FCM32F1xx/FCM32H1xx 应用相关注意事项

(C 版 IC)

FCM32F1xx/FCM32H1xx 为基于 Cortex M4 内核的控制器，与 STM32F1xx 系列兼容，由于生产工艺/电路设计等原因，两者之间仍然会存在一些差异，以下逐一说明。

目录

1 相同点

- 脚位
- 内存映射
- 开发环境（Keil/IAR）
- 烧写工具
- 库函数/例程
- 同频性能
- 时序兼容

2 不同点

项目	STM32F1	FCM32F1	FCM32H1
Cortex 内核	M3	M4	M4
VDD	2.0~3.6V	1.8~5.5V	1.8~5.5V
VDDA	2.0~3.6V	1.8~5.5V	1.8~5.5V
VBAT	√	×	×
工作温度范围		-40~85	-40~85
SRAM	20KB	20KB	32KB
Coremark 性能	181.0@72MHz	191.4@72MHz	239.5@72MHz 319.4@96MHz
CPU/AHB/APB2 最高频率	72MHz	72MHz	96MHz
APB1 最高频率	36MHz	72MHz	96MHz
Flash 工作频率	24MHz	24MHz	32MHz
Flash Turbo	-	-	√
TIM1/2/3/4 最高时钟	72MHz	72MHz	192MHz
SPI1 作为 I2S1	×	√	√
SPI 支持 NSSP 模式	×	√	√
SPI 支持 TI 模式	×	√	√
Boot loader	USART	USART	USART
工作电流(HSI 8MHz)	9.85mA	4.92mA	
工作电流(HSI+PLL72MHz)	35.4mA	26.7mA	
工作电流(HSI+PLL96MHz)	45.9mA	35.2mA	
Stop 电流(I_{DD}+I_{DDA},LDO=Run)	20.7uA	18.2uA	18.2uA
Stop 电流(I_{DD}+I_{DDA},LDO=LPR)	10.4uA	7.5uA	7.5uA
Standby 电流(I_{DD}+I_{DDA})	1.9uA	7.2uA	7.2uA

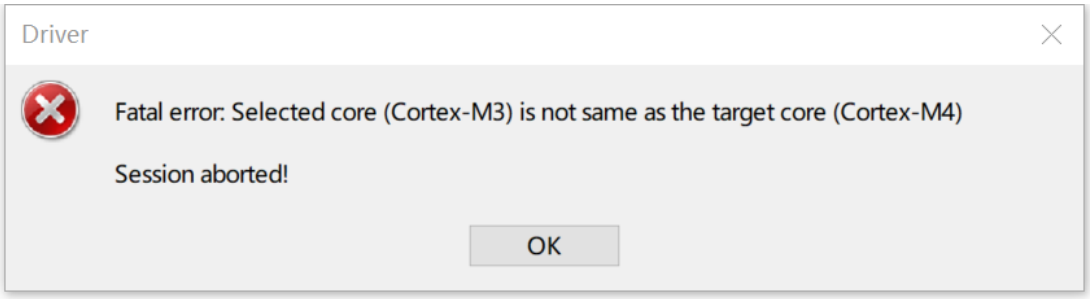
注：

1. 以 STM32F103CBT6 为参考。
2. 测试条件为 3.3V/25C，LSI/IWDG OFF，VDDA Monitor ON。
3. 96MHz 为 STM32F103 超频工作。

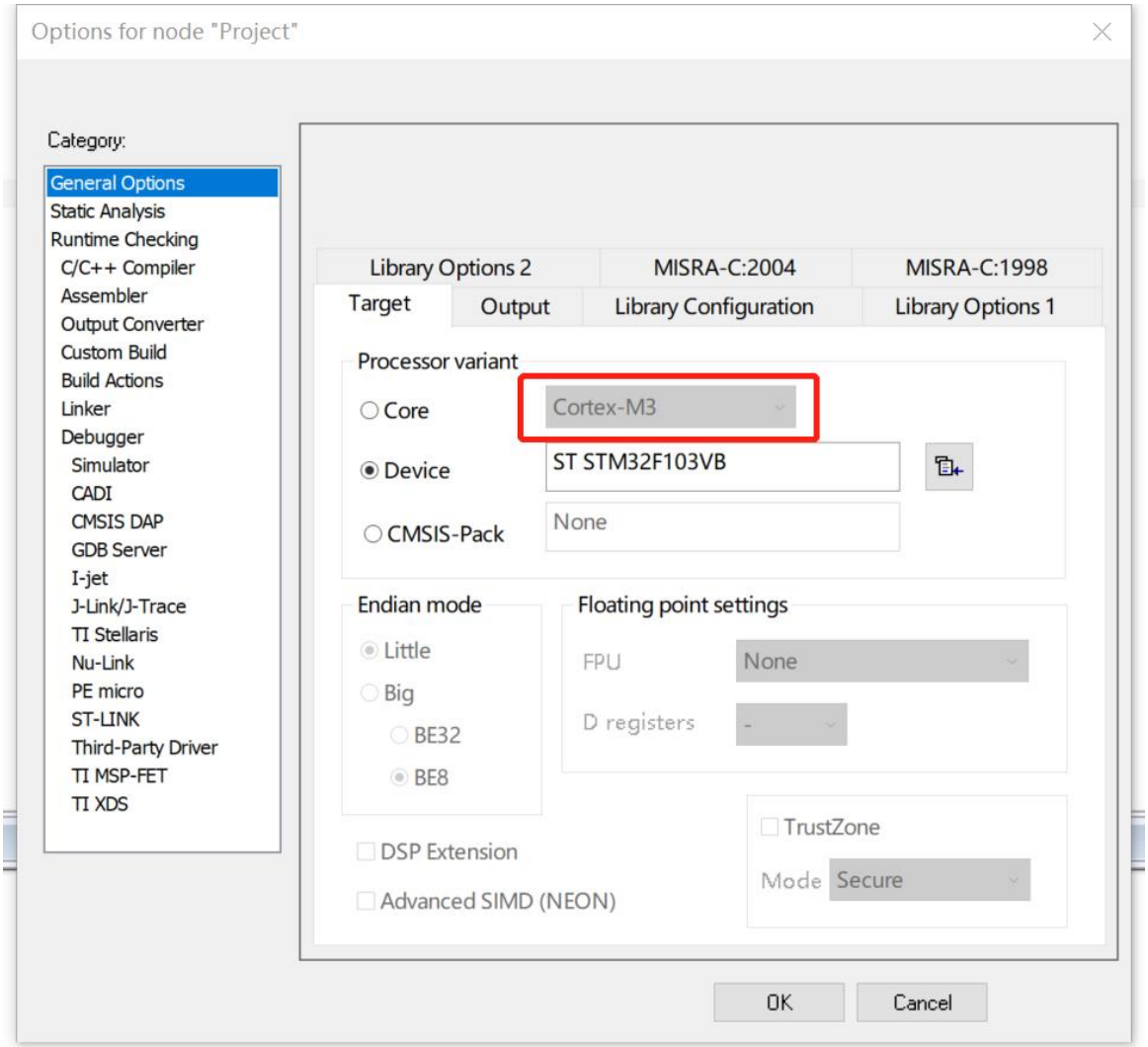
3 注意事项

3.1 Cortex 内核

FCM32x103 使用的是 Cortex M4（不带 F）内核，完全兼容 M3 的程序。但在 IAR 中调试时，由于其会判断内核版本，因此无法进行调试，出错信息如下：



打开工程选项，会发现 Core 使用的是 M3 内核：



解决方案：
使用 FCM32 提供的器件支持包。
下载地址：FCM32F103-深圳市闪芯微电子有限公司 (flashchip.com.cn)

3.2 VBAT 供电

FCM32x103 系列 IC 不提供掉电模式的后备电池供电，VBAT 为空脚。

3.3 ADC

3.3.1 ADC 精度

部分应用中 ADC 精度不够，因为 FCM 的 ADC 电气特性与 S**不尽相同，相同的采样时间，FCM 的输入阻抗（RAIN）要低，规格书中已列出 RAIN 表格。

解决方案二选一，或二者结合：

1) 根据外部输入阻抗，调整采样时间
2) 降低外部输入阻抗（通常为等比例降低分压电阻的阻值，或者使用运放做一级 buf）

3.3.2 单次转换

在使用单次转换模式，如果通道序列长度设置为大于 1 次（ADC_SQR1->L 或 ADC_JSQR->JL 不为 0），则每次触发会按通道设定依次转换，而 S**只会转换第一个通道。

解决方案：

需要单次转换时，ADC_SQR1->L 或 ADC_JSAR->JL 设为 0。

3.4 CAN

3.4.1 FCM 的 CAN 与 S**不能 BIN 兼容

可以使用 FCM 提供的 CAN 函数库重新编译。

3.4.2 CAN 时钟精度要求

在使用 CAN 功能时，MCU 时钟源必须使用晶振，不能使用内部 RC OSC。

3.5 USART

3.5.1 SR 的 TC 位不能通过写 0 清除

解决方案三选一：

1) 使用 TXE 代替 TC 位来控制发送
2) 在写入 DR 后再打开 TCIE 中断；在 TC 中断中，发送完最后一个数据之后关闭 TCIE 中断

```
//   发送第一个数据
__disable_irq();
temp = USART1->SR;
USART1->DR = buf[0];
__enable_irq();

//   中断函数
void USART1_IRQHandler(void)
{
    if(USART_GetFlagStatus(USART1,USART_FLAG_TC)!=RESET)
    {
        if (tx_count-- != 0)
        {
            USART1->DR = buf[p++];
        }
        else
            USART1->CR1 &= ~USART_CR1_TCIE;  //发送完最后一个数据，关掉 TCIE 中断
    }
    …
}
```

3) 采用 DMA 方式发送

3.5.2 USART3 REMAP 功能不正常

USART3_REMAP=3（映射到 PD8/PD9），工作不正常

解决方案：

无。

3.5.3 使用 HSI 时，波特率受温度影响过大

HSI 受温度影响可能超过 3%，因此可能导致串口通讯异常。

解决方案：

采用波特率自适应算法，使用定时器捕捉 RXD 的低电平宽度，估算出低电平的位数（接收到的数据中连续的 0 的位数未知），再算出实际波特率，重新设定波特率值。

3.6 RCC

3.6.1 RCC->CSR 寄存器不能使用 bit band 方式修改

在使用 FWlib 固件库时，会通过 bit band 的方式来开启 LSI，此时程序会停止。

解决方案：

不使用 bit band 方式，直接操作整个寄存器。
原来程序：

```
void RCC_LSICmd(FunctionalState NewState)
{
  /* Check the parameters */
  assert_param(IS_FUNCTIONAL_STATE(NewState));
  *(__IO uint32_t *) CSR_LSION_BB = (uint32_t)NewState;
}
```

修改后：

```
void RCC_LSICmd(FunctionalState NewState)
{
  /* Check the parameters */
  assert_param(IS_FUNCTIONAL_STATE(NewState));
//  *(__IO uint32_t *) CSR_LSION_BB = (uint32_t)NewState;
  if (NewState == ENABLE)
    RCC->CSR |= RCC_CSR_LSION;
  else
    RCC->CSR &= ~RCC_CSR_LSION;
}
```

3.6.2 选择 HSI 作为 PLL 时钟源时，需要系统频率为>=72MHz （*FCM 新增功能*）

解决方案：

FCM32x103 的 RCC_CFGR2 扩展了一位，用于选择 PLL 的输入时钟，使其在使用内部 RC 振荡器时可以支持 72MHz 及以上的频率。
RCC_CFGR2
Address: 0x2C
[31]：　PLLSRC[0]，默认=0

RCC_CFGR

Address: 0x04
[16]: PLLSRC[1]

PLLSRC[1:0]: PLL 时钟源选择
00 = 选择 HSI/2 作为 PLL 输入时钟
01 = 选择 HSI 作为 PLL 输入时钟
10 = 选择 HSE 作为 PLL 输入时钟
示范代码：
1) 将 PLL 配置成使用 HSI/2，倍频数为 9 倍（假设需要 8*9=72MHz）
2) 在 PLL enable 之前，插入一行：
(volatile uint32_t)(RCC_BASE+0x2c) |= 1<<31;

3.6.3 TIM 输入时钟需要 2 倍 PLLCLK （*FCM 新增功能*）

解决方案：

通过 RCC_CFGR3 的 TIMxSW 位实现。当选择 2x PLLCLK 时，SYSCLK/HCLK/PCLK 必须同频，即不经过任何分频。
RCC_CFGR3
Address: 0x30
Reset value: 0x0000 0000
[9]: TIM1SW, 1 = select 2x PLLCLK as TIM1 clock source
[24]: TIM2SW, 1 = select 2x PLLCLK as TIM2 clock source
[25]: TIM34SW, 1 = select 2x PLLCLK as TIM3/4 clock source

3.6.4 使用 HSE 时，时钟配置失败

程序配置时钟时，在启动 HSE 后，未等到 HSERDY，导致时钟配置失败。
原因为 FCM 103 的复位电压在 1.6V 左右，而部分晶振在此电压下比较难以起振，需等到 VDD 上升到一定值后才会有 HSERDY 信号，这样在 VDD 上升较慢时，1.6V 电压下 MCU 已经开始执行时钟配置程序并进行溢出超时计数，计数溢出前无 HSERDY 信号，因此时钟配置失败。

解决方案：

修改头文件中有关 HSE 溢出时间的定义为最大：
#define HSE_STARTUP_TIMEOUT ((uint16_t)0xffff) /*!< Time out for HSE start up */

3.6.5 HSI 温飘较大

在环境温度<0C 或>60C 时，内置 HSI OSC 误差范围可能超过 3%，会影响串口通讯。

解决方案：

环境温度范围要求较大时，使用外部晶振。

3.7 GPIO

3.7.1 GPIO 的 PIN2/PIN3 的特定设置组合会导致 PIN3 I/O 作为输入功能不对

每一组 I/O 的 PIN3（PA3/PB3…）在设置成通用输入且带上/下拉功能时，如果该组 I/O 的 PIN2（PA2/PB2…）的 MODE1 位为 1，则会导致 PIN3 作为 alternate function 输出，无法作为输入。

解决方案：

将 PIN2 的输出速度设置为 10MHz。

3.8 H103 使用 Flash Turbo（FLASH 加速）功能

在 FCM32H103 系列中，使用了 FLASH Turbo 模块代替了 F103 系列的 Prefetch-buf（预取指缓冲），其打开与关闭仍然使用 FLASH_ACR.PRFTBE 位控制，且其 cache 的操作是完全硬件化，无需软件干预。

3.9 FCM 器件识别

FCM32F1/H1 系列 MCU 型号，在信息区以下地址可以读到，以和其它厂家区别开来。

地址	内容	说明
0x1FFF_F7C0	0x46433332	‘FC32’ ASCII code
0x1FFF_F7C4	0x0046xxxv/ 0x0048xxxv	46 = ‘F’, 48 = ‘H’ xxx=type, eg. 103 v=version, eg. A/B/C

3.10 FLASH

3.10.1 信息区（information block）

FCMx103 中密度系列器件，其信息区大小和分配如下表：

Block	Name	Addesses	Size(bytes)
Information block	System memory	0x1FFF_E600 – 0x1FFF_F7FF	4.5k
	Option bytes	0x1FFF_F800 – 0x1FFF_F9FF	0.5k

3.10.2 读保护

FCM 的信息区不受读保护。即使开启了读保护，信息区仍然可读（System memory 可读，Option bytes 可读写）。

3.11 滚码烧写失败

某些型号的烧录器在启用滚码写入功能时，会出现滚码写入失败。

解决方案：

将滚码地址设在 FLASH 的前 16K 之内，并在源程序中保留该地址，初始化为全 FF，并重新编译。

例如滚码为 4 个字节，其起始地址为 0x080007fc，在源程序中加入以下代码：

```
const uint32_t roll_size[4] __attribute__ ((at(0x080007fc))) = {0xff,0xff,0xff,0xff};
```

3.12 RTC

3.12.1 RTC 闹钟标志 ALRF 不对

ALRF 标志位的置位发生在秒中断标志（SECF）置位时，因此，RTC 全局中断中的闹钟中断不可用。

3	10	settable	RTC	RTC global interrupt	0x0000_004C

解决方案：

使用通过 EXTI 的闹钟中断。

41	48	settable	RTCAlarm	RTC alarm through EXTI line interrupt	0x0000_00E4

3.12.2 RTC 闹钟时间间隔越来越长

例如闹钟溢出时间设置为 5ms，在每次闹钟溢出后，程序会重新设置下次闹钟溢出的时间（当前溢出时间+5ms），FCM 的溢出时间会按 5ms、10ms、15ms…的间隔溢出，只有第一次溢出是正确的。

原因为 RTC 计数器（RTC_CNT）的写入有一级 buf，在闹钟配置完成后（RTC_CRL->CNF 清 0 时）而 RTC_CNT 未写入的情况下，RTC_CNT 也会发生装载，而装载值为第一次对 RTC_CNT 的写入值，而不是当前的 CNT 继续计数。

解决方案：

闹钟溢出后，设置新的溢出时间时，将 RTC_CNT 读出再写入。

3.13 SPI

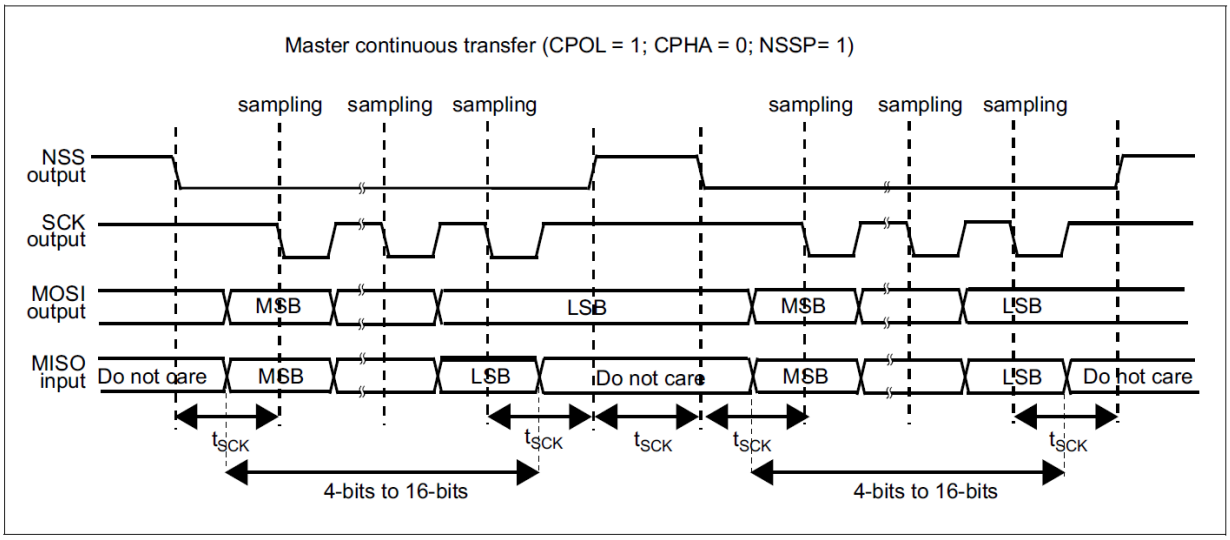
3.13.1 SPI 支持 NSSP 模式

NSSP 模式为两帧数据之间 NSS 插入无效电平。可将 NSSP 写 1 来允许该功能（仅在 FRF=0, CPHA=0 时有效，CPOL 忽略）。

SPIx_CR2[3] ：

NSSP ： 1=NSS 脉冲模式使能。复位=0。

NSSP 模式时序：



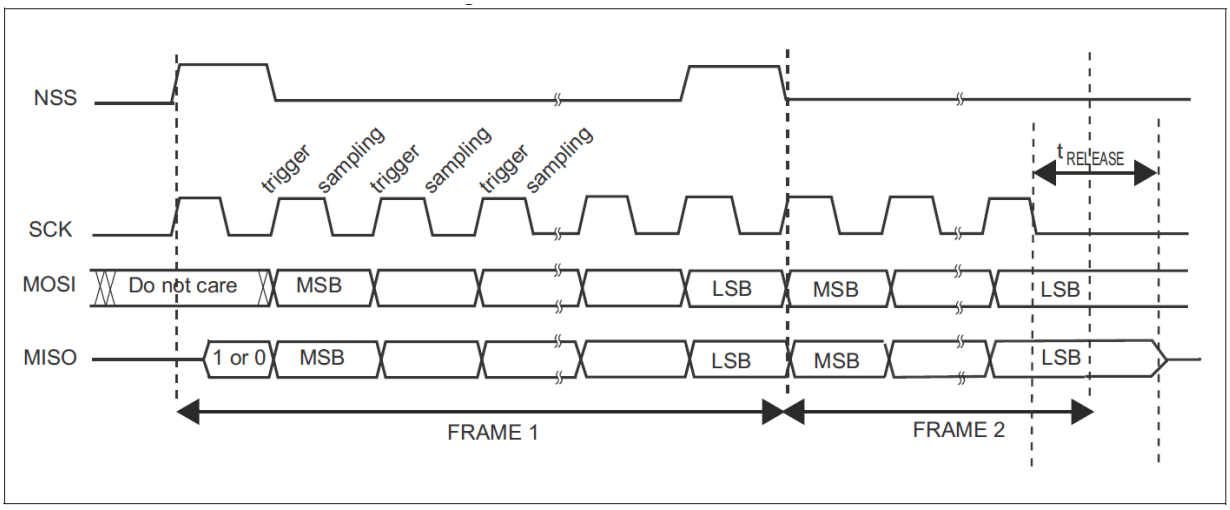
3.13.2 SPI 支持 TI 模式

将 FRF 写 1 来支持 TI 模式。

SPIx_CR2[4] ：

FRF ： 1=TI 模式使能。复位=0。

TI 模式时序：


NSS
SCK
MOSI
MISO
trigger
sampling
trigger
sampling
trigger
sampling
t RELEASE
Do not care
MSB
LSB
MSB
LSB
1 or 0
MSB
LSB
MSB
LSB
FRAME 1
FRAME 2

3.14 -

4 版本历史

Date	Revision	Author	Changes
2021/5/6	0.10	Dick Hou	初版，适用于 C 版 IC
2021/5/14	0.11	Dick Hou	第二节不同点，增加 I2S 相关内容； 三.7.2 节增加示范代码； 增加三.9 节
2021/5/28	0.12	Dick Hou	增加三.4 节
2021/7/13	0.13	Dick Hou	修改格式 增加三.7 节 增加三.10 节
2021/7/22	0.14	Dick Hou	三.4 节 CAN 增加时钟相关 增加三.11 节 FSMC
2021/8/9	0.15	Dick Hou	删除 FSMC 相关描述
2021/11/4	0.16	Dick Hou	增加 3.11 滚码烧写
2022/1/24	0.17	Dick Hou	增加 3.6.5 HSI 温飘的说明； 增加 3.12 RTC
2022/2/21	0.18	Dick Hou	增加 3.14 SPI 章节
2022/2/23	0.19	Dick Hou	增加 3.5.3 章节
2022/5/7	0.20	Dick Hou	增加 3.5.2 章节； 简化 3.5.3 章节

5 其它