



CMS32S28F应用注意事项

应用注意事项

Rev. 1.0.0

请注意以下有关CMS知识产权政策

* 中微半导体（深圳）股份有限公司（以下简称本公司）已申请了专利，享有绝对的合法权益。与本公司MCU或其他产品有关的专利权并未被同意授权使用，任何经由不当手段侵害本公司专利权的公司、组织或个人，本公司将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨本公司因侵权行为所受的损失、或侵权者所得的不法利益。

* 中微半导体（深圳）股份有限公司的名称和标识都是本公司的注册商标。

* 本公司保留对规格书中产品在可靠性、功能和设计方面的改进作进一步说明的权利。然而本公司对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，本公司不保证和不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。本公司的产品不授权适用于救生、维生器件或系统中作为关键器件。本公司拥有不事先通知而修改产品的权利，对于最新的信息，请参考官方网站 www.mcu.com.cn

目录

1. 仿真/烧录应用注意事项	3
2. I/O 口	3
3. 触摸	3
4. FLSAH 使用	3
1.1 FLASH 锁定	3
5. 上电	4
6. 1T 模式下 TA 寄存器操作注意事项	4
1.1 注意事项	4
1.2 推荐操作	4
7. 其他注意事项	5
8. 版本修订说明	6

1. 仿真/烧录应用注意事项

- 芯片支持双线（DSDA、DSCK）/单线（SWE）仿真烧录，可通过用户配置选择。仿真时需配置为调试模式，调试完成后需关闭调试模式。

7. DEBUGEN（调试模式使能）

- ◆ DISABLE（默认）

调试模式禁止，DSCK，DSDA 引脚用做普通 IO 口

- ◆ ENABLE

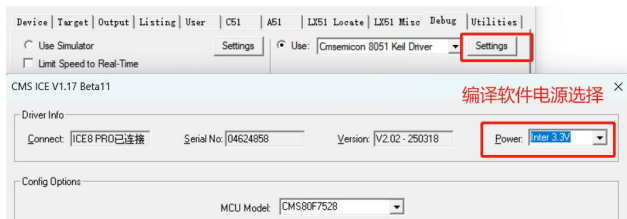
调试模式使能，DSCK，DSDA 引脚配置成调试口，引脚对应的其他功能关闭

8.DBGSEL（调试口选择，需要调试模式使能）

- ◆ TWI（默认双线调试）

- ◆ SWI（单线调试）

- 调试状态下，DSCK/DSDA 作为专用调试口，严禁复用成触摸功能，不能实现其 GPIO 及复用功能。
- 使用双线仿真/烧录时，选择 5V 电压如果出现通讯失败情况，可采用以下其中一种方式处理。
 - 仿真/烧录电压选择 3.3V，如下图。
 - 使用带屏蔽的通讯线，不使用排线，避免通讯时相互干扰。
 - 烧录口（DSDA、DSCK）需串接 100R 电阻。（新方案推荐使用）
 - 选择单线 SWE 仿真/烧录。



2. IO 口

- 建议外接的排线使用时靠近 IO 串接电阻，例如 NTC 应用，管脚先经过电阻再并接电容。

3. 触摸

- 触摸按键与 MCU 之间需串接电阻（建议 1K），且靠近 MCU 放置，具体触摸应用电路可参考《CMS 系列芯片硬件使用注意事项》。
- 使用项目实际电源、整机功能进行触摸可靠性评估，具体操作参看《触摸调试手册》。
- 根据应用需求选择 CS 库和普通库。

CS 库：提升触摸 CS 测试性能，需要严格评估触摸信噪比，具体操作参看 2.0 触摸调试软件使用说明。

普通库：具有高降噪、响应快、适用范围广等特点；项目无 CS 要求，请选用普通库。

4. FL5AH 使用

1.1 FLASH 锁定

对 FLASH 进行操作前需要对 FLASH 进行解锁，操作完 FLASH 需要给 FLASH 上锁，可直接操作 MLOCK 寄存器。

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MLOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
读写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0

Bit7~Bit0 MLOCK<7:0>: 存储器操作使能位;
AAH= 允许存储器相关W/E/R/CRC操作;
其他= 不允许存储器相关W/E/R/CRC操作。

对不需要程序操作的 FLASH 部分，用户配置可选择保护，增加 FLASH 抗干扰。

15. WRITE_PROTECT 程序分区保护（可保护区间，所有默认区间为不保护）

- ◆ 0000H-07FFH（保护/不保护）
- ◆ 0800H-0FFFH（保护/不保护）
- ◆ 1000H-17FFH（保护/不保护）
- ◆ 1800H-1FFFH（保护/不保护）
- ◆ 2000H-27FFH（保护/不保护）
- ◆ 2800H-2FFFH（保护/不保护）
- ◆ 3000H-37FFH（保护/不保护）
- ◆ 3800H-3FFFH（保护/不保护）
- ◆ 4000H-47FFH（保护/不保护）
- ◆ 4800H-4FFFH（保护/不保护）
- ◆ 5000H-57FFH（保护/不保护）
- ◆ 5800H-5FFFH（保护/不保护）
- ◆ 6000H-67FFH（保护/不保护）
- ◆ 6800H-6FFFH（保护/不保护）
- ◆ 7000H-77FFH（保护/不保护）
- ◆ 7800H-7FFFH（保护/不保护）

5. 上电

确保上电斜率满足手册要求且留有余量，保证可靠的上电复位。芯片上电过程中要确保芯片的 VDD、GND 脚接触良好，保证芯片处于正常上电工作模式。

6. 1T 模式下 TA 寄存器操作注意事项

1.1 注意事项

对 TA 寄存器操作期间，要求 TA=0xAA 和 TA=0x55 之间不能中断。所以在对 TA 进行操作时，我们需要将中断使能关闭(EA=0)。常规范例为 EA = 0; → TA = 0xAA; → TA = 0x55

当指令时钟选择≥2T 时，此范例是可以正常无任何风险；

当指令时钟选择 1T 时，由于内部时序控制加快，需要对指令进行预取指，当内核在执行 EA=0 过程中，有中断发生，此时内核会先响应中断并将 EA=0 的下一条指令执行完成后开始执行中断程序。这个过程中由于 TA=AA\55 不连续导致解锁寄存器失效。

1.2 推荐操作

常规范例	推荐范例
EA = 0;	EA = 0;
TA = 0xAA;	nop ();
TA = 0x55;	TA = 0xAA;
	TA = 0x55;

对 WDCON、IREMP 和 WDKEY 寄存器操作时需要写 TA 时序，时序寄存器 TA 操作时要求整个过程不能被中断，在写 TA 时序前先将中断关闭，等配置完成后根据需要再打开。

注意避免编译器选择高优化等级(9 级)把 TA 操作被优化成了一个函数，导致寄存器操作失效，建议该部分程序使用宏定义或优化等级选择 8 级或者以下。

```
*****  
//在CPU_WAITCLOCK选择1T的模式时，在EA=0后必须加nop，选择多T时不加。  
#define WDT_ClearWDT() do{\n    uint8_t temp = EA;\n    EA = 0;\n    _nop_();\n    TA = 0xAA;\n    TA = 0x55;\n    WDKEY = 0X99;\n    EA = temp;\n}while(0)  
*****
```

7. 其他注意事项

- 对于需要连续设置几个语句来完成一个操作的程序，请不要在连续语句中间加延时或断点；如果中断在中途发生，会中断连续设置的逻辑，从而影响设置的成功率。
- 对程序寄存器的位操作，建议先将寄存器的值读出来，放在一个临时变量里，然后再根据对应的位，来修改临时变量的值；最后将临时变量的值写入到寄存器里。

8. 版本修订说明

版本号	时间	修改内容
V1.0.0	2025 年 3 月	初始版本