断点。如下代码所示，断点设置的写 sram 80 地址时触发断点，此时调试停止光标停止在第二行（未执行），在 debug run 不会触发断点。数据断点地址为 SFR 时无该现象。
- ```
movlw    0x55  
movwf   sram_80
```
- 15、 目前 CS-LINK（固件版本号：V0.6）及之前版本不支持外部供电，烧录或仿真时需要关闭外部电源。
- 16、 halt 模式下由于内核时钟停止，所以窗看门狗复位时 wwdtif 标志位无法置一。