

FCM32F0xx 应用相关注意事项**Ver 0.27**

适用型号	适用芯片版本	说明
F030xC	B C	
F051xx		
F070xx/F071xx/F072xx		
F091xx/F092xx/F095xx/F096xx		

FCM32F0xx 为深圳市闪芯微电子有限公司设计的基于 Cortex M0 内核的控制器，与 S**32F0xx 系列兼容，由于工艺不同等原因，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.....	5
2. 不同点.....	6
2.1 差异对比.....	6
2.2 功能增强.....	7
2.3 优化调试.....	7
3. 注意事项.....	7
4.1 器件识别.....	7
4.2 唯一 ID (UID)	7
4.3 ADC.....	8
4.3.1 ADC 输入阻抗	8
4.3.2 ADC 位数选择	8
4.3.3 ADC 低温精度下降	8
4.3.4 使用过 ADC 后, 低功耗模式功耗过大	8
4.4 DAC.....	9
4.4.1 DAC 数据未转换	9
4.4.2 PA4/PA5 漏电	9
4.4.3 DAC 仅开启 CH2 时未转换.....	9
4.5 COMP.....	10
4.6 OP	10
4.6.1 OP 出厂校准值装载不正确	10
4.6.2 定时器切换输入功能不可用	11
4.7 VBAT 供电	11
4.8 CAN.....	11
4.8.1 CAN 与 USBHD 同时使用.....	11
4.8.2 支持 CAN-FD	11
4.8.3 双 CAN	12
4.8.4 单 CAN 型号在某些应用中接收不正常	12
4.8.5 CAN 发送使用 FIFO 模式时发送错误.....	12
4.8.6 CAN 时钟精度要求	13
4.9 RCC	13
4.9.1 TIM1/2/3 输入时钟需要 2 倍 PLLCLK	13
4.9.2 使用 HSE 时, 时钟配置失败.....	13
4.9.3 RCC_CSR->IWDGRSTF 未置位	13
4.9.4 RCC_CFGR2->USBSW 位不起作用	13
4.9.5 单独使用 USBHUB 时, 内置 HSI48 无法通过 SOF 校准.....	14
4.10 SPI.....	14
4.10.1 SPI 增强功能	14
4.10.2 SPI 从机背靠背数据接收错误	16
4.10.3 8 位数据, 16 位打包且使用 DMA 时接收不全	16
4.11 I ² C	16
4.12 HAU	16
4.13 PLib	16
4.14 USART	17

4.15	USBHD	17
4.15.1	Device 模式端点配置成双缓冲时，某些程序工作不正常.....	17
4.15.2	作为 Host 使用.....	18
4.15.3	DP、DM 引脚不要串入电阻.....	18
4.16	USBHUB.....	18
4.16.1	USBHUB 外部下行端口可能无法使用	18
4.16.2	DP、DM 引脚不要串入电阻.....	18
4.17	SRAM.....	18
4.17.1	程序下载失败.....	18
4.17.2	程序执行错误.....	19
4.18	IWDG	19
4.19	RTC.....	19
4.19.1	MCU 进入 STANDBY 后，PC13/14/15 无法输出.....	19
4.19.2	MCU 进入 STANDBY 后，TAMP/TS 检测无法正常工作.....	19
4.20	POWER	20
4.20.1	STOP 模式功耗稍大.....	20
4.21	内置 Bootloader.....	20
4.22	DMA 重映射.....	20
4.23	-.....	20
4.	版本历史.....	21
5.	声明.....	22

1. 相同点

- M0 内核
- 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

2.1 差异对比

项目	S**32F0	FCM32F0
VDD/VDDA	2.0~3.6V	1.8~5.5V
工作温度范围	-40~85	-40~105
SRAM (F091xx)	32KB	24KB
Coremark 性能	103.8@48 MHz	103.7@48 MHz
CPU/AHB/APB 最高频率	48 MHz	72 MHz
TIM1/2/3 时钟	48 MHz	144 MHz
SPI 最高频率	24 MHz	36 MHz
UART 最高波特率	6 Mbps	9 Mbps
SPI 增强	x	✓
硬件除法运算	x	✓
硬件开方运算	x	✓
PLib	x	✓
CAN 数目	1	2
CAN 支持 FD-CAN	x	✓
CAN 滤波器数目	14 组	28 组 (单个 CAN 时)
USB HOST	x	✓
USB HUB	x	✓
LED 灯条控制接口	x	✓
OP	x	1
Boot loader	USART/IIC/USB	x
工作电流(HSI8MHz)	4.9mA	5.5mA
工作电流(HSI48MHz)	26.9mA	19.1mA
工作电流(HSI8+PLL72MHz)	-	26.6mA
Stop 电流(I _{DD} +I _{DDA})	6.5uA	18.3uA
Standby 电流(I _{DD} +I _{DDA})	3.3uA	18.0uA
内部 LDO 可以关闭	✓	x
独立 VBAT 供电	✓	x

注：

1. 以 S**32F072/F091 为参考。
2. 测试条件为 3.6V/25C, LSI/IWDG OFF, VDDA Monitor ON。
3. S**32F0 根据型号不同, 支持从不同的通讯口 boot, 不是所有器件都支持表格里的接口。
4. HSI8MHz 的工作电流, 排除了 USBHUB 外设。

2.2 功能增强

FCM32F0xx 相较于 S**32F0xx，有以下功能增强：

- ✓ 更宽的工作电压范围
- ✓ 更高的工作频率
- ✓ 更低的功耗
- ✓ 通讯接口更快的通讯速度
- ✓ SPI 增强，适应更多应用场景
- ✓ 硬件除法及开方，弥补 M0 内核的劣势
- ✓ USB 和 CAN 相互独立，可以同时使用
- ✓ 支持 CAN-FD
- ✓ 支持 USB HOST、USB HUB
- ✓ 支持 LED 灯条驱动
- ✓ 集成 OP

2.3 优化调试

S**32F0xx 部分外设标志位具有硬件自动清除或置位（例如 USART 的 RXNE，MCU 读取 DR 后自动清 0）功能，这样导致在调试时，如果打开了寄存器窗口，会由于寄存器窗口的刷新，导致这些标志位被清除，使得 MCU 的程序运行受到影响。FCM32 系列 MCU 采取了优化措施，屏蔽了调试器对寄存器读取时清除标志位，从而不影响 MCU 程序执行。

3. 注意事项

4.1 器件识别

FCM32F0 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_F790	0x46433332	'FC32' ASCII code
0x1FFF_F794	0x0046xxxv/ 0x0048xxxv	46 = 'F', 48 = 'H' xxx=type, eg. 030 v=version, eg. A/B/C

4.2 唯一 ID (UID)

FCM32F0 全系都包含唯一 ID，其地址在 0x1FFF_F7AC 开始的连续 3 个字（96 位）。

4.3 ADC

4.3.1 ADC 输入阻抗

部分应用中直接将 S** 的二进制烧录，ADC 精度不够，因为 ADC 的电气特性与 S** 不尽相同，相同的采样时间，FCM 的输入阻抗 (R_{AIN}) 要低，规格书中已列出 R_{AIN} 表格。

解决方案:

- A. 根据外部输入阻抗，调整采样时间
- B. 降低外部输入阻抗

3.2.2 ADC 位数选择

ADC 支持数据位设为 6/8/10/12 位，但 ADC 转换时间不会随位数减少而减少，转换时间固定为 12 位数据位的转换时间。

3.2.3 ADC 低温精度下降

ADC 在低于负 25C 左右后，精度会明显下降。

解决方案:

在极低温时如果需要高精度模拟信号测量，该 IC 可能满足不了。如果需要在极低温时监测某个电压，可使用内置的比较器来实现。

3.2.4 使用过 ADC 后，低功耗模式功耗过大

ADC 配置为异步时钟，在进入低功耗前，运行了 ADC 的停止及去初始化程序：

```
HAL_ADC_Stop(&AdcHandle);
```

```
HAL_ADC_DeInit(&AdcHandle);
```

但功耗仍然偏大，比正常功耗大概高出 1mA 左右。

解决方案 (二选一):

- ADC 时钟初始化为同步时钟

```
// AdcHandle.Init.ClockPrescaler      = ADC_CLOCK_ASYNC_DIV1;  
AdcHandle.Init.ClockPrescaler         = ADC_CLOCK_SYNC_PCLK_DIV4;  
AdcHandle.Init.Resolution              = ADC_RESOLUTION_12B;  
AdcHandle.Init.DataAlign               = ADC_DATAALIGN_RIGHT;
```

- 进入低功耗前，复位 ADC 模块

```
RCC->APB2RSTR |= RCC_APB2RSTR_ADC1RST;
```

```
RCC->APB2RSTR &= ~RCC_APB2RSTR_ADC1RST;
```

4.4 DAC

4.4.1 DAC 数据未转换

在 DAC 通道未使能前，数据未转换。原因为 FCM 的设计上，DOR 寄存器需要在通道使能（DAC->CR.ENx）打开时才会装载，程序如果未打开 DAC 通道而先写入数据，则数据不会装载至 DOR。

解决方案：

先允许 DAC 的通道，再写入数据。

4.4.2 PA4/PA5 漏电

在 DAC 未开启时，PA4/PA5 引脚有错误的对地导通回路（对地有一小 NMOS）。由此导致，当该引脚输出/输入高电平时，会有~300uA 漏电流；如果引脚相应内部上拉打开，则会有~70uA 漏电流，且不能正确的上拉至高电平。

解决方案：

PA4/PA5 不使用；或仅用于 DAC 引脚。

C 版已修正。

4.4.3 DAC 仅开启 CH2 时未转换

如果 DAC CH1 触发（DAC_CR_TEN1 位域）未使能，会导致 CH2 不会触发。

解决方案：

如果启用了 CH2，即使 CH1 未启用，也需要将 DAC_CR_TEN1 置 1。

```

static void DAC_Config(void)
{
    /* Configure the DAC peripheral instance */
    DacHandle.Instance = DAC1;

    /*##-1- Initialize the DAC peripheral #####*/
    if (HAL_DAC_Init(&DacHandle) != HAL_OK)
    {
        /* Initiliazation Error */
        Error_Handler();
    }

    /*##-1- DAC channel Configuration #####*/
    sConfig.DAC_Trigger = DAC_TRIGGER_T6_TRGO;
    sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_ENABLE;

    if (HAL_DAC_ConfigChannel(&DacHandle, &sConfig, DAC_CHANNEL_2) != HAL_OK)
    {
        /* Channel configuration Error */
        Error_Handler();
    }

    DAC1->CR |= DAC_CR_TEN1;
    /*##-2- Enable DAC Channel and associeted DMA #####*/
    if (HAL_DAC_Start_DMA(&DacHandle, DAC_CHANNEL_2, (uint32_t*)Sine12bit,
                          sizeof (Sine12bit) / sizeof (uint32_t),
                          DAC_ALIGN_12B_R) != HAL_OK)
    {
        /* Start DMA Error */
        Error_Handler();
    }
}

```

Add this line

4.5 COMP

比较器如果选择连接到了内部 VREF，内部 VREF 不会自动打开。

解决方案：

将 ADC->CCR.VREFEN 置 1。

4.6 OP

OP 使用方法说详见相关文档。OP 功能兼容 S**32F303 的规格。

4.6.1 OP 出厂校准值装载不正确

OP 出厂已校准过，但不会自动装载校准值。

解决方案：



用户可手动装载一下校准值。将系统信息区的 0x1FFFF7D4 的 TRIM 值，写入 OPAMP1->CSR 的相应位，再将 CSR->USER_TRIM 设为 1 即可。

0x1FFFF7D4:

[4:0] – TRIMOFFSETN[4:0]

[12:8] – TRIMOFFSETP[4:0]

4.6.2 定时器切换输入功能不可用

定时器切换输入是通过 TIM1 CC6 来实现，但 TIM1 为 4 通道，没有 CC6。

4.7 VBAT 供电

FCM32F0 系列 MCU 不提供掉电模式的后备电池供电，当主 VDD 不提供后，VBAT 在 IC 内部由二极管会继续向 VDD 供电。因此，对功耗在乎的应用，应该采取其它供电措施。

4.8 CAN

4.8.1 CAN 与 USBHD 同时使用

CAN 与 USBHD 未共享 RAM，且 CAN 的 RAM 没有分配 MCU 直接地址，因此，CAN 和 USBHD 没有冲突，可同时使用！

注：

1. 因为 CAN 的 RAM 不能由 MCU 直接地址访问，因此在调试界面无法直接查看 CAN 的 RAM 内容。
2. USBHD 包含 2KB RAM，地址为 2 个连续的 1KB。其后 1KB RAM 地址和 CAN1 地址重叠。因此，如果同时使能了 USBHD 和 CAN1，则硬件上屏蔽了 USBHD 的后 1KB RAM，USBHD 只允许使用前 1KB RAM。
3. 如果不使用 CAN1，同时使能 USBHD 和 CAN2，则 USBHD 可使用 2KB RAM。

4.8.2 支持 CAN-FD

exCAN 支持 CAN-FD，请参考 FCM 提供的相应 exCAN 的文档。该 exCAN 完全兼容 G*32C113 的规格。

4.8.3 双 CAN

FCM32F0 包含双 CAN 的型号，完全兼容 S**32F105/107，双 CAN 共享 28 个滤波器。

4.8.4 单 CAN 型号在某些应用中接收不正常

因为 FCM32F0 系列有包含 2 个 CAN 的型号，因此 FMR 寄存器包含 CAN2SB 位段（即使是单 CAN 的型号）。如果用户程序对该位段直接写 0，会导致 CAN1 工作不正常。

解决方案：

即使在使用单 CAN 时，也要保证 FMR->CAN2BS 写入正确值（0x0E 为复位默认值，该值将 CAN1 配置为 14 个过滤器）。

4.8.5 CAN 发送使用 FIFO 模式时发送错误

在 FIFO 模式，用户未按 TSR->CODE 指示的空闲邮箱来填充发送数据，而是每次总是从邮箱 0 开始判断是否空闲，按 0、1、2 空闲判断顺序写入邮箱，导致发送错误。

解决方案：

按 TSR->CODE 指示来填充邮箱，或采用队列模式（MCR->TXFP=0）。

```
static CAN_TxMailBox_TypeDef *can_find_free_mailbox(can_data_t *hcan)
{
    CAN_TypeDef *can = hcan->instance;
    uint32_t tsr = can->TSR;

#ifdef TM_CODE
    if ( tsr & CAN_TSR_TME0 ) {
        return &can->sTxMailBox[0];
    } else if ( tsr & CAN_TSR_TME1 ) {
        return &can->sTxMailBox[1];
    } else if ( tsr & CAN_TSR_TME2 ) {
        return &can->sTxMailBox[2];
    } else {
        return 0;
    }
#else
    if ( tsr & (CAN_TSR_TME0 | CAN_TSR_TME1 | CAN_TSR_TME2) ) {
        return &can->sTxMailBox[(tsr & CAN_TSR_CODE) >> 24];
    } else {
        return 0;
    }
#endif
}
```

注：

使用 S** 固件没有问题，以上代码示例为某自制固件。

4.8.6 CAN 时钟精度要求

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

如果在包含 USB 的应用中，开启了时钟校准功能，且 USB 正常连接至 PC，则可以使用内部 RC OSC 作为时钟源，因为此种应用 RC OSC 被实时校准至<0.25%的误差。

4.9 RCC

4.9.1 TIM1/2/3 输入时钟需要 2 倍 PLLCLK

TIM1/2/3 外设可以使用 2*PLLCLK 作为定时器的时钟，方法如下：

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。具体的位分配请参见规格书。

4.9.2 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 FCM32 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 */
```

（以上修改仅限 HSE 溢出等待采用软件循环。如果该常量定义的是 ms 数值，则适当加大即可。）

该故障的典型现象是：上电后无任何反应，将 MCU 的 nRST 复位引脚对地短接一下，则 MCU 正常工作。

4.9.3 RCC_CSR->IWDGRSTF 未置位

当 IWDG 溢出时，IWDGRSTF 未置位。

4.9.4 RCC_CFGR2->>USBSW 位不起作用

USBSW 位永远为 0，无法修改，使得 USB 时钟总是为 HSI48。

解决方案：

程序总是使用 HSI48 为 USB 时钟。

4.9.5 单独使用 USBHUB 时，内置 HSI48 无法通过 SOF 校准

因为 CRS 的 USB SOF 校准是使用的 USB1 的信号，因此，只使用 USBHUB 时，无法校准 HSI48，因此其误差可能超过 USB 规定的 0.25%。

解决方案：

此种情形 CRS 的校准源应该选择其它信号。

4.10 SPI

4.10.1 SPI 增强功能

SPI 增加了控制寄存器 CR3，使其时序可以匹配更多应用。

SPI_CR3

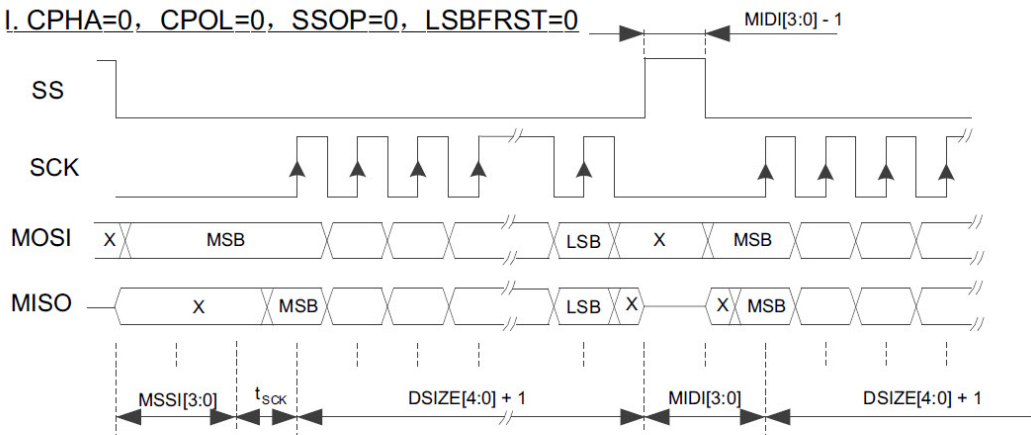
Address: 0x24

Reset value: 0x0000 0000

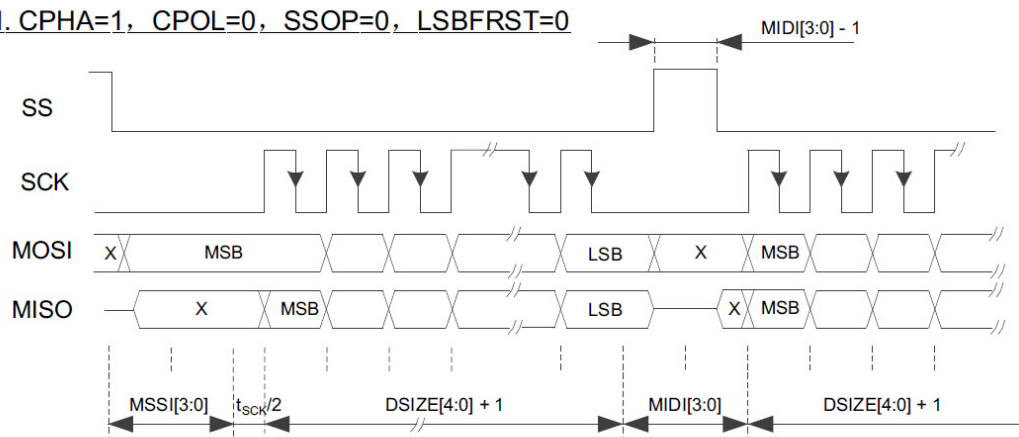
- [9]: SSOM, 主模式 SS 输出管理 (SS output management in master mode)
SS 输出允许时，该位用于主模式。它允许两个连续传输之间配置 SS 输出。
0: SS 保持为有效电平，直到数据传输完成，随即会通过 EOT 标志变为无效状态
1: 当 MIDI[3:0]>1 时，SPI 数据帧与 SS 非有效脉冲交错
- [8]: SSOP, SS 输出极性 (SS output polarity)
该位仅在主模式下使用
0: SS 信号低电平有效
1: SS 信号高电平有效
- [7:4]: MSSl, 主模式 SS 空闲 (Master SS Idleness)
指定在 SS 输出时在主模式下额外插入 SS 有效边沿和第一个数据事务开始之间的额外延迟 (用 SPI 时钟周期数表示)
- [3:0]: MIDl, 主模式数据间空间 (Master Inter-Data Idleness)
指定主模式下在两个连续数据帧之间插入的最小时间延迟 (以 SPI 时钟周期表示)

相关波形示意如下图：

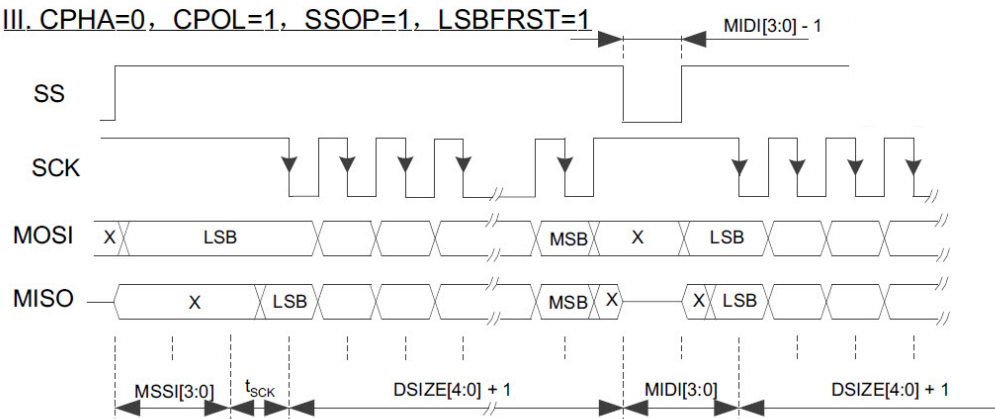
I. CPHA=0, CPOL=0, SSOP=0, LSBFRST=0



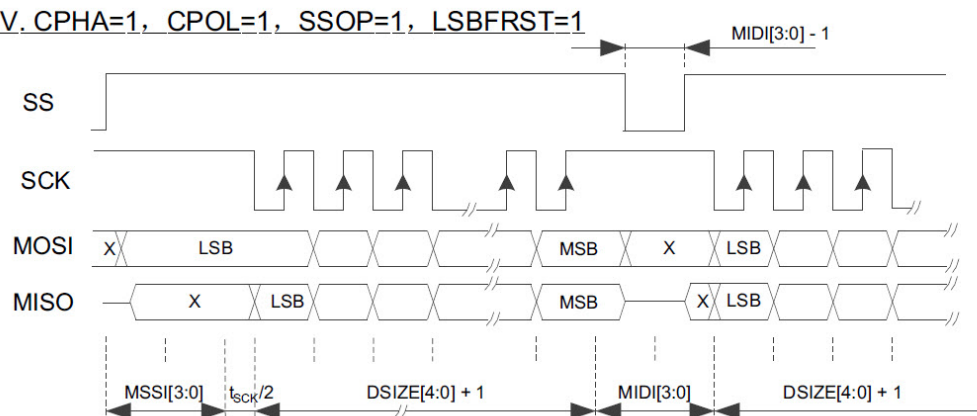
II. CPHA=1, CPOL=0, SSOP=0, LSBFRST=0



III. CPHA=0, CPOL=1, SSOP=1, LSBFRST=1



IV. CPHA=1, CPOL=1, SSOP=1, LSBFRST=1



4.10.2 SPI 从机背靠背数据接收错误

当 SPI 工作在从机模式，SCK 为主频/2，且接收数据为背靠背（字节之间没有空隙）时，接收错误。

解决方案：

要求 $SCK \leq \text{主频}/3$ 。

4.10.3 8 位数据，16 位打包且使用 DMA 时接收不全

当 SPI 通讯为 8 位数据格式，采用 DMA 收发，且 DMA 配置为 16 位，当接收数据字节数为奇数时，最后一个字节无法产生 DMA 请求，导致 DMA 完成中断无法产生。

解决方案：

DMA 配置为 8 位数据大小。

4.11 I²C

I²C 模拟滤波器未起作用。

4.12 HAU

此系列 MCU 增加了硬件除法和整数开方运算单元 HAU，具体用法详见相关文档。

4.13 PLib

私有库 PLib 可保护重要代码不被读出，具体用法详见相关文档。

PLib 在启用后，如果移除 PLib，MCU 的系统 FLASH 区（包含配置信息）会被误擦除，因此 MCU 不能再使用。

解决方案：

PLib 一旦启用，不要移除。

4.14 USART

USART 工作在 LIN 模式时，通过 USART_CR2->LBDF 位选择 10 位或 11 位的 BREAK 检测，但芯片存在 bug，实际检测要少一位，即检测到连续的 9 位或 10 位低电平视为 BREAK。因此，在将 LBDL 设置为 0（检测 10 位）时，会将正常数据 0 视为 BREAK（1 位起始位+8 位数据位=9 位 0）。

解决方案：

将 USART_CR2->LBDF 设置为 1。

4.15 USBHD

4.15.1 Device 模式端点配置成双缓冲时，某些程序工作不正常

端点在双缓冲模式，依靠 DTOG 和 SW_BUF 位来定义缓冲标志：

Buffer flag	'Transmission' endpoint	'Reception' endpoint
DTOG	DTOG_TX (USB_EPnR bit 6)	DTOG_RX (USB_EPnR bit 14)
SW_BUF	USB_EPnR bit 14	USB_EPnR bit 6

应用程序和 SIE 依靠标志位的组合来确定缓冲的使用情况：

Endpoint Type	DTOG	SW_BUF	Packet buffer used by USB Peripheral	Packet buffer used by Application Software
IN	0	1	ADDRn_TX_0 / COUNTn_TX_0 Buffer description table locations.	ADDRn_TX_1 / COUNTn_TX_1 Buffer description table locations.
	1	0	ADDRn_TX_1 / COUNTn_TX_1 Buffer description table locations.	ADDRn_TX_0 / COUNTn_TX_0 Buffer description table locations.
	0	0	None ⁽¹⁾	ADDRn_TX_0 / COUNTn_TX_0 Buffer description table locations.
	1	1	None ⁽¹⁾	ADDRn_TX_0 / COUNTn_TX_0 Buffer description table locations.
OUT	0	1	ADDRn_RX_0 / COUNTn_RX_0 Buffer description table locations.	ADDRn_RX_1 / COUNTn_RX_1 Buffer description table locations.
	1	0	ADDRn_RX_1 / COUNTn_RX_1 Buffer description table locations.	ADDRn_RX_0 / COUNTn_RX_0 Buffer description table locations.
	0	0	None ⁽¹⁾	ADDRn_RX_0 / COUNTn_RX_0 Buffer description table locations.
	1	1	None ⁽¹⁾	ADDRn_RX_1 / COUNTn_RX_1 Buffer description table locations.

某些应用程序在启用双缓冲端点前，将 DTOG 和 SW_BUF 位配置成同一个值，导致 USB 的相应端点通讯错误，现象为双缓冲端点只收、发一帧数据，后续数据帧无法进行。

解决方案：

将 DTOG 和 SW_BUF 初始化成相反的值。

4.15.2 作为 Host 使用

USBHD 为 2.0 FS 规格，支持 Host/Device。Host 功能完全兼容 S**32G0xx 规格。

4.15.3 DP、DM 引脚不要串入电阻

IC 内部已串入阻抗匹配电阻，不要在外部再串入！

4.16 USBHUB

USBHUB 为全速，支持 5 个内置设备。具体用法详见相关文档。

4.16.1 USBHUB 外部下行端口可能无法使用

当 USBHUB 开启外部端口时，如果 CRS 的时钟源选择的是 USB SOF，或未开启 CRS，外部端口可能无法使用。

这是因为章节 4.9.5 的原因，HSI48 不够精确。

解决方案：

使用 USB SOF 以外的信号源通过 CRS 校准内部 HSI48。

4.16.2 DP、DM 引脚不要串入电阻

IC 内部已串入阻抗匹配电阻，不要在外部再串入！

4.17 SRAM

F030xC/F09xx 的型号，FCM 的 SRAM 为 24KB，而 S**T 为 32KB，需要注意此差异的影响。

4.17.1 程序下载失败

在对 FLASH 编程时，如果工具的编程算法使用了 32KB 的 SRAM（例如 S*Link Utility，或在 IAR 开发环境中使用了 S**T 的支持包），则可能编程出错。

解决方案:

使用 FCM 提供的支持包。

4.17.2 程序执行错误

如果原程序使用了超过 24KB 的 SRAM，则可能会得到错误运行结果。

解决方案:

将 SRAM 大小修改为 24KB 再重新编译；或使用 FCM 提供的支持包并重新编译。

4.18 IWDG

IWDG 在计数至 0x800 以下时，如果重设装载值 $\geq 0x800$ ，则会产生 IWDG 复位。

解决方案:

设置小于 0x800 以下的重装载值；或是在 IWDG 计数至 $<0x800$ 之前进行重载。

4.19 RTC

4.19.1 MCU 进入 STANDBY 后，PC13/14/15 无法输出

在 STANDBY 后，PC13/14/15 的 I/O 功能被关闭，因此无法输出。

解决方案:

STANDBY 模式不使用 PC13/14/15 作为输出。

4.19.2 MCU 进入 STANDBY 后，TAMP/TS 检测无法正常工作

在 STANDBY 后，TAMP/TS 引脚被配置成模拟输入，其输入至内部电路的数字信号被强制成低电平。因此，如果 TAMP 配置成低电平触发，则总是不停唤醒；如果配置成高电平或边沿触发，则总是无法唤醒。

解决方案:

如果要在 STANDBY 使用 TAMP 检测来唤醒 MCU，可以改成 WAKEUP 唤醒。

4.20 POWER

4.20.1 STOP 模式功耗稍大

在 STOP 模式，即使 LDO 被配置成低功耗（LP）模式，休眠电流和 LDO 非低功耗模式没有变化。原因为 LDO 的 LP 模式无效。

解决方案：

无。

4.21 内置 Bootloader

此版本芯片无法在上电或复位时进入内置的 USART bootloader。

解决方案：

无。

4.22 DMA 重映射

TIM17 DMA 默认连接至 CH1，重映射至 CH2 无效；TIM16 DMA 默认连接至 CH3，重映射至 CH4 无效。

解决方案：

当 CH1/2 已占用时，可以将 TIM17 映射至 CH7，TIM16 映射至 CH6。

4.23 -

4. 版本历史

Date	Revision	Author	Changes
2023/4/19	0.10	Dick Hou	初版
2023/5/4	0.11		* 删除 3.8.3 HSI 温漂 * 增加 3.8.3 HSI/HSI48 * 增加 3.8.5 USB SW 位 * 删除 3.9 USART * 增加 3.15 FLASH 编程 * 增加 3.4.2 DAC 数据未转换
2023/5/5	0.12		* 增加 3.5 COMP，后续章节编号顺移 * 增加 3.4.1 BUF 使能控制
2023/5/11	0.13		* 增加 3.4.3 DAC 导致芯片静态功耗过大 * 增加 3.17 IWDG * 增加 3.18 RTC * 增加 3.6.1 OP 校准说明 * 增加 3.19 POWER * 增加 4.10.2 SPI 从机背靠背接收错误
2023/5/29	0.14		4.13 PLIB 增加移除的说明
2023/6/5	0.15		* 适配 B 版 * 删除不需要的章节
2023/6/26	0.16		插入 4.14 USART
2023/6/30	0.17		* 4.8.6 增加 CAN 使用内部 RC OSC 作时钟源的说明 * 4.17 修改为 SRAM 章节
2023/7/4	0.18		增加 4.21 内置 bootloader
2023/7/24	0.19		增加 4.4.2 PA4/PA5 漏电
2023/7/25	0.20		增加 3.2.3 ADC 低温精度下降
2023/8/17	0.21		增加 4.6.2 定时器切换输入不可用
2023/9/14	0.22		增加 4.22 DMA 重映射
2023/9/20	0.23		增加 4.9.5 增加 4.16.1
2024/5/13	0.24		增加 4.10.3 增加 4.15.3 增加 4.16.2
2024/5/24	0.25		增加 4.4.3
2024/5/29	0.26		* 增加 2.2 节 * 增加 2.3 节
2024/8/7	0.27		增加 3.2.4 节

5. 声明

闪芯微电子有限公司（下称“闪芯微”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的闪芯微产品（下称“产品”）为公司所有。

闪芯微在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

闪芯微保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用者在下单购买前联系闪芯微获取此文档的最新版本。

闪芯微竭力提供准确可信的资讯，但即便如此，并不推定闪芯微对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，闪芯微不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

闪芯微对于产品在系统或设备中的应用效果没有任何故意或保证，如果任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用应被视为“不安全使用”。不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使闪芯微免于因为这类不安全使用而导致被诉、支持费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证责任，闪芯微可以法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。