



复旦微电子

# *FM36LV0XXA* *车用系列MCU*

## 软件开发注意事项

---

V1.0.0



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.....	1
2 驱动及例程版本.....	1
3 上下电处理注意事项.....	1
4 独立看门狗(IWDT).....	1
5 电源电压检测(SVD).....	2
6 复位管理单元(RMU).....	2
7 时钟管理单元(CMU).....	2
7.1 时钟源启动稳定时间.....	2
7.2 高频 flash wait cycle 设置.....	3
7.3 RCHF 调校.....	3
8 电源管理单元(PMU).....	4
8.1 休眠唤醒.....	4
8.2 模拟 BUFFER 控制寄存器.....	4
9 总线与存储.....	5
9.1 复位.....	5
9.2 Flash 擦写/编程.....	5
9.3 Flash 读取等待.....	5
9.4 工作时钟源.....	5
10 GPIO 模块.....	6
11 ADC 模块.....	6
11.1 基准源.....	6
11.2 冗余 SOC 注意事项.....	6
12 定时器模块.....	7
13 LIN 模块.....	8
14 经典控制器区域网络(CAN).....	8
15 UART 模块.....	9
16 I2C 模块.....	9
17 DMA 模块.....	9
18 SPI 模块.....	9



19 FL 驱动库使用注意事项 ..... 10

版本信息..... 11

上海复旦微电子集团股份有限公司销售及网点..... 12



## 1 说明

本文档为 FM36LV0xxA 系列车用级 MCU 软件开发注意事项,用户在项目开发前期可参考此文档中所阐述的各项建议。

## 2 驱动及例程版本

需确认驱动及参考例程版本是否为复旦微电子开发者论坛汽车电子模块所更新的最新版本。

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,可以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(IWDT)

(1) 建议上电最先初始化看门狗模块,且是 main 函数第 1 个执行的功能。

(2) 若休眠周期大于看门狗周期,配置唤醒源周期唤醒清狗或使能 IWDT 在休眠模式下冻结计数。

注 1: 需确保生产时编程器配置正确。

(3) 若使用 IWDT 的窗口功能,需开启中断清除开门狗。

(4) IWDT 的读结果会保留在总线上,建议读取 IWDT 寄存器后必须读 IWDT 一个为 0 的寄存器,SEVR 寄存器读一直为 0。

如: `READ_REG(IWDTx->CNT);`读 IWDT 寄存器 CNT,结果留在总线上;

`READ_REG(IWDTx->SERV);`读 IWDT 寄存器 SERV,总线留存的数据清 0;



驱动函数中已做处理，可以直接调用 IWDT 相关函数。

## 5 电源电压检测(SVD)

用户软件在完成 IWDT 初始化后，应立即启用 SVD 以监控 MCU 电源(后端 VDD)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，以避免在低压状态下运行所带来的风险。

MCU 上电正常运行后用户软件应使用 SVD 对 MCU 的电源电压进行掉电检测，当低于目标阈值时，用户软件应有保护性设计，以防止应用程序在低压状态下运行。

详细设计及软件实现请参考《FM33 系列 MCU 上下电处理注意事项》应用笔记和 FM36LV0xxA 标准例程。

## 6 复位管理单元(RMU)

上电过程中用户软件使用 SVD 检测到 MCU 电源电压大于目标阈值后，应及时配置复位管理单元的 BOT 下电复位阈值，推荐至少配置 BOR 至 2.0V 以上，即使进入 MCU 休眠也应禁止关闭。

驱动默认开启 BOR 至 2.0V，需确保 VDD 电压大于 2.0V 后才能实现稳定工作。

## 7 时钟管理单元(CMU)

### 7.1 时钟源启动稳定时间

(1) 若使用 XT HF，需等待 XT HF 起振后再配置为主频，例程考虑不同频率和振荡幅度设置为 2ms，起振时间可根据晶体匹配报告合理设置；初始化停振检测中断时，需先清除停振中断标志位。

(2) 若使用 PLL，使能之前必须确保 PLL 使用的时钟基准源已稳定工作，使能 PLL 后用户软件必须判断 PLL 锁定标志置位后，才能继续执行，需严格按照标准例程。

(3) 在选择系统时钟之前，用户软件必须判读时钟源已稳定工作。



## 7.2 高频 flash wait cycle 设置

(1) 若使用锁相环 PLL 倍频输入时钟作为系统时钟，需等待 PLL 完成建立后再切换切换系统时钟源为 PLL。当  $24\text{M} < \text{系统时钟频率} \leq 48\text{M}$  时，需配置 FLASH 读等待周期为 1wait； $48\text{M} < \text{系统时钟频率} \leq 64\text{M}$  时，需配置 FLASH 读等待周期为 2wait。

(2) 若使用 RCHF，需根据目标频率向 RCHF 调校寄存器写入对应 TRIM 值，保证时钟精度；若是时钟频率  $> 24\text{MHz}$ ，需合理配置 FLASH Wait Cycle。

## 7.3 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;

注 1：使用 RCHF 32MHz 时，需配置 1 个 flash wait cycle。

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



## 8 电源管理单元(PMU)

### 8.1 休眠唤醒

(1) 可根据是否使能全局中断实现唤醒后是否进入中断，若是不进入中断，唤醒后需清除唤醒源 ISR 标志位，再清除 NVIC Pending 标志。

(2) 低功耗模式的唤醒源外设，必须使能 NVIC，否则可能导致外设唤醒而内核休眠的状态。

(3) GPIO 唤醒 MCU，不开滤波时间为 1~2 个 LSCLK，开滤波为 5~6 个 LSCLK，才能检测到下降沿。

### 8.2 模拟 BUFFER 控制寄存器

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用 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(VREF->BUFCR)

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	VREFBUFFER_OUTPUTEN	Vref Buffer 模块开关通道输出使能信号，高电平使能有效。(VREF Buffer Output Enable)
0	VREFBUFFER_EN	Vref Buffer 模块使能信号，高电平使能有效。(VREF Buffer Enable)





## 9 总线与存储

### 9.1 复位

FLASH 擦写过程中，需避免出现复位和低压的情况，否则无法保证 FLASH 的内容。

### 9.2 Flash 擦写/编程

(1) FLASH 擦写过程中，CPU 取指被暂停，需评估 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。

(2) FLASH 擦写结束后，需关闭 FLASH 总线时钟和工作时钟，并锁定 Key。

(3) 若使用 FLASH 模拟 EEPROM，擦写前需对地址进行校验，防止越区操作。需根据实际应用对模拟区域的擦写寿命进行评估。

### 9.3 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源之前必须确保 Flash 读取控制寄存器(FLS\_RDCCR)满足如下要求：

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，提前使能 1wait；

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

禁止目标 CPU 主频大于 64Mhz；

### 9.4 工作时钟源

Flash 擦写/编程工作时钟源硬件固定为 RCHF，支持 8MHz、16MHz 和 24MHz，因此执行擦写/编程功能前，必须使能 RCHF 且配置目标频率应小于等于 24MHz。

如系统时钟同样选择 RCHF 且使用到 Flash 擦写/编程功能，则需要注意 RCHF 频率应小于等于 24MHz。



## 10 GPIO模块

- (1) 对于 **double bonding** 的引脚复用，同 1 个 PAD 只能选择单一引脚使用。
- (2) 若使用 GPIO 外部中断，由于 4 个 GPIO 共用一个 EXTI 中断入口，需避免外部中断引脚位于同一 EXTI 分组内。例如避免同时使用 PA0 与 PA1 作为外部中断引脚，否则无法确定触发引脚号。

GPIO	EXTI 输入选择	EXTI
PA0~PA3	EXTI_ASEL[1:0]	EXTI[0]
PA4~PA7	EXTI_ASEL[3:2]	EXTI[1]

- (3) 禁止任何引脚的输入电压高于 MCU 电源电压。
- (4) 浮空引脚建议配置成高阻抗模式，可调用驱动库函数，默认为高阻抗模式。
- (4) 不建议 SWD 口复用，如必须复用，则 MCU 启动后延时 100ms 再配置，确保复位后外部设备能及时与 SWD 口创建连接。

## 11 ADC模块

### 11.1 基准源

推荐 ADC 基准源选择 VDDA，内部 VREFP 基准源引脚连接至 VDDA。禁止 ADC 输入模拟信号高于 VDDA。

### 11.2 冗余 SOC 注意事项

- (1) 使能 ADC 采样前，需做一次 EOC 标志清除操作，例程已作处理。
- (2) ADC 校准完成后会多产生一个 EOC 信号，因此建议等待校准完成后，关闭 ADC，消除 EOC 信号，关闭 ADC 校准值仍然保存。

```
while (FL_ADC_IsActiveFlag_EndOfCalibration(ADCx) == 0); //等待校准完成
FL_ADC_ClearFlag_EndOfCalibration(ADC);
```

FL\_ADC\_Disable(ADCx);

采样时再清下标志

FL\_ADC\_ClearFlag\_EndOfConversion(ADC);

(3) ADC 在进行校准时不能设置过采样。

(4) 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的转换，导致 EOCIF 标志的置位，影响软件流程的逻辑以及后续转换值的读取。

①使用我们例程的流程可以规避。以查询例程为例主要的做法是在采样一个通道(或一个序列)结束后立刻关闭 ADC(阻止多余的 EOC 产生)，在下次采样在使能 ADC 前清除一下 ADC 的完成标志(即使有多余的 EOC，清除掉消除影响)。这样可以消除多余的 EOC 带来的影响。

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

(5) 使用 ADC 的 DMA 循环模式时需要注意，同样可能会有第 5 点中提到的问题，解决方法是将 wait=0。DMA 的非循环模式没有这个问题

## 12 定时器模块

(1) SysTick 在例程中用作软件延时，为弱定义。若应用层需要用到 SysTick，则需重新定义延时函数。

(2) 定时器 DMA 在搬移 CCR 数据时，搬运 0x00 数据时，引起 DMA 数组中 0x00 后第二个数据会丢失。

解决办法：

① 可通过定时器数字功能切 GPIO 功能，拉低延时实现 PWM 低电平

② 或者配置定时器输出比较模式，将波形固定为低

(3) 若使用 ATIM 从机触发模式捕获占空比和频率，需对捕获值加 1。

(4) LPTIM 的 ARR 值不能写 0、1。

(5) 若使用定时器输出比较模式输出 PWM，需注意：

① ATIM 和 GPTIM 的比较值 CCR 在设置目标值时不需要减 1，LPTIM 需要减 1。

② ATIM 的互补通道输出频率和占空比和正向通道保持一致。

③ 由于同一定时器共用一个自动重装载值寄存器，所以同定时器下不同通道输出频率相同，占空比不同的 PWM。

## 13 LIN 模块

(1) 软件模拟 LIN 方案中，LIN 所用 UART 中断优先级为最高，不可被打断。

(2) LIN 协议栈要求主频不低于 24M，目前例程为 24M，其他频率移植过程中需调整。仅支持 9600 和 19200 波特率。

(3) LIN Stack 默认为首帧自适应，并非每帧自适应。

(4) 建议严格按照标准例程编写对应代码。

## 14 经典控制器区域网络(CAN)

(1) CAN 外设时钟源精度是否满足 ISO11898-1 要求。

(2) CAN 取消 FIFO 发送，可复位 CAN 模块实现。

(3) CAN 接收时，需要判断 RXNEMPIF 标志位，不能使用 RXOKIF 判断，易导致接收数据错位问题。

(4) CAN 默认开启 busoff 自恢复功能，若是需要实现 busoff 快慢恢复，可配置 CAN 为 config 模式关闭自恢复，具体可参考 CAN BUSOFF 快慢恢复例程。

(5) CAN 唤醒可数字功能切 GPIO 实现，需评估主机厂唤醒丢帧要求。

(6) 访问 CAN 寄存器需要打开 CAN 总线时钟，关闭 CAN 外设不建议关闭总线时钟。

(7) CAN 配置及应用，需严格按照标准例程及应用笔记执行。



## 15 UART模块

- (1) 推荐使用 UART0/1,资源较其他 UART 口更为丰富。
- (2) 修改 UART 波特率前,需将 RX 和 TX 使能同时关闭,修改完成后再进行配置;否则需等待当前计数结束后波特率才完成修改。
- (3) UART 超时仅在接收到数据后才起作用,总线空闲不会触发。
- (4) UART 支持软件模拟 LIN, LPUART 不支持。

## 16 I2C模块

使用 I2C 器件时,需要确认器件是否支持 ACK 信号的 SCL 和 SDA 之间无延时。

## 17 DMA模块

(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。拿 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。

## 18 SPI模块

SPI 判断发送完成, TXBUF 标志不等于移位寄存器发送完成, BUSY 标志置位有个同步过程,有 0 到 1 的一个过程,写完 TXBUFF, BUSY 没有立刻置 1 还是 0,在 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。因此建议在开发过程中采用以下方式:

(1) 全双工模式：SPI 原理是主从移位寄存器交换，可以判断 RXBUF 标志来判断接收完成。

(2) 半双工模式：发送完成不能判断 RXBUF，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
```

## 19 FL 驱动库使用注意事项

(1) 延时。在 FM36LV0xxA FL 库中，使用 SysTick 内核定时器作延时函数，实现毫秒延时(FL\_DelayMs)、微秒延时(FL\_DelayUs)等函数，这些函数均使用 \_\_WEAK 弱定义，方便用户进行修改，FL 库使用 FL\_Init() 函数对如上功能进行初始化，在 main 函数中需要初始化。

FL 库使用了 CMSIS 标准的全局系统时钟频率值变量 SystemCoreClock 进行对 SysTick 延时的计算(位于 system\_fm33lc0xx.c/.h)。为了确保延时初始化正确，请在调用 FL\_Init 函数初始化前，按如下流程进行配置：

1) 按照正常流程配置芯片时钟树；

2) 如果主时钟源自内部来源(例如 RCHF)，那么仅需要调用 SystemCoreClockUpdate() 函数更新 SystemCoreClock 变量；如果主时钟源自外部(XTHF)，那么需要配置全局变量 XTHFClock 为当前使用的外部晶体频率值(单位 Hz，默认值为 8M)，再调用 SystemCoreClockUpdate() 函数更新 SystemCoreClock 变量；

(2) 驱动 system\_fm36lv0xxa.c 中包含 SystemInit() 函数，该函数为 mian 函数启动前执行的函数，默认会装载 RCHF 8MHz 的 Trim 值，若是 boot 和 app 时钟频率存在变化，需注意应用配置。



版本信息

版本号	发布日期	更改说明
1.0,0	2024.5	首次发布





## 上海复旦微电子集团股份有限公司销售及服 务网 点

### 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/>