

应用笔记

Application Note

AN0003

MM32F103 常见问题解答

版本:1.0

目录

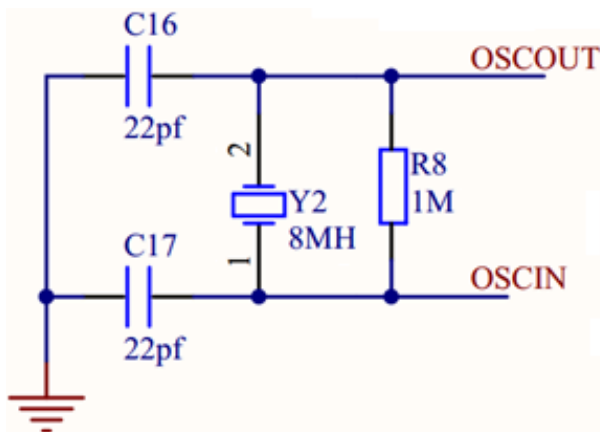
1. 系统时钟	4
1.1 外部高速时钟接法	4
1.2 外部时钟异常常见原因	4
1.3 内部高速时钟做系统时钟频率是多少.....	4
1.4 PLL时钟源有哪些各是多少.....	4
1.5 系统为什么经常出现hard_fault.....	5
2. ADC	5
2.1 ADC输入频率最大值和分频系数	5
2.2 ADC读取的数据浮动较大	5
2.3 ADC工作模式和注意事项	5
2.4 如何判断ADC数据有效性	6
2.5 ADC外部触发有什么作用	6
3. USB	6
3.1 USB电路连接为什么不需要上拉电阻	6
3.2 USB怎么支持热插拔	6
3.3 USB只支持特定的系统或不支持HUB.....	6
3.4 USB枚举不通过或枚举后数据收发异常.....	6
4. CAN	6
4.1 CAN收发不正常有哪些原因	6
4.2 CAN数据溢出数据处理.....	7
4.3 CAN过滤器有哪些.....	7
5. SPI	7
5.1 SPI支持哪几种模式.....	7
5.2 SPI主机通信不正常有哪些原因	7
6. I2C	8
6.1 I2C配置注意什么.....	8
6.2 I2C 主机通讯不正常有哪些原因	8
6.3 I2C 时钟什么时候会被拉低.....	8

6.4	怎么判断是主机还是从机拉低时钟	8
7.	RTC.....	8
7.1	RTC读计数器数据时与预期有一个值或几个值的偏差	8
7.2	RTC写入RTC_PRL&RTC_CNT&RTC_ALR寄存器的步骤	8
8.	Vbat 的电源接入有什么要求;	8
9.	KEIL 例程编译失败可能导致的原因	9
10.	KEIL 程序编译通过了,可是下载程序失败	9
11.	确定JTAG/SWD 硬件连接都是好的,可就是读不到芯片ID号?.....	9

1. 系统时钟

1.1 外部高速时钟接法

外部无源晶振电路如下图所示，晶振两脚接约 22pf 电容，并上 1M 反馈电阻。因芯片内部没有集成反馈电阻，为保证 XTAL 起振，必须接 1M 欧姆电阻；



1.2 外部时钟异常常见原因

运行程序通常用到外部高速时钟做系统时钟源，有时候在调试中会遇到系统时钟异常导致程序停止运行，以下列出几点可能的原因：

- 1) 外部晶振未加反馈电阻，导致外部无稳定时钟输入；
- 2) 外部晶振范围 8~24MHz；
- 3) 晶振与芯片引脚间断路；
- 4) 晶振质量问题导致，不正常起振；
- 5) 芯片系统时钟配置过程错误等等。

1.3 内部高速时钟做系统时钟频率是多少

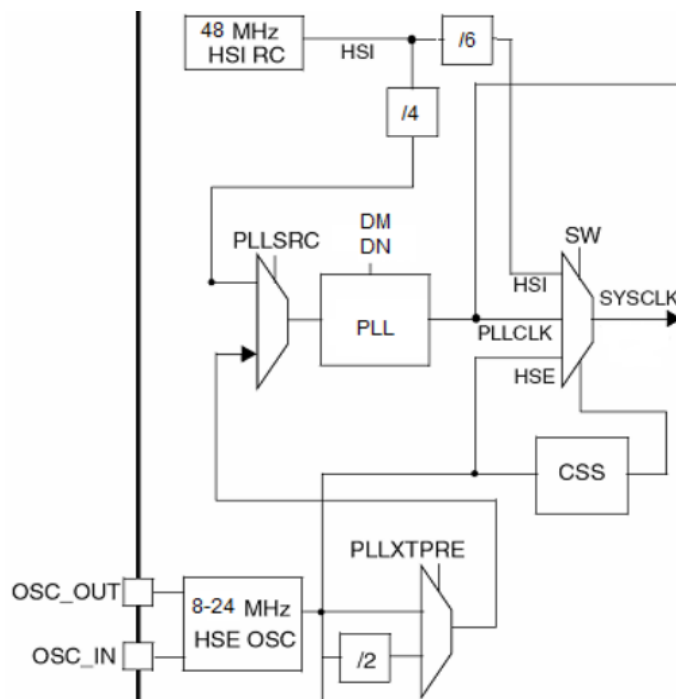
内部高速振荡电路 HSI_RC 为 48MHz，精度正负 1%，做系统时钟时为 6 分频即 8MHz。

内部晶振使用有以下 2 种情况：

- 1) 直接开启内部晶振作为系统时钟，不使用 PLL，即 48MHz 经过 6 分频后为 8MHz
- 2) 使用经过 PLL 倍频后的内部时钟作为系统时钟，是以 48MHz 经过 4 分频之后的 12MHz 为基准往上倍频的。(例如想要得到内部 72MHz 的系统时钟，应该 PLL 设置为 6 倍频，即 $12\text{MHz} \times 6 = 72\text{MHz}$ 。)

1.4 PLL 时钟源有哪些各是多少

- 1) 内部 48MHz HSI_RC 时钟 4 分频做 PLL 时钟源；
- 2) 外部时钟 2 分频做 PLL 时钟源；
- 3) 外部时钟 1 分频做 PLL 时钟源。



1.5 系统为什么经常出现 hard_fault

- 1) 系统时钟配置与 FLASH 延时周期不匹配(0~24M,24~48M,48~72M,72~96M 分别对应 0,1,2,3);
- 2) 野指针被赋值;
- 3) 指针作为函数形参被传入了非法值或不使用该参数时随便填入的值遭到调用;
- 4) 程序访问了非法空间,或进行了非法操作等.

2. ADC

2.1 ADC 输入频率最大值和分频系数

ADC 最大输入时钟为 15MHz;ADC 分频系数由 ADCFG 寄存器中 bit4~6 实现,最大可配置为 7, 即 PCLK2 时钟 $2*(7+1)=16$ 分频.

2.2 ADC 读取的数据浮动较大

- 1) ADC 的参考电源尽量干净,尽量远离或隔离嘈杂的环境;
- 2) 在满足实际应用的情况下尽量加大 ADC 分频,即减小 ADC 输入时钟, ADC 时钟最大应不超过 15MHz;
- 3) 缓慢变化的信号可以并入合适电容滤去高频, 信号强度较弱时应使用运放或其
- 4) 它器件放大信号强度.

2.3 ADC 工作模式和注意事项

- 1) 单次转换模式:

每次转换一个 AD 通道, 转换结束后 ADCR bit8 即转换使能位自动清零, 转换完成后直接 读取数据, 转换使能位不要手动清零。

- 2) 单周期转换模式:

每次开始转换时,会将使能的通道全部转换一次, 转换结束后 ADCR bit8 即转换使能位自动清零, 转换时可查询 ADDATA 寄存器读取数据, 也可以读对应的 ADDR_x 来获取数据,注意判断数据有效性, 转换使

能位不要手动清零.

3) 连续扫描模式:

开始转换后会一直周期性转换使能的通道, 转换结束后 ADC bit8 不清除, 不要手动清零;读取数据通单周期转换模式.

2.4 如何判断 ADC 数据有效性

ADC 外设中有 ADDATA 寄存器与分立的 ADDR_x 寄存器,获取数据有两种方式 (ADC 查询方式)

- 1) 查询 ADDATA 寄存器数据并判断对应通道编号和数据有效标志;
- 2) 查询对应 ADDR_x 寄存器数据并判断数据有效标志.

2.5 ADC 外部触发有什么作用

产生特定的事件时 ADC 开始启动转换, 对用户来说 ADC 采样的时机更加灵活.

3. USB

3.1 USB 电路连接为什么不需要上拉电阻

上拉电阻集成在内部, USB_TOP 寄存器 bit1 置 1 时 D+加上拉电阻, 反之上拉电阻断开.

3.2 USB 怎么支持热插拔

将枚举过程及数据过程循环调用即可实现, 较常见的方法是使用 USB 中断.

3.3 USB 只支持特定的系统或不支持 HUB

常见原因有:

- 1) 在响应 USB 主机复位信号时, 将 USB 设备 1.5K 电阻断开又重连;
- 2) 目标 HUB 不支持设备连接;

3.4 USB 枚举不通过或枚举后数据收发异常

常见原因有:

- 1) USB 配置不正确, 端点中断或 USB 相关中断未使能, 表现为进不了相关中断;
- 2) USB 函数响应不正确, 表现为数据收发错误;
- 3) USB 函数响应不完全, 表现为特定数据未发送(host 显示枚举完成时类描述符可能未传完);
- 4) USB 响应时序乱, 表现为不符合 USB 四大传输过程;
- 5) 未清 USB 中断, 表现为程序不出中断;
- 6) 数据过程出错可能为逻辑判断出错, 大部分原因同枚举阶段.

4. CAN

4.1 CAN 收发不正常有哪些原因

- 1) 波特率不匹配, CAN 通讯要求双方具有同样的波特率设置, 否则表现为收发数据失败;
- 2) 配置不正确, 表现为相关标志位不能正常置位;
- 3) CAN 过滤器设置不正确, 表现为发送正常接收异常;
- 4) 硬件连接异常, 表现为 CAN_H 或 CAN_L 上无信号;

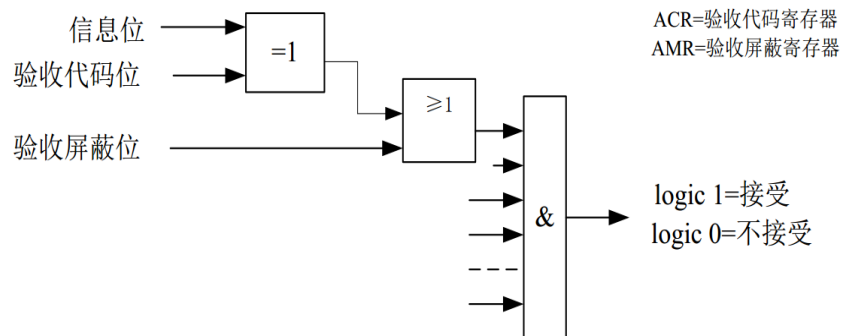
- 5) 数据溢出,表现为发送数据正常,接收数据异常;
- 6) 工作模式设置不正确等.

4.2 CAN 数据溢出数据处理

可复位 CAN 相关寄存器,长期处于数据溢出状态会导致 CAN 模块效率变低.

4.3 CAN 过滤器有哪些

单个 32 位或两个 16 位过滤器,每个过滤器包含若干验收屏蔽寄存器和若干验收代码寄存器,由这两种寄存器共同决定是否接收当前数据帧,下图为验收逻辑.



5. SPI

5.1 SPI 支持哪几种模式

按传输方向分

- 1) 全双工模式,同时收发数据,同时使能 TX 和 RX;
- 2) 半双工,在不同时间段进行读写,ENABLE TX 时 DISABLE RX,ENABLE RX 时 DISABLE TX;

按采样时分

- 1) 模式 0,空闲时时钟为低,第一个时钟沿采样;
- 2) 模式 1,空闲时时钟为低,第二个时钟沿采样;
- 3) 模式 2,空闲时时钟为高,第一个时钟沿采样;
- 4) 模式 3,空闲时时钟为高,第二个时钟沿采样;

5.2 SPI 主机通信不正常有哪些原因

常见原因:

- 1) 配置不正确,表现为无时钟输出;
- 2) 模式配置不正确,表现为采样点与预期不一致(SPI_CCTL bit0 = 0 为第二个时钟沿采样);
- 3) 速度配置过快,表现为波形异常;
- 4) 数据位数配置与 device 不一致,表现为 CLK 个数不对.
- 5) CS 信号不正确,表现为 CS 信号与 device 时序不对;

6. I2C

6.1 I2C 配置注意什么

- 1) 配置 I2C_TAR 寄存器时应保持 I2C 未使能,且使能时 CLK&I2C 为高电平;

6.2 I2C 主机通讯不正常有哪些原因

常见原因

- 1) 配置不正确,表现为 I2C 无时钟输入;
- 2) 从机地址不正确,表现为 I2C 有时钟且数据发送正常,但是无应答;
- 3) 通讯时序不正确,表现为 I2C 接收到应答信号,且有第一或前几帧数据送出;
- 4) I2C 时钟被拉低,总线异常;
- 5) 未接上拉电阻.

6.3 I2C 时钟什么时候会被拉低

对于主机或从机:

- 1) 发送数据时发送缓冲寄存器为空;
- 2) 接收数据时,数据未读出(不一定每个器件都有此操作).

6.4 怎么判断是主机还是从机拉低时钟

断开主机与从机的时钟,测量主机和从机的任意一个 CLK,即可得出结论.

7. RTC

7.1 RTC 读计数器数据时与预期有一个值或几个值的偏差

- 1) RTC 读取数据时需要先进行同步操作;
- 2) 配置 RTC 相关中断前应先清除对应中断标志.

7.2 RTC 写入 RTC_PRL&RTC_CNT&RTC_ALR 寄存器的步骤

CNF&RTOFF 均在 RTC_CRL 寄存器中

- 1) 查询 RTOFF 位,直到 RTOFF 的值变为 1;
- 2) 设置 CNF 值为 1,进入配置模式;
- 3) 写以上几个寄存器;
- 4) 清除 CNF,退出配置模式;
- 5) 查询 RTOFF 位,直到变为 1 表示写操作完成.

8. Vbat 的电源接入有什么要求;

如果在应用中没有外部电池, 建议 VBAT 在外部通过一个 100nF 的陶瓷电容与 VDD 相连. 如外接为电池, 为保证不损坏, 建议在外部 VBAT 和电源之间连接一个低压降二极管。如无外接电池, 即使不用 RTC 功能, 也需要给 VBAT 供电;

9. KEIL 例程编译失败可能导致的原因

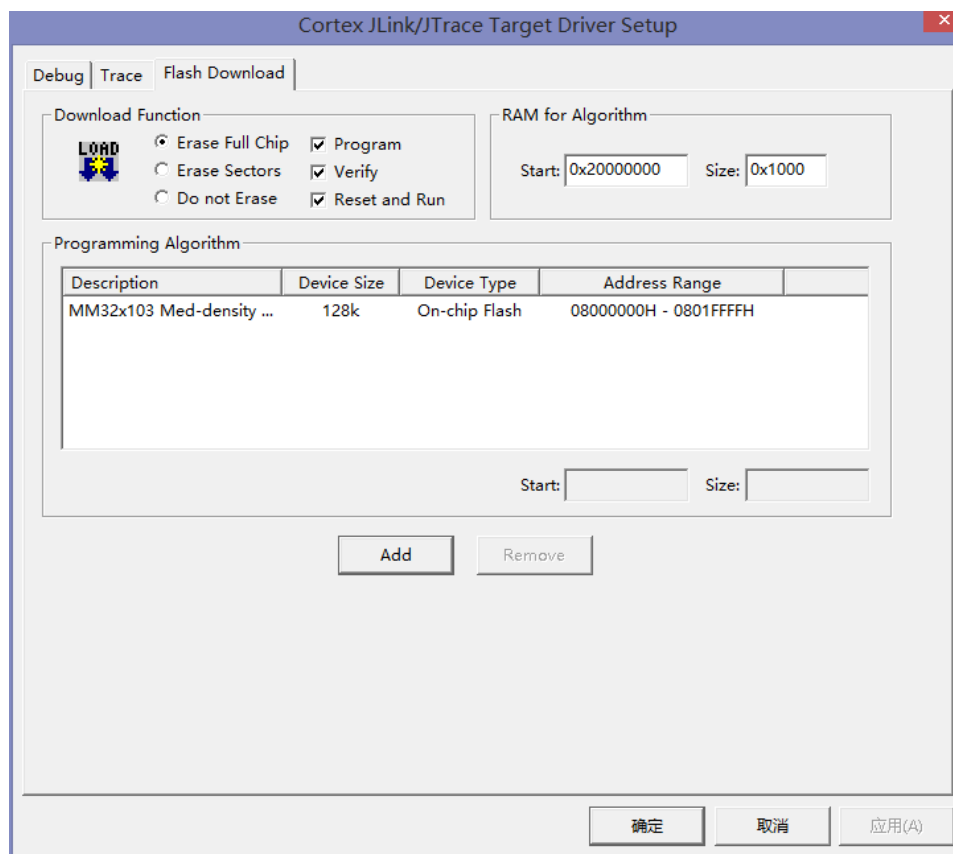
常见原因

- 1) 没有安装 keil4 的兼容包，软件名称为：MDKCM516_legacySupportMDK4（出问题最多）
- 2) 移动了 keil 工程文件，导致无法找到相对路径的文件
- 3) 程序太大，超过 FLASH 或者 SRAM 容量

10. KEIL 程序编译通过了，可是下载程序失败

常见原因

- 1) 硬件电路没有接好，查看 debug 工具是否连接上板子中的 JTAG 或者 SWD，能否读到芯片 IDCODE。
- 2) 若是 Debug 工具能读到芯片 ID,但是无法下载，原因是没有选择 Description 型号，具体配置选择如下图所示。



11. 确定 JTAG/SWD 硬件连接都是好的，可就是读不到芯片 ID 号？

常见原因

- 1) 非法操作 RCC 时钟，导致 PLL 锁死
- 2) MCU 进入睡眠、停止、待机模式
- 3) VBAT 没有供电

解决办法

针对 1 和 2 的问题，将 BOOT0 和 BOOT1 接到高电平，复位或者重新上电一下，然后再读 ID。若能读到 ID，则在此模式下擦除程序，然后再将 BOOT0 和 BOOT1 接低电平，这时候 ID 就能读到了。