



复旦微电子

FM33FG0xxA 车用系列 MCU 软件开发注意事项

应用笔记

V1.0.1



本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。

在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。

采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。

未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站(<http://www.fmsh.com/>)。

如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。

商标

上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。



目 录

1 说明.....	3
2 驱动库.....	3
3 上下电处理注意事项.....	3
4 独立看门狗 (IWDT)	3
5 电源电压监测 (SVD)	3
6 复位管理单元 (RMU)	4
6.1BOR 下电复位	4
7 总线与存储.....	4
7.1Flash 读取等待	4
7.2Flash 擦写/编程	5
7.2.1 软件流程.....	5
7.2.2 复位.....	5
7.2.3 程序延迟.....	6
7.2.4 工作时钟源.....	6
7.3 ECC	6
8 时钟管理单元 (CMU)	7
8.1 时钟源启动稳定时间.....	7
8.2 RCHF 调校.....	7
8.3 XTHF 停振检测及软件冗余措施.....	8
8.4 PLL	8
8.5 LSCLK 配置	9
9 电源管理单元 (PMU)	9
9.1 Deep Sleep 配置	9
9.2 模拟 BUFFER 控制寄存器(PMU_BUFCCR)	9
10 I/O 端口 (GPIO)	10
10.1 FM33FG0x5A (48PIN) 用户代码要求	10
10.2 外部引脚中断(EXTI)	10
10.3 GPIO 上拉和下拉.....	11
10.4 GPIO 驱动强度.....	11



10.5 SWD 引脚.....	11
11 经典控制器区域网络（FSCAN）	11
12 灵活速率控制器区域网络（RS-CANFD-Lite）	11
13 通用异步收发传输器（UART）	12
13.1 硬件 LIN.....	12
14 模数转换器（ADC）	13
15 数模转换器（DAC）	14
16 时钟监控（CLM）	14
17 直接存储访问控制器（DMA）	14
18 定时器模块.....	15
19 SPI 模块.....	15
20 调试支持.....	16
21 地址对齐访问	16
22 HardFault_Handler	16
23 NMI_Handler	16
版本信息.....	18
上海复旦微电子集团股份有限公司销售及服务中心	19



1 说明

本文档为 FM33FG0xxA 车用系列 MCU 的应用笔记，用于描述客户在使用该系列 MCU 进行产品开发时应在嵌入式软件方面所注意的事项。

2 驱动库

用户在产品开发前请到复旦微开发者论坛更新最新版 FM33FG0xxA 驱动库，里面包含《CMSIS》和《FM33FG0xxA_FL_Driver》两个库文件。

复旦微开发者论坛地址：<http://www.fmdevelopers.com.cn>

3 上下电处理注意事项

用户在产品开发前请务必仔细阅读《FM33 系列 MCU 上下电处理注意事项》应用笔记，它能有效提高软件在上下电过程中的工作可靠性。

4 独立看门狗（IWDG）

用户软件应在进入 main() 后的第一步就配置 IWDG 的目标溢出周期并使能，从而确保对后面执行的软件起到监控目的。用户软件有责任按照应用程序的要求，以所需的溢出/超时周期插入程序流执行清狗服务程序，但在定时中断服务中清狗是要避免的。

如用户使用 MCU 休眠功能，则推荐用户定时唤醒 MCU 执行清狗服务程序，定时唤醒源可以使用定时器、RTC 等。当然 MCU 也提供在休眠模式下冻结 IWDG 计数功能，用户可通过对 IWDG_CR 寄存器的 FREEZE 位写 1 使能此功能，使能后 MCU 在休眠模式下 IWDG 停止计数。

5 电源电压监测（SVD）

用户软件在完成 IWDG 初始化后，应立即启用 SVD 以监控 MCU 电源是否上升到目标电压阈值（推荐 4.2V），在未上升到目标电压阈值前，不建议继续初始化 MCU 的其他外设或执行应用功能，以避免在低电压状态下运行所带来的风险。

MCU 上电正常运行后用户软件同样应使用 SVD 对 MCU 的电源电压进行掉电监测,当低于目标阈值时(推荐 4.2V),用户软件应有保护性设计,以防止应用程序在低电压状态下运行。详细流程及软件实现请参考《FM33 系列 MCU 上下电处理注意事项》应用笔记和 FM33FG0xxA 标准例程。

6 复位管理单元 (RMU)

6.1 BOR 下电复位

上电过程中用户软件使用 SVD 监测到 MCU 电源电压大于目标阈值后,应及时配置复位管理单元的 BOR 下电复位电压阈值(推荐 falling 2.0v reset)并保持使能,即使进入 MCU 休眠模式也禁止关闭。

先监测电源电压再配置 BOR 的目的是防止 MCU 在上电过程中振荡复位。

详细流程及软件实现请参考《FM33 系列 MCU 上下电处理注意事项》应用笔记和 FM33FG0xxA 标准例程。

7 总线与存储

7.1 Flash 读取等待

MCU 复位启动后系统时钟源默认为 RCHF 8MHz,用户软件在配置系统时钟之前必须确保 Flash 读取控制寄存器 (FLS_RDCR) 满足如下要求:

➤ Code Flash 读等待周期配置 (CODE_WAIT):

当目标 CPU 主频小于等于 24MHz 时,不需要开启 wait;

当目标 CPU 主频大于 24MHz 小于等于 48MHz 时,提前配置 1wait;

当目标 CPU 主频大于 48MHz 小于等于 64MHz 时,提前配置 2wait;

禁止目标 CPU 主频大于 64Mhz;

➤ Data Flash 读等待周期 (DATA_WAIT):

基于目标 CPU 主频,提前配置 DATA_WAIT 等待周期,确保 CPU 访问 Data Flash 时钟小于等于 12.5MHz;



典型错误示例：

- (1) MCU 上电默认系统时钟是 RCHF 8MHz，CODE_WAIT 为 0；
- (2) 先配置系统时钟为 64MHz；
- (3) 再配置 CODE_WAIT 为 2；

分析：

在 (2) 步骤执行完成但 (3) 步骤还未执行时刻，系统时钟为 64MHz，CODE_WAIT 为 0，此时 CPU 继续访问 Code Flash 取指，但此刻访问时钟已超过 24MHz，会存在取指错误并导致程序无法正确运行。

7.2 Flash 擦写/编程

7.2.1 软件流程

软件实现流程请严格参考复旦微 FM33FG0xxA 标准例程关于 Code Flash 和 Data Flash 的擦写示例，为提高 Flash 数据安全，如下要求是要遵守的：

- (1) 每次擦写/编程完毕都要及时 Flash Key 锁定；
- (2) 每次擦写/编程完毕都要及时关闭 Flash 的操作时钟源；
- (3) 每次擦写/编程完毕都要及时关闭 Flash 寄存器的访问时钟源；
- (4) 针对 Code Flash 每次擦写/编程前禁止全局中断，确认擦写/编程完毕后恢复全局中断；

7.2.2 复位

在对 Code Flash 和 Data Flash 执行擦写/编程时刻如发生全局复位，则会对 Flash 的机器码或数据产生不可预知的后果（如误改变 Flash 机器码或数据，但仅限于 Code Flash 对 Code Flash、Data Flash 对 Data Flash 的影响，不会交叉影响），

因此如用户软件使用到 Flash 擦写/编程功能，用户有责任对软件执行如下排查，且是强制要求：

- (1) 执行 Flash 擦写/编程时刻，不会存在看门狗复位的可能性；



(2) 执行 Flash 擦写/编程时刻，不会存在 NRST 被拉低复位的可能性；

(3) 执行 Flash 擦写/编程时刻，如系统电源掉电，软硬件方面有机制确保不会与 BOR/PDR 下电复位发生冲突；

7.2.3 程序延迟

CPU 通过 AHB 总线对 Code Flash 访问，当 CPU 对 Code Flash 执行擦写/编程操作时，Code Flash 外设数据总线被占用，读取即取指操作将会暂停，直至擦写/编程完成，因此需评估 Code Flash 擦写时间对应用层的影响，可将实时性要求高的程序移植至 RAM 运行。

当 CPU 对 Data Flash 执行擦写/编程操作时，不会影响 Code Flash 的程序正常运行，如中断服务的响应等。

7.2.4 工作时钟源

Code Flash 和 Data Flash 擦写/编程工作时钟源由 MCU 固定为 RCHF，因此软件禁止关闭。

7.3 ECC

FM33FG0xxA 的 RAM、Code Flash 和 Data Flash 都支持 ECC，可实现纠 1 检 2 (SEC-DED)，上电后 ECC 默认开启，软件无法关闭。

当 ECC 检测到 2bit 及以上错误时会触发 NMI_Handler 中断服务，由于 NMI_Handler 属于不可屏蔽中断，因此用户软件**必须定义 void NMI_Handler(void) 中断并在服务中清除 ECC 检错标志**，从而避免 ECC 触发 NMI 后程序死机，具体执行可参考复旦微 FM33FG0xxA 标准例程。



8 时钟管理单元 (CMU)

必须在使能 IWDG、MCU 电源电压大于 SVD 监测阈值和使能 BOR 之后才允许配置 CMU 单元。

8.1 时钟源启动稳定时间

(1) 如使用 XTLE, 必须参考元器件手册技术参数配置启动等待时间 (CMU_XTLFCR: WAIT), 同时用户软件必须判断 XTLE 稳定标志 (CMU_XTLFCR: RDY) 置起后才能继续执行。

(2) 如使用 XTHF, 必须参考元器件手册技术参数配置启动等待时间 (CMU_XTHCFR: WAIT) 和振荡强度 (CMU_XTHCFR: HF_CFG), 同时用户软件必须判断 XTHF 稳定标志 (CMU_XTHCFR: RDY) 置起后才能继续执行, 注意 XTHF 的振荡强度值必须大于等于 0x8。

(3) 如使用 PLL, 使能之前必须判断 PLL 使用的时钟源已稳定工作, 使能 PLL 后用户软件必须判断 PLL 锁定标志 (CMU_PLLCR: LOCKED) 置起后才能继续执行。

(4) 在选择系统时钟源 (CMU_SYSCCLKCR: SYSCLESEL) 之前, 用户软件必须判读时钟源已稳定工作。

8.2 RCHF 调校

RCHF 是 MCU 内部的高频 RC 振荡器, 典型振荡频率为 8/16/24/32MHz, 为确保全温度范围的频率精度, 每个频率类型都需要调用厂内标定的调校值写到 RCHF 调校寄存器 (CMU_RCHFTR), 具体如下:

使用 RCHF 8MHz 时: CMU->RCHFTR = RCHF8M_TRIM;

使用 RCHF 16MHz 时: CMU->RCHFTR = RCHF16M_TRIM;

使用 RCHF 24MHz 时: CMU->RCHFTR = RCHF24M_TRIM;

使用 RCHF 32MHz 时: CMU->RCHFTR = RCHF32M_TRIM;

MCU 复位启动后 RCHF 缺省为 8MHz, SystemInit() 函数会执行 CMU->RCHFTR = RCHF8M_TRIM 语句, 当用户切换到其他频率值时, 需用户软件将正确的调校值写到 RCHF 调校寄存器。

当用户软件使用到 MCU 休眠功能时, MCU 唤醒后时钟源默认使用 RCHF, 但频率值由 PMU_CRWKFSEL 寄存器决定, 因此如 Active 运行的 RCHF 频率与唤醒后的 RCHF

频率不一致，则需用户软件及时切换频率，并在频率切换后将正确的调校值写到 RCHF 调校寄存器。

11:10	WKFSSEL	Sleep/DeepSleep 唤醒后的系统频率 00: RCHF-8MHz 01: RCHF-16MHz 10: RCHF-24MHz 11: RFU
-------	---------	--

8.3 XTHF 停振检测及软件冗余措施

MCU 硬件有 XTHF 停振检测电路，与 XTHF 一同使能或关闭，当停振检测电路检测到 XTHF 停振，会置位停振标志位(CMU_ISR 的 HFETIF 标志位)，如系统主时钟直接或间接(如 PLL)使用 XTHF，则系统主时钟由硬件自动切换到 RCHF，必须由软件恢复。

从可靠上考虑，用户软件必须有检查停振标志位并修复系统主时钟的软件措施，可推荐的方案如下：

➤ **系统主时钟源是 XTHF 经 PLL 倍频：**

当检测到 XTHF 停振时立即切换成 RCHF 经 PLL 倍频方案(目标频率不变)，查询到 XTHF 重新建立稳定状态后，则再切换回 XTHF 经 PLL 倍频方案，具体可参考《FM33FG0xxA 外部高频晶体停振系统主时钟处理方案》应用笔记。

➤ **系统主时钟源是 XTHF：**

当检测 XTHF 停振时立即切换成 RCHF 方案(目标频率不变)，查询到 XTHF 重新建立稳定状态后，则再切换回 XTHF 方案，具体可参考《FM33FG0xxA 外部高频晶体停振系统主时钟处理方案》应用笔记。

8.4 PLL

U01 版芯片如使能了 PLL 电路，则必须禁止使用 ECC_C_RAMP_FDET_IRQHandler 中断服务。



8.5 LSCLK 配置

LSCLK 主要作为 SVD、RTC 以及 Sleep 模式下外部引脚中断采样的时钟源。

MCU 启动后 SystemInit() 函数会禁止 LSCLK 自动切换功能(CMU_SYSCLKCR:LSCATS), 并选择 RCLP 作为 LSCLK 的时钟源 (CMU_LSCLKSEL: LSCATS)。

如用户有使用外部 XTALF, 建议选择 XTALF 作为 LSCLK 的时钟源, 并使能 LSCLK 自动切换功能, 相关软件实施流程可参考复旦微 FM33FG0xxA 标准例程《LSCLK_SelXTALF 示例》。

使能自动切换功能后, 如 XTALF 停振硬件将自动选择 RCLP 作为时钟源, 如 XTALF 恢复振荡则将自动重新选择 XTALF 作为时钟源, 提高了时钟安全性。

9 电源管理单元 (PMU)

9.1 Deep Sleep 配置

如用户使用到深度休眠, 不要通过 CPU 系统控制寄存器 (SCB) 的 SLEEPDEEP 位实现, 而应通过低功耗控制器 (PMU_CR) 的 SLPDP 位实现。

MCU 休眠配置流程如下:

- 配置 PMU_CR 寄存器的 WKFSSEL 位: 选择 Sleep/DeepSleep 唤醒后的系统频率;
- 配置 PMU_CR 寄存器的 DeepSleep 位: 选择是 Normal Sleep 或 Deep Sleep;
- 配置 PMU_CR 寄存器的 PMOD 位: 选择 Sleep mode/DeepSleep mode;
- 执行 __WFI() 或 __WFE() 指令, 等待中断或事件唤醒。

不建议用户使用 Deep Sleep 模式。

9.2 模拟 BUFFER 控制寄存器(PMU_BUFCCR)

如用户软件使用到 ADC 模块且选择 VDDA 为参考源时, 为保证采样精度一般会使能 AVREF 输出缓冲 (AVREFBUF_EN), 同样如应用软件使用到 ADC 模块的温度采样时, 会使能 Vpata Buffer (VPTATBUFFER_EN)。使能此两个 BUFFER 输出会有 7uA 左右的功耗, 因此如用户软件有 MCU 休眠功能, 在休眠之前应禁止此两个 BUFFER 输出。



模拟 BUFFER 控制寄存器 (PMU_BUFCCR)

位号	助记符	功能描述
31:6	-	RFU: 未实现, 读为 0
5	AVREFBUF_OUTEN	AVREF 输出缓冲输出使能, ADC 采样 AVREF 时建议使能
4	AVREFBUF_EN	AVREF 输出缓冲使能, ADC 采样 AVREF 时建议使能
3	VPTATBUFFER_OUTPUTEN	Vptat Buffer 模块开关通道输出使能信号, 高电平使能有效。(PTAT Buffer Output Enable)
2	VPTATBUFFER_EN	Vptat Buffer 模块使能信号, 高电平使能有效。(PTAT Buffer Enable)
1	-	RFU: 未实现, 读为 0
0	-	RFU: 未实现, 读为 0

10 I/O 端口 (GPIO)

10.1 FM33FG0x5A (48PIN) 用户代码要求

FM33FG0x5A 型号针对的是 48PIN 封装芯片。

48PIN 封装芯片的第 43 引脚同时连接着 PD0 和 PE7 端口, 驱动库的 fm33fg0xxa_fl_gpio.c 文件针对该引脚有防配置冲突的处理机制, 需要用户在程序中宏定义 **FM33FG0x5A** 型号以使能该处理机制。

即客户使用 48PIN 封装芯片时, 必须在用户程序中添加宏定义: **FM33FG0x5A**

10.2 外部引脚中断(EXTI)

- (1) 每个数字引脚均可配置为 EXTI 功能。
- (2) GPIO 需先配置输入功能口, 再配置 EXTI 相关参数。
- (3) 在 Sleep 模式下使用 EXTI 功能 (如中断唤醒), 采样时钟源必须选择 LSCLK (CMU_OPCCR1 寄存器), 同时通过下面寄存器使能此功能:

系统时钟配置寄存器 (CMU_SYSCLKER) 的 Sleep/DeepSleep 模式下 EXTI 采样设置 (SLPENEXTI):

- 1: Sleep/DeepSleep 模式下使能外部引脚中断采样 (采样时钟为 LSCLK)
- 0: Sleep/DeepSleep 模式下禁止外部引脚中断采样 (将无法产生 EXTI 中断)



10.3 GPIO 上拉和下拉

同一个 IO 口严禁同时使能上拉和下拉功能。

10.4 GPIO 驱动强度

建议均保持驱动次强 X3。

10.5 SWD 引脚

SWD 引脚不推荐功能复用，MCU 复位启动后该引脚缺省为 SWD 数字功能\使能上拉（50k）。如一定要使用复用功能，则推荐在 MCU 复位启动后等待 100ms 再配置复用口，以给 SWD 口连接留有窗口期。

11 经典控制器区域网络（FSCAN）

如使用到该外设，用户软件必须注意以下重要事项：

（1）初始化 CAN 模块时，必须使能接收数据预采样同步功能（软件使能 MSR.RX_PRESYNC_EN 位）。

（2）CAN_SR 寄存器中的 ACFBSY 无法置位，用户软件不可以参考该位确认滤波器是否正在工作。

（3）只有 BUSOFF 恢复产生中断，BUSOFF 发生不产生中断，如有需求用户软件可以开启错误状态转换中断，结合 CAN_SR 中的 RSTS/TSTS=0x3 判断 BUSOFF 发生。

（4）16bit 滤波器模式下无法区分扩展数据帧和扩展远程帧，因此建议扩展帧不使用 16bit 滤波器模式，而是使用 32bit 滤波器模式。

（5）TxFIFO 寄存器无法被复位为 0，即配置过一次 TxFIFO 后，该寄存器内容即保留，无法被复位为 0，用户软件注意判别。

（6）读取 RxFIFO 要以 Receive FIFO Not Empty 中断标志为准，当 FIFO 为空此时读取 FIFO 的数据是不正确的。

12 灵活速率控制器区域网络（RS-CANFD-Lite）

如使用到该外设，用户软件必须注意以下重要事项：



(1) 各寄存器的写操作有模式保护，用户软件必须在合法的模式下写寄存器，以避免可能的错误

(2) 模块全局或者通道模式切换时，必须等待对应标志置起后才可继续执行其他操作，如 CH_OPERATION 到 CH_RESET，必须等待 CFDC0STS.CRSTSTS 位置起。

(3) 由于分配给 CANFD 模块的 RAM 是有限的，配置 RXMB+FIFO buffer 参数时，如 FIFO 深度，负载数据大小等，需要注意总的分配空间不可超过 RAM 限制。

13 通用异步收发传输器 (UART)

13.1 硬件 LIN

如使用到该外设，用户软件必须注意以下重要事项：

(1) 帧长度超时检测功能是针对 Break 持续低电平长度超时，建议用户软件不要使用此功能，可由软件定时器实现帧长度超时检测。

(2) 发送冲突后由硬件发送隐性电平的功能未实现，即硬件会继续向总线发送剩余位，因此在产生发送冲突中断后如不希望继续发送剩余位时，则需要软件取消发送。

(3) 未到 WKUP_LEN 长度的低电平即能检测到唤醒信号，且检测中断延迟超过 2 个 RCLP，WKUP_LEN 为 n，表示唤醒长度至少 n-2 个 RCLP 时钟周期，Linwkdf 标志延迟 2~3 个 RCLP。

(4) BERM(冲突检测)使能情况下，WKUP 段检测 LINBER(LIN Break Error)置起位置会有变动，原因是 WKUP 发送逻辑独立于 UART 且使用 RCLP 工作，因此建议软件在 WKUP 阶段不使能 BERM 功能。

(5) Break 长度检测与波特率挂钩，导致【break+55 +break+55+正常帧通讯】无法识别，因此在接收帧过程中如发生 FrameError 且当前接收数据为 0x00，则该帧需软件特殊处理。

(6) 当发送 Break+非 0x55，UARTx_ISR 寄存器中的 LINSYNCF 位(LIN 同步段接收完成)连同 SER 位(LIN sync error)、LINBER 位(LIN Break error)会一起置起。因此软件必须先查 SER 再查 LINSYNCF 标志的有效性。



(7) 当同时置位 UARTx_LINCR 寄存器中的 TX_BF(发送 Break 信号)和 TX_WKUP(发送唤醒信号), 实际会发送持续时间较长的信号。

(8) 在检测到同步段不成功时, UARTx_LINSPR 寄存器的同步段预同步计数值为 0。

(9) 在任意初始波特率下, 无法自适应检测 1K~20K 全范围波特率, 因此需根据应用确定要检测的波特率区间, 选定合适的初始波特率。

(10) 开启自测试模式后, IO 也会同时输出到总线。

14 模数转换器 (ADC)

如使用到该外设, 用户软件必须注意以下重要事项:

(1) ADC 校准需要使能通道后再校准, 否则 CALEN 标志位无法正常清除, 需要校准前打开通道, 校准后关闭, 例程驱动已配置。

(2) ADC 触发信号不能小于 1 个 ADCCLK, 否则导致之后触发信号无法置位 SOC。

(3) 使能 ADC 采样前, 需做一次 EOC 标志清除动作, 参考例程。

(4) 禁止 ADC 输入模拟信号高于 VDDA。

(5) ADC DMA 采样仅支持正向/反向扫描, 软件需处理匹配关系。

(6) ADC 在校准时不能设置过采样。

(7) ADC 使用的工作时钟是异步时钟(即不是 APBCLK), ADC 在配置为自动模式 (SEMI=0)、单次转换 (CONT=0), 等待模式 (wait=1) 时, 有一定概率会在转换序列完成后多一次冗余 SOC, 导致 EOC 的冗余, 影响软件例程的逻辑以及后续转换值的读取, 规避措施如下:

①使用例程流程可规避。以查询例程为例, 主要做法是在采样一个通道(或一个序列)结束后立即关闭 ADC(阻止多余的 EOC 产生), 在下次采样再使能 ADC 前清除一下 ADC 的完成标志(即使有多余的 EOC, 可清除掉消除影响)。这样可以消除多余的 EOC 带来的影响。

②wait=0 也可规避这个问题, 但是需要注意在下次转换前取走 ADC 数据。



(8) 使用 ADC 的 DMA 循环模式时需要注意, 同样可能出现上述第(5)/(7)点出现的问题, 解决办法是将 wait=0, DMA 的非循环模式没有这个问题。

15 数模转换器 (DAC)

如使用到该外设, 用户软件必须注意以下重要事项:

(1) DAC 外设复位和 EN 后, DAC 使能后输出的是上一次 DAC 输出的电平。DAC 有两个数据寄存器, 软件可操作的是 DHR, 数据写入 DHR, 写入动作会将 DHR 的数据搬入另一个寄存器 DAC_DOR (程序不可见), DAC 才输出对应的电平, DAC 输出的是 DOR 中的电平。程序复位 DAC 模块或重新使能 DAC 模块和写寄存器的动作不同, 初始化的是 DHR 寄存器, DOR 还是保存之前的数据, 因此在使能 DAC 之前, 把需要输出的数据先写入数据保存寄存器 DHR。

(2) DMA 触发 DAC 只是把 RAM 中数据搬入 DHR, 需要第二个数据搬入 DHR, 才把第一个数据搬入影子寄存器从 DAC 搬出。原理如上一条描述, 另外 DMA 模式的 DAC 是触发模式, 需要触发采样把 DHR 数据搬入 DOR。

(3) DAC 无驱动能力, 需外接驱动 buffer。

16 时钟监控 (CLM)

MCU 在休眠唤醒后软件重新使能 CLM 时, CPU/SWD 单写 CLM_EN 无法开启 CLM, 需要先写 0 再写 1。

17 直接存储访问控制器 (DMA)

如使用到该外设, 用户软件必须注意以下重要事项:

(1) DMA 的传输完成标志只是代表外设和 RAM 之前数据搬移完成, 不代表外设完成相应动作。如 UART 发送的 DMA 应用, DMA 通道传输完成标志置起, 但 UART 的发送还没有完成。以 UART 举例, 建议在 DMA 传输完成标志置起后再查询 UART 的发送完成标志 TXSE。

(2) size 不会自动更新, memory 地址会自动更新, 因此查询发送的进度可以看



memory 地址。

(3) DMA 的通道请求映射为自上而下，与编号无关。

(4) 用户软件必须确保避免使用 DMA 访问非法地址。

18 定时器模块

如使用到相关定时器外设，用户软件必须注意以下重要事项：

(1) 使用定时器输出比较功能进行 PWM 输出时需注意：

①ATIM、GPTIM0/1/2、LPTIM16、LPTIM32 定时器模块各自共用同一重载定时器，所以各自通道输出 PWM 频率固定，TAU 模块各通道可单独设置不同输出频率。

②ATIM、GPTIM0/1/2、TAU 的比较值 CCR 在配置时无需减 1，LPTIM16、LPTIM32 的比较值在 CCR 配置时需做减 1 处理。

(2) 进行输入捕获资源规划时需注意：TAU0、TAU1、LPTIM32 通道支持双沿捕获，ATIM、GPTIM0/1/2、LPTIM16 通道仅支持单沿捕获。

(3) TAU 模块中，只有 TAU0 支持级联计数模式，TAU1 不支持级联计数。

19 SPI 模块

关于 SPI 判断发送完成，SPIx_ISR 寄存器中的 TX Buffer Empty 标志位不等于移位寄存器发送完成；同时在写完 SPIx_TXBUF 发送缓存寄存后，SPIx_ISR 寄存器中 BUSY 有个同步过程，即从 0 到 1 有个延时，当 SPI 的工作时钟和系统时钟相差特别大时，这个问题会特别明显。因此关于判断发送完成建议用户软件采用以下方式：

(1) **全双工模式**：SPI 原理是主从移位寄存器交换，可以判断 SPIx_ISR 寄存器中 RX Buffer Full 标志位来判断接收完成。

(2) **半双工模式**：发送完成不能判断 SPIx_ISR 寄存器中 RX Buffer Full 标志位，但可以使用 BUSY，先判断 BUSY 为 1，再判断 BUSY 为 0，发送完成。

FL_WriteTXBuff(SPI1,data); //写 TXBUF



`while (SPI1->ISR &(1U<<8)) ==(1U<<8));`//等待 BUSY 置 1，使用寄存器操作可以确保主时钟在 64MHZ 也可以保证程序逻辑没有问题

`while(FL_SPI_IsActiveFlag_Busy(SPI1));`//BUSY 变 0，发送完成

20 调试支持

(1) 需在 NMI_Handler 服务函数中增加清除 CPU Debugger 唤醒标志位，具体执行可参考复旦微 FM33FG0xxA 标准例程。

(2) MCU 在 Sleep 模式下如用户要仿真调试，需用户软件配置 MCU DEBUG 配置寄存器的休眠模式下的 debug 配置位 (DBG_SLEEP)：

0：正常进入休眠，关闭 HCLK 和 FCLK，debugger 无法保持连接

1：不关闭 HCLK 和 FCLK，debugger 能够在休眠模式下保持连接

注：正式发布程序建议配置为 0。

21 地址对齐访问

FM33FG0xxA 使用的 CPU 内核为 Cortex-M0+，因此用户软件对 RAM、FLSAH 和寄存器进行数据访问时，必须遵循 ARM 要求的地址对齐访问规则，如地址非对齐访问软件将进入 HardFault_Handler 硬件错误服务。

22 HardFault_Handler

用户软件应定义 `void HardFault_Handler(void)` 硬件错误服务函数，并在服务函数内部增加软件复位命令，以提高系统的可靠性。

23 NMI_Handler

FM33FG0xxA 的 RAM、Code Flash 和 Data Flash 都支持 ECC，可实现纠 1 检 2 (SEC-DED)，上电后 ECC 默认开启，软件无法关闭。

当 ECC 检测到 2bit 及以上错误时会触发 NMI_Handler 中断服务，由于 NMI_Handler 属于不可屏蔽中断，因此用户软件必须定义 `void NMI_Handler(void)` 中断并在服务中



清除 ECC 检错标志，从而避免 ECC 触发 NMI 后程序死机，具体执行可参考复旦微 FM33FG0xxA 标准例程。



版本信息

版本号	发布日期	更改说明
1.0.0	2023.7	首次发布
1.0.1	2023.8	删除 SVD 门控功能的描述
1.0.2	2024.4	增加 10.1 FM33FG0x5A（48PIN）用户代码要求



上海复旦微电子集团股份有限公司销售及服务中心

上海复旦微电子集团股份有限公司

地址：上海市国泰路 127 号 4 号楼

邮编：200433

电话：(86-021) 6565 5050

传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司

地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室

电话：(852) 2116 3288 2116 3338

传真：(852) 2116 0882

北京办事处

地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室

邮编：100007

电话：(86-10) 8418 6608

传真：(86-10) 8418 6211

深圳办事处

地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室

邮编：518028

电话：(86-0755) 8335 0911 8335 1011 8335 2011 8335 0611

传真：(86-0755) 8335 9011

台湾办事处

地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室

电话：(886-2) 7721 1889

传真：(886-2) 7722 3888



新加坡办事处

地址: 237, Alexandra Road, #07-01, The Alexcier, Singapore 159929

电话: (65) 6472 3688

传真: (65) 6472 3669

北美办事处

地址: 2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA

电话: (480) 857-6500 ext 18

公司网址: <http://www.fmsh.com/>