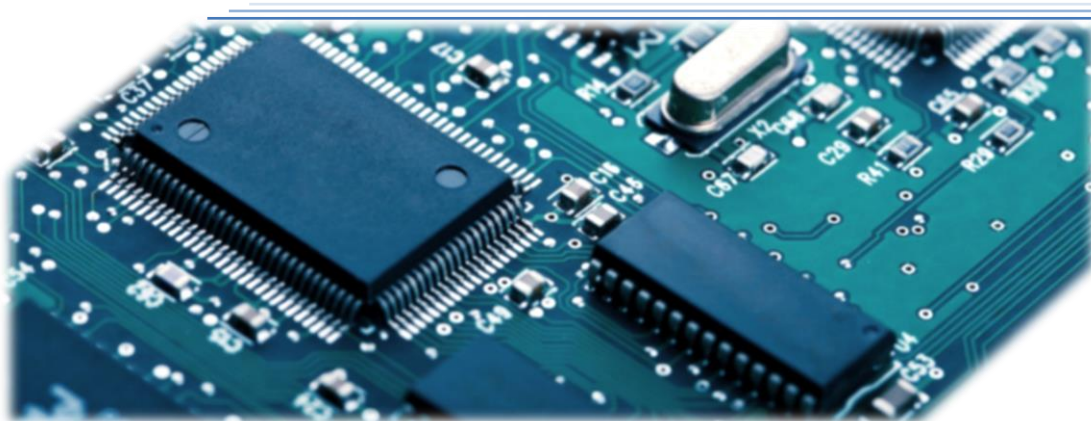




FM33G0XX 开发注意事项



上海复旦微电子集团股份有限公司

Shanghai Fudan Microelectronics Group Company Limited

开发者论坛: <http://www.fmdevelopers.com.cn>



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，不保证其中不含任何瑕疵，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、不承担用于包括军事、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。

前言:

FM33G0XX 是一款 M0 内核的 ARM 芯片, 在客户开发中通常会遇到一些普遍的问题, 本文将分硬件、编译环境、软件和驱动排查简单阐述这些问题, 加快客户的开发过程。详细细节可以参考相关手册、例程和 AN。

1. 硬件

电源

FM33G0XX 系列芯片有电源 VDD 和 VDD15。VDD 为 MCU 的电源, VDD15 为 VDD 通过芯片内部 LDO 输出的 1.5V 电压用于数字电路的电源。

在硬件设计时 VDD 和 VDD15 不能直接连接, VDD15 只需要外接 0.1uF 电容就可以, 详细见推荐原理图。

编程接口

FM33G0XX 系列芯片的编程接口使用的是 SWD 接口, 芯片在处于低功耗状态时编程口不能正常工作, 所以在通常的 SWD 接口的基础上增加了 RST 引脚。通过 RST 引脚去复位芯片, 才能进行仿真器的操作。

假如客户需要使用 RTC, 并对时间的精度要求比较高, 还需要增加 FOUT 接口, 使用 FOUT 接口可以进行外接晶体的温度调校。

外接低速晶体

XTLF 外接 32768K 晶体, FM33G0XX 在没有特殊配置下, 外部晶体停振会自动切换到内部 32K 晶体 RCLP。假如不接外部晶体, 为了防止外部干扰需要将 XTLEFIN 接地。

在对 RTC 时间要求比较高时, 需要使用一致性较好, 精度比较高的外部晶体。匹配电容需要按照晶体手册进行设计。

复位电路

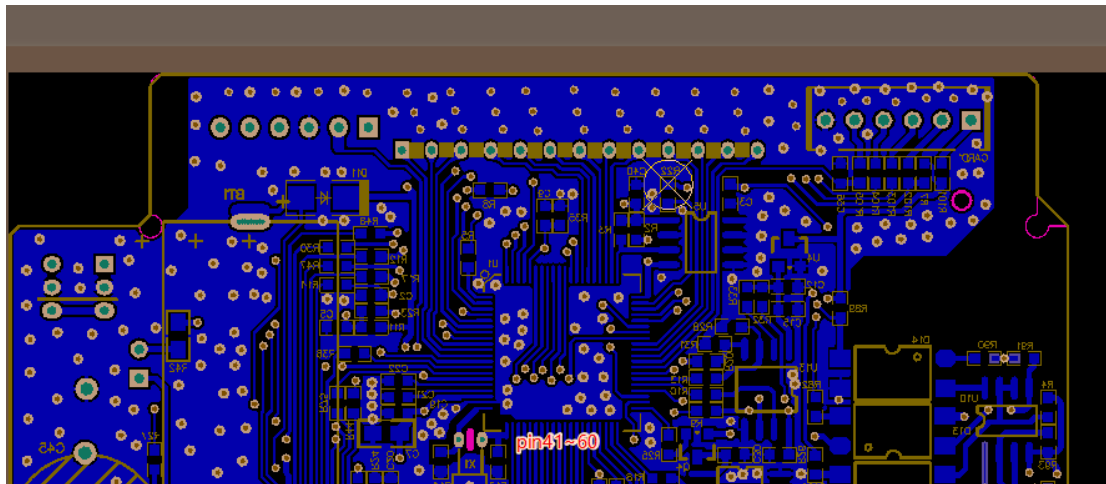
FM33G0XX 系列芯片, 内部集成了上下电复位电路, 不需要在 RST 引脚外部接复位电路, 但可以根据实际情况加滤波电容消除干扰信号。

GPIO

I2C 的 GPIO 引脚是真开漏引脚, 假如需要输出高电平必须外接上拉电阻。其余 GPIO 均可设置推挽或者开漏输出。

PCB 布板

以 80pin 为例，为了保证更好的静电特性 pin41~pin60 这一侧在布板时注意放置在 PCB 内侧。



其他封装的如下

pin64: Pin33-48

pin48: Pin25-36

pin32: Pin17-24

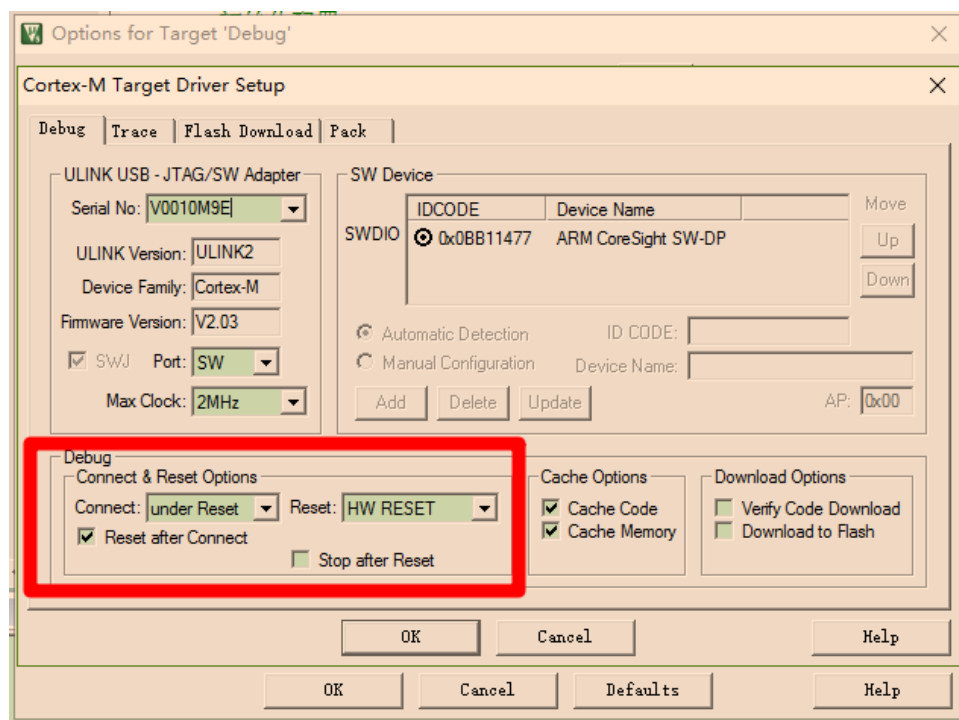
pin16: Pin1、Pin15-16

2. 编译器环境配置

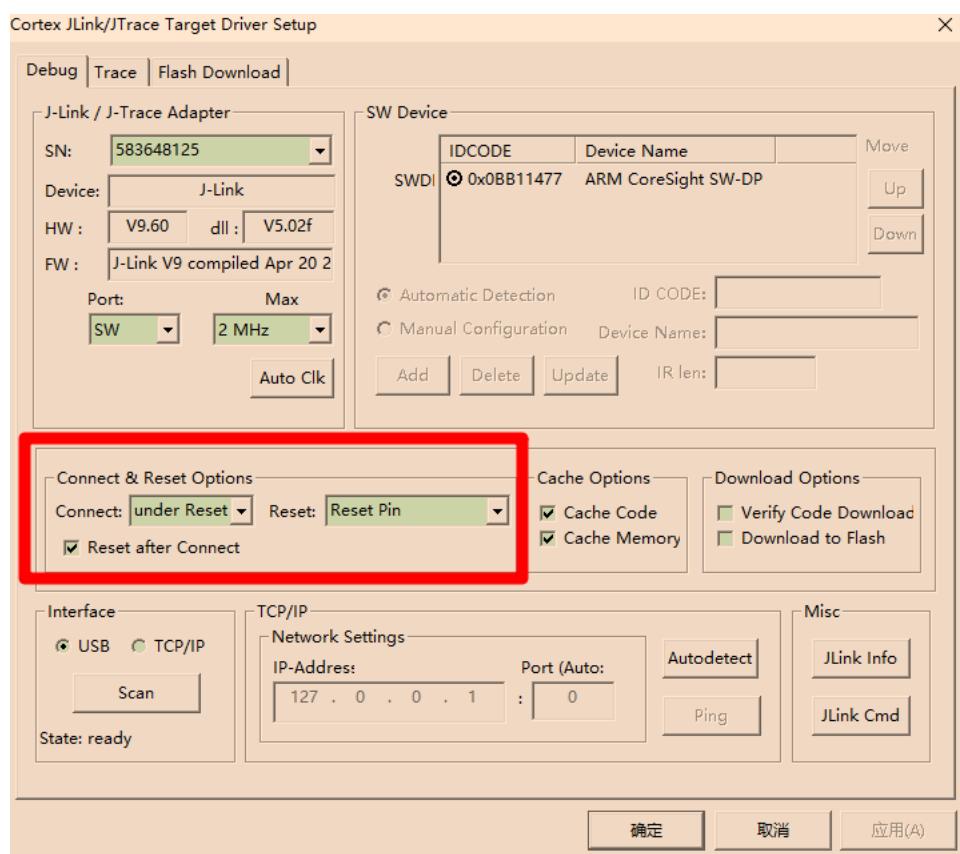
Keil 和 IAR 两种编译器 FM33G0XX 都可以支持。为了在休眠下能够顺利连接仿真器，除了在硬件上编程口需要连接 RST 脚。在编译器的配置上需要如下设置。

Keil

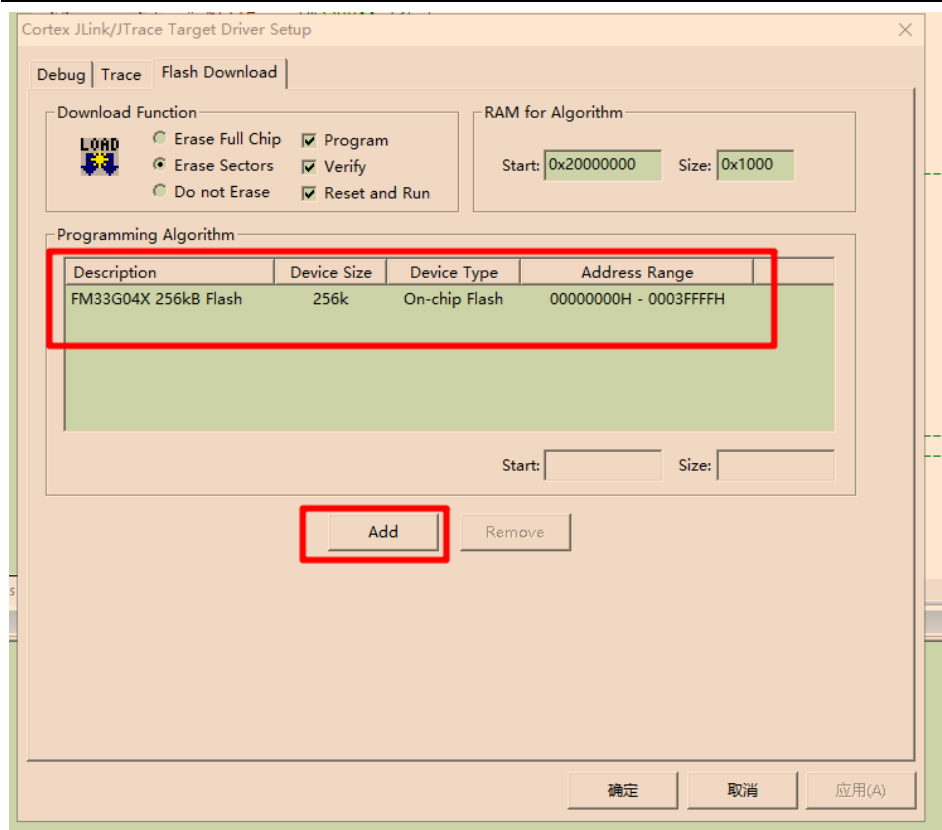
Keil 在使用 ULINK 时 debug 选项中 Connect 和 Reset, 建议按下图配置



Keil 在使用 JLINK 时 debug 选项中 Connect 和 Reset, 建议按下图配置

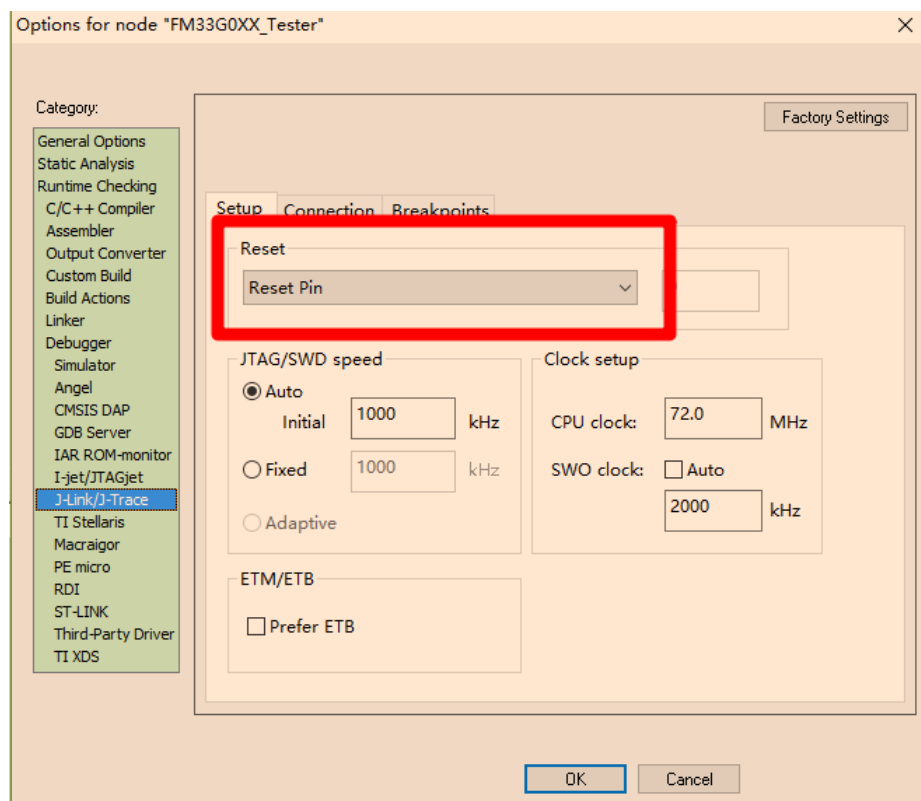


在 Flash Download 中最好确认下 Flash 的配置信息和芯片选型是否一致, 假如不一致需要点击 Add 手动添加。

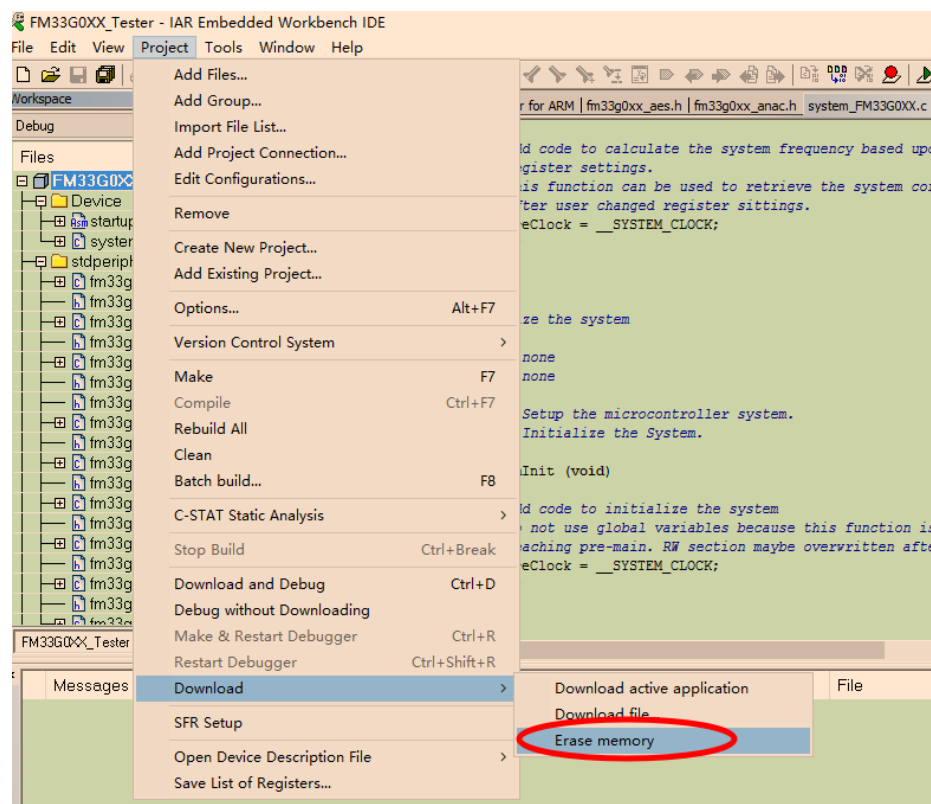


IAR

IAR Reset 选择 Reset pin 如下图



在实际使用中，假如这种配置不能成功烧写，需要先按照下图先擦写再编程



3. 软件

系统主时钟

FM33G0XX 系列可以使用 RCHF 或者 PLL 作为系统主时钟。PLL 做系统主时钟虽然精度高，但是 EMC 性能弱于 RCHF 且 PLL 功耗较大。一般我们建议系统主时钟使用 RCHF。为了确保 RCHF 在全温区的精度满足要求可以使用外部晶体校准 RCHF。（可以参考例程和相应的 AN）。

RCHF 8MHZ 的 TRIM 为上电自动加载，其余频率需要程序手动加载。

MCU 休眠唤醒后默认的主时钟为 8MHZ，可以通过 RCC -> SYSCLKSEL 中的 WKUPCLK 配置为 512K。

仿真时定时器与看门狗

在仿真状态下定时器和看门狗默认为关闭状态，可以操作 MCUDBGCR 来改变状态。

复位

FM33G0XX 有上电复位和下电复位电路，在程序中下电复位必须要打开。

I2C

FM33G0XX 的 I2C 只支持主机模式

I2C 在发送错误后，需要使用专门的函数来复位 I2C 的状态机。具体如下（例程 G3.02 后的版本中有描述）

```
/******  
I2C模块复位函数  
功能：复位I2C，恢复到IDEL状态  
*****/  
void I2C_ResetI2C(void)  
{  
    volatile uint8_t result;  
  
    I2C->I2CIR =0x01;  
    I2C->I2CCTRL =0x6000;  
    result=(I2C->I2CSTA);  
    I2C->I2CSTA = 0x00;  
    result = (I2C->I2CBUF); //清接收完成标志  
    I2C->I2CERR =0x07;  
    for(result=0x00; result<0x03; result++ );  
}
```

UART

FM33G0XX 有 6 路 UART，每一路 UART 都具有红外功能。

UART 的中断标志位有 buffer 空产生和移位寄存器空产生两种，建议使用移位寄存器空。

UART 有 7bit、8bit、9bit 三种，其中 7bit 和 9bit 无奇偶校验。

LPUART

FM33G0XX 的 LPUART 最高波特率为 9600。在接收中断事件唤醒芯片时，LPUART 不判断校验位和停止位。这种情况下假如选择 1byte 数据接收完成唤醒时，不是非常可靠。建议选择接收数据匹配唤醒 MCU。

在发送数据时一定要发送缓冲器空再写入新的数据。

LPUART 有 7bit 和 8bit 两种，具体的应用可参考相应的 AN。

HSPI 和 SPI

FM33G0XX 的 HSPI 和 SPI 支持主从模式，在主机模式下，HSPI 和 SPI 的 SSN 可以由硬件 SSN 来控制，也可以使用任意 GPIO 来模拟。

假如选择 GPIO 来模拟，寄存器 SIPCR2 -> SSNSEN 配置为软件控制，SPICR2 -> SSN 配置为 SSN 输出低电平。GPIO 配置为输出模式，输出 0 或 1 控制从机。

假如选择硬件 SSN 控制，将 SSN 相对应的管脚配置为数字功能，寄存器 SIPCR2 -> SSNSEN 配置为软件控制，SPICR2 -> SSN 配置成 0 或者 1，SSN 管脚就可以输出 0 或者 1。

7816

FM33G0XX 有两路相互独立的 7816 模块，其中 78160 稍有特殊，在 78160 由发转收时，必须等待 1 个 ETU。而 78161 没有这个限制。

DMA

DMA 的数据 buffer 尽量使用全局变量，防止 buffer 释放掉后 DMA 还在接收数据，将堆栈冲破。

DMA ch4 的数据 buffer 必须半字对齐，可以参考相应例程。

DMA SPI 必须先打开发送再打开接收，可以参考相应例程。

RTC

如果应用代码中没有对时钟调校寄存器操作，在每次上电时 RTC 计时精度将因为时钟调校寄存器的随机值发生变化。在需要 RTC 时钟精度的应用中注意时钟调校寄存器的复位值问题，或者写 0，或者按温度进行相关的数据填充。

GPIO

FM33G0XX 在上电过程中，PF03, PF04, PF05, PF06, PF11 为特殊功能测试管脚，在芯片复位电路放开前，一直是高电平，复位电路放开后是高阻态。其余 GPIO 在 LD015 在达到阈值电压之前为不定态，LD015 在达到阈值电压之后为高阻态。

芯片上电至 1.8V 以前，GPIO 的状态不确定，可能是输入、输出或三态，请注意 GPIO 与外部器件的连接关系和逻辑关系，尤其是在应用系统中将 GPIO 用作输出功能时需要特别注意，在电源电压上升至 1.8V 前可能会发生误动作

PF3 的复用数字功能 LPUART_RX 和 UART0_RX 在手册中没有寄存器描述管脚复用哪个功能，在使用中只需将管脚复用为数字功能，使用哪个模块开哪个模块的使能即可。

PC15 的模拟功能 ACMP2 _INP0、ADC_IN6 在物理上是一根线不必区分。

休眠时建议不使用的 GPIO 配置为开漏输出高，SWD 的两个管脚配置为数字功能并且使能上拉电阻。

FM33G0XX 有两个强驱动管脚，PE2 可以最高有 15mA 的灌电流，PG6 可以最高有 15mA 的拉电流。其余 GPIO 最高有 5mA 的拉灌电流。

FM33G0XX 的 GPIO 都有施密特特性，对高低电平的判断见图。

V_{TL}	施密特输入低电平	$V_{DD}=5V$	2.13	2.33	2.53	V
		$V_{DD}=3V$	1.22	1.37	1.52	
V_{TH}	施密特输入高电平	$V_{DD}=5V$	2.3	2.6	2.9	V
		$V_{DD}=3V$	1.49	1.64	1.79	

Flash

- 1 FM33G0XX 没有 Block Erase 功能。注意不要使用
- 2 在程序中进行 FLASH 单次编程时时注意 DMA 的全局使能需要关闭

4. 驱动排查

CRCFlash 校验

适用范围:客户自己编写的驱动

问题描述: 使用 CRC 外设的 Flash 校验功能时, 使能 CRCFLASH 校验后, 下一条指令即访问外设总线, 将引起代码执行错误。

解决办法: 在 CRC 外设 FLASH 校验功能使能后, 插入一条 __NOP() 指令即可。

排查内容及方法

- 1、如果应用中没有使用 CRC 的 Flash 校验功能, 则不必关心该问题;
- 2、如果应用中使用的是调用库函数使能该功能或通过库函数查询外设状态等, 因函数调用会有 PUSH/POP 操作, 可以不关心该问题。
- 3、如果是通过自编写的直接操作寄存器的代码, 请在 CRCFLSEN 置位后插入 1 个 NOP 指令。

DMA

适用范围: 驱动初版~0.3.02G

问题描述: 调用驱动初始化 DMA 时, 奇数地址缓冲区的地址写入 dma 地址寄存器后会被强制清除最后一个 bit, 也就是地址会减 1; 原因是驱动中先对地址寄存器赋值, 后配置 dma, 此时 dma 默认定时器捕捉通道, 需要配置半字对齐的地址, 如果写入奇数地址, DMA 会自动将写入地址的最低比特清零。

解决方案:

- 1、将 dma 缓冲区全部定义为 4 字节对齐的数组，则任何版本的驱动或操作顺序都不会受此问题影响
- 2、使用更新后的驱动库
- 3、先配置通道再写地址寄存器

排查内容及方法：

1. 查看是否使用了 DMA 功能，如果没有使用则不受影响
2. 如果是用户自己编写的驱动，则根据问题描述对 DMA 初始化顺序进行排查并修改

驱动库 SVD 设置 0.7V 问题

适用范围：驱动初版~0.3.02G

问题描述：调用驱动函数 ANAC_SVDVOL_CFG(uint32_t SetValue) 设置 SVD 的基准电压为 0.7V 时，实际的基准电压是 0.8V

解决方案：修改驱动函数，如下

```
void ANAC_SVDVOL_CFG(uint32_t SetValue)
{
    if(SetValue==ANAC_SVDVOL_VOP7EN_Msk)
    {
        ANAC->SVDVOL=ANAC_SVDVOL_VOP7EN_Msk;
    }
    else if(SetValue==ANAC_SVDVOL_VOP75EN_Msk)
    {
        ANAC->SVDVOL=ANAC_SVDVOL_VOP75EN_Msk;
    }
    else
    {
        ANAC->SVDVOL=ANAC_SVDVOL_VOP8EN_Msk;
    }
}
```

排查内容及方法：FM33G0 的客户是否使用 SVD 0.7V 的基准电压，假如有按上述修改



版本	改版内容
初版	
1.0	1 增加驱动排查章节 2 软件部分增加 FLASH BLOCK 说明 3 软件部分增加 GPIO 复位前的状态说明 4 硬件部分增加了其他封装的布板说明
1.1	FLASH 单次编程时时注意 DMA 的全局使能需要关闭