

文档编号: AN2037

上海东软载波微电子有限公司

应用笔记

ES32F010x/ES32F0131

修订历史

版本	修订日期	修改概要
V1.00	2021-9-29	初版
V1.01	2021-12-17	增加 HRC 时钟配置注意事项
V1.02	2022-1-24	统一整理 IWDT 和 WWDT 注意事项
V1.03	2023-2-28	1.添加 ES32F0131 相关信息 2.增加配置字注意事项 3.增加 HOSC 时钟注意事项
V1.04	2023-6-7	1.删除“位带操作” 2.新增“ES32F0101NF 的 PA7 和 PA9 不可产生电平冲突”的描述
V1.05	2024-1-31	更新 USART 注意事项 1
V1.06	2024-9-23	1.IAP 操作程序小节补充 FLASH 编程注意事项 2.新增 DMA 模块注意事项
V1.07	2025-1-8	1.新增“低功耗模式注意事项”

地 址：中国上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：http://www.essemi.com/

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与

上海东软载波微电子有限公司联系

目 录

内容目录

第 1 章	嵌入式软件开发注意事项	5
1.1	配置字	5
1.2	开发环境	5
1.3	库函数选择	5
1.4	HRC 时钟	5
1.5	HOSC 时钟	7
1.6	寄存器写保护	7
1.6.1	系统写保护	7
1.6.2	IAP 操作保护	7
1.6.3	IWDT 写保护	7
1.6.4	WWDT 写保护	7
1.7	IAP 操作程序	8
1.8	FLASH 读保护	8
1.9	GPIO 模块	8
1.10	IWDT 模块	8
1.11	WWDT 模块	9
1.12	I2C 模块	9
1.13	USART 模块	9
1.14	SPI 模块	10
1.15	ADC 模块	10
1.16	DMA 模块	10
第 2 章	硬件开发注意事项	12
2.1	外部晶振	12
2.2	GPIO PA7/8/9	12

第1章 嵌入式软件开发注意事项

1.1 配置字

芯片出厂时，配置字地址单元可能为空，在进行嵌入式软件开发前和量产烧录时，需要对目标芯片进行正确的配置字编程，避免配置字为空。用户使用 ES-Burner (PC 软)+ES-LinkII (调试器) 对目标芯片进行配置字编程。

1.2 开发环境

推荐用户使用 Keil5、IAR8.11 或者 iDesigner 进行固件开发。由于 Keil4 不支持 PACK 机制，故不推荐用户使用 Keil4。

在使用 Keil5 或 IAR 进行开发前，需要使用 ESBurner+ES-Link II 烧录合适的配置字选项，如：正确的启动地址 (BOOT)、BORVS 配置到工作电压以下、看门狗关闭、FLASH 保护全部禁止。进行【擦除】→【配编】后才可以使用 Keil5 或 IAR 进行程序调试。

1.3 库函数选择

ES32 系列芯片提供 2 种类型库函数 ALD 和 MD：

- ALD：提供较为完善的封装接口，提供更为人性化的 API，可降低用户的开发周期，但时间复杂度和空间复杂度较高；
- MD：提供外设模块初始化接口，以及寄存器级别的“读”、“写”接口，适合对芯片底层较为熟悉的用户。

由于本产品 RAM 空间为 4KB，建议用户使用时间和空间复杂度都较精简的 MD 库函数。

1.4 HRC时钟

注意事项 1：当系统时钟为 48MHz 或 52MHz 时，需旁路系统时钟滤 (CFT_CMD@CMU_CSR)，否则可能会造成系统时钟失效。

注意事项 2：ES32F010x/0131 系列 MCU 可配置内部高速 RC 振荡器 (HRC) 为 48(52)/32/16/2MHz，其中：

- ES32F0100 可配置内部高速 RC 振荡器 (HRC) 为 48/32/16/2MHz，不支持 52MHz。
- ES32F0101 可配置内部高速 RC 振荡器 (HRC) 为 52/32/16/2MHz，不支持 48MHz。

此外，使用 ES32F010x/0131 的 SDK 时，需要注意编译器中的宏定义，以 ES32F0100 为例，在 Keil 和 iar 的宏定义配置可参考图 1-1 及图 1-2。

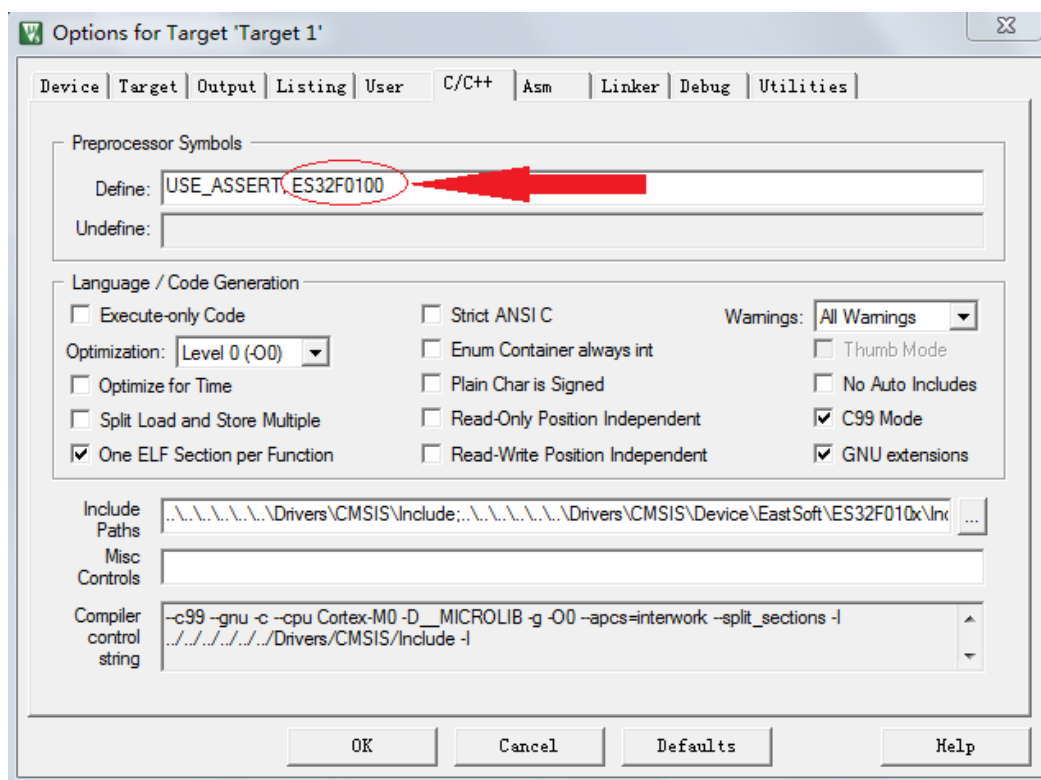


图 1-1 MDK 中宏定义

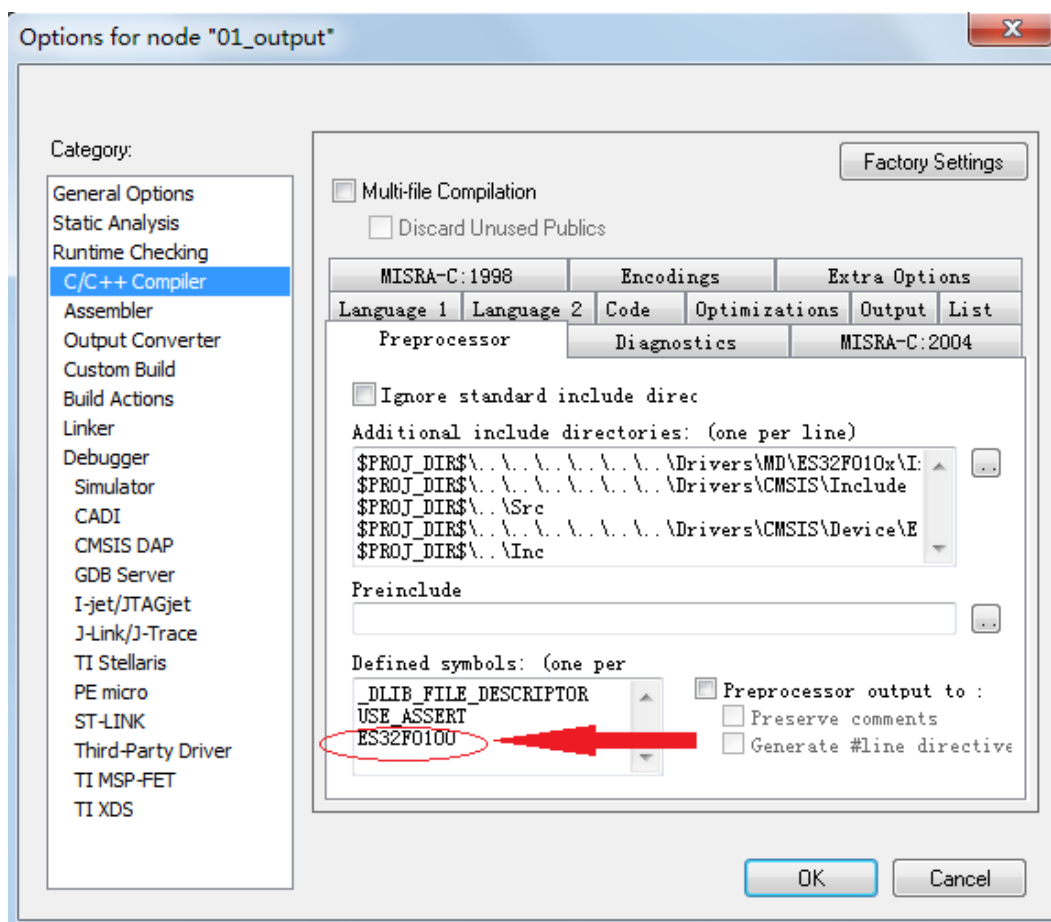


图 1-2 IAR 中宏定义

1.5 HOSC时钟

注意事项 1: 当外部晶振频率高于 12MHz 时, 需设置寄存器 CMU_CLKENR 的 HOSCFLYBPS 位为 1, 旁路外部时钟滤波器, 以免时钟信号被误滤除。

1.6 寄存器写保护

为避免程序的异常导致运行错误, 芯片写保护寄存器用于阻止对被保护的寄存器误操作。

系统控制单元, WDT 等模块支持寄存器写保护, 对被保护的寄存器进行写之前需要解除写保护状态 (允许写), 否则无法对被写保护寄存器写入。操作完成后, 再使能写保护 (禁止写)。库函数中均提供相应宏定义进行解除保护和使能保护。

1.6.1 系统写保护

系统控制寄存器的访问操作会影响整个芯片的运行状态, 芯片提供系统设置保护寄存器 SYSCFG_PROT。对 SYSCFG_PROT 寄存器以字方式写入 0x55AA6996 会解除写保护, 对该寄存器写入其他任何值都会使能写保护。

可通过读 SYSCFG_PROT 寄存器确认写保护状态, 读出值为 0x1, 表示当前处于写保护状态; 读出值为 0x0 表示当前处于写保护解除状态。

SYSCFG_PROT 保护的寄存器为除 SYSCFG_PROT 寄存器外的 SYSCFG、PMU、CMU、RMU 模块所有寄存器。

1.6.2 IAP 操作保护

软件通过写 MSC_FLASHKEY 寄存器, 可解除对程序区的 IAP 操作保护, 处于保护状态时, 无法进行擦除和编程的操作。通过检查 MSC_FLASHKEY.STATUS 是否为 0, 判断 Flash 是否处于保护状态。

1.6.3 IWDt写保护

对 IWDt_LOCK 寄存器以字方式写入 0x1ACCE551 会解除写保护, 写入其他值使能写保护。

可通过读 IWDt_LOCK 寄存器确认 IWDt 模块是否处于写保护状态, 读出值为 0x1, 表示当前处于写保护状态; 读出值为 0x0 表示 IWDt 模块处于写保护解除状态。

该寄存器保护除自身外的 IWDt 所有寄存器。

1.6.4 WWDt写保护

对 WWDt_LOCK 寄存器以字方式写入 0x1ACCE551 会解除写保护, 写入其他值使能写保护。

可通过读 WWDt_LOCK 寄存器确认 WWDt 模块是否处于写保护状态, 读出值为 0x1, 表示当前处于写保护状态; 读出值为 0x0 表示 WWDt 模块处于写保护解除状态。

该寄存器保护除自身外的 WWDt 所有寄存器。

1.7 IAP操作程序

注意事项 1: IAP 操作时会占用 FLASH 总线, 所以 IAP 操作不可在 FLASH 中执行。芯片内置 ROM 固化 IAP 操作函数, 由硬件电路实现。IAP 操作既可放在 SRAM 中执行寄存器操作, 也可以在 FLASH 中调用 ROM 固化的 IAP 操作函数。推荐用户调用内置 ROM 固化的 IAP 操作函数, 以减少 SRAM 中的 IAP 操作代码量。

注意事项 2: 在进行 FLASH 编程时, 无论是否编写相同的数据, 在 FLASH 编程前均必须先进行擦除。禁止通过对一个 word 的多次写入实现按 byte 或按 bit 修改。

1.8 FLASH读保护

当 FLASH 的读保护级别设置为 level1 或 level2 时, 运行在 SRAM 中的程序不能有读 FLASH 的操作。

典型应用 1: 运行在 SRAM 中的程序想读取 FLASH。将读 FLASH 操作放在 FLASH 中执行;

典型应用 2: 程序运行在 SRAM 中, 响应中断请求。将中断向量表拷贝至 SRAM 中, 并设置中断向量偏移地址 (SYSCFG_VTOR 和 SYSCFG_MEMRMP.VTOEN)。

1.9 GPIO模块

注意事项 1: 未使用的 GPIO 管脚建议设置为输出固定电平并悬空, 若设置为输入, 须加上拉或下拉电阻接到电源或地;

注意事项 2: MRST 不能复用为 GPIO (PA2), 若程序中将 PA2 的复用模式 FUNC 修改为非 0, 该 IO 上拉可能失效导致发生复位。禁止用户修改 PA2 的 FUNC 复用模式, 必须保持 FUNC=0 (即 GPIOA_BASE 组的寄存器位 GPIO_FUNC0<11:8>=0)。

注意事项 3: ES32F0101NF 内部将 PA7、PA9 连接在一起, 所以应用程序不可将 PA7 和 PA9 同时输出不同电平。

1.10 IWDG模块

注意事项 1: 通过配置字使能 IWDG 为硬件看门狗 (硬件强制使能), 可使 IWDG 脱离软件配置, 系统则会更可靠。

注意事项 2: 当 IWDG 做为软件看门狗使用时, 配置字 IWDTEN 配置成“由软件使能”, 并通过寄存器对 IWDG 进行初始化。

注意事项 3: IWDG 如果通过配置字使能为硬件看门狗, 则计数时钟固定为 LRC 时钟, 上电默认的溢出周期最小值约 0.6 秒, 用户可通过程序修改 IWDG_LOAD 寄存器来调整计数周期。IWDG 模块固定为使能, IWDG 复位和中断也固定为使能, 软件无法关闭, 寄存器 IWDG_CON 的 CLKS, RSTEN, IE, EN 位均无效。

注意事项 4: 当 IWDG 作为硬件看门狗使用, 或作为软件看门狗使用并将时钟源配置为 LRC 时钟时, 两次喂狗的时间间隔应小于 IWDG 的计数溢出时间。例如: 当 IWDG_LOAD 值为 0x0000_4000 时, IWDG 的最小溢出时间约为 0.32s (LRC 频率最大为 51.2KHz), 则用户程序的喂狗间隔应小于 0.32s。另外, 必须禁止 NVIC 寄存器的 IWDG 中断响应 IRQ。

注意事项 5: 可靠的系统在休眠下也应该保持 IWDG 处于工作状态, 用户可以使用 WWDG 定时唤醒后对 IWDG 进行喂狗, 并且依然要保证 IWDG 的喂狗周期小于溢出周期。

1. 11 WWDG模块

注意事项 1: WWDG 禁止用户在喂狗窗口外喂狗, 建议用户在 WWDG 中断服务程序内喂狗。

1. 12 I2C模块

注意事项 1: I2C 工作在主模式时, 遇到从机不应答的情况会一直拉低时钟线, 此时无法通过发送 stop 或 start 信号来释放时钟线, 可通过关闭 I2C 模块的方式来释放时钟线。

注意事项 2: I2C 作为主机与外部 EEPROM 进行通信, 如果由于异常状态发生导致主机 SDA 线一直被 EEPROM 器件拉低, 发生死锁现象, 此时可以通过重新给 EEPROM 上电解除死锁, 也可以通过主机 SCL 线模拟输出 9 个时钟信号后, 再模拟发送一个停止信号, 使得 EEPROM 终止此次通信并释放 SDA 线, 解除死锁。

1. 13 USART模块

注意事项 1: RX 接收某种异常序列后 USART 接收会出现故障。规避方法: 在 USART 初始化时打开 IDLE 中断使能 (IDLEIE@USART_CON0), 并在对应中断服务程序中清除中断标志 (IDLEIF@USART_STAT)。

注意事项 2: USART 使能后 RX 管脚由高电平变为低电平, 如果 RX 持续为低电平, 则 USART 保持数据接收状态, RXNEIF 则无法被清除, 直到 RX 恢复高电平。USART 使能前, 如果 RX 持续为低电平, 则 USART 使能后不会进行数据接收, RXNEIF 不会被置起, 直到 RX 出现下降沿脉冲信号, 才开始数据接收。

注意事项 3: 帧错误标志 (FERRIF) 有失效的可能, 不建议使用。

注意事项 4: USART 的 IRDA 接收 IRDA 接收数据时, 有几率存在数据接收错误情况, 不建议使用。

注意事项 5: USART 引脚电平反向寄存器位 (TXINV@USART_CON0 和 RXINV@USART_CON0) 不可读, 置位 TXINV/RXINV 两位后, 再以 SET_BIT 方式使能 TX/RX 会导致 TXINV/RXINV 被清零。针对上述现象, 使能电平反向后, 请使用库中特殊处理的 TX/RX 使能函数 md_usart_enable_rx_tx_with_inv。

1. 14 SPI模块

注意事项 1: SPI 在只读模式下 (RXO 位置 1), BUSY 位会一直被置起, 此时用户无法通过该位判断通信是否完成, 只能通过应用程序自行判断通信是否完成 (如使用定时器做超时检测); 并且在只读模式下, 关闭 SPI 模块时 (SPIEN 位清零), BUSY 位也不会清 0, 可在通信完成后, 将 RXO 位清零, 退出只读模式, 则 BUSY 位会清零。

注意事项 2: SPI 从机收发使用软件片选时, 软件控制 NSS (SPI_CON1.SSEN = 1), 此时 SPI 从机固定为被选中的状态, 与 SPI_CON1.SSOUT 的设置无关。

1. 15 ADC模块

注意事项 1: 当正向参考电压选择 VDD 时 (VRPSEL=0), 因 VDD 到 ADC 参考端的路径有约 10~20mV 的压降, 使得 ADC 不能完全满量程 (例如当 VDD=5V 时, ADC 模块的实际参考电压约 4.98~4.99V), 模拟输入信号 V_{AIN} 需小于 VDD-20mV。

注意事项 2: 为了保证 ADC 转换结果的稳定可靠、避免噪声干扰, 强烈建议在模拟输入通道接外部电容 (100nF 或 10nF) 进行滤波。

注意事项 3: ADC 转换时钟速率与参考电压有关, 当参考电压为 5V 时, ADC 时钟最大可配置为 12MHz; 当参考电压为 3.3V 时, ADC 时钟最大可配置为 6MHz; 当参考电压为 2.5V 时, ADC 时钟最大可配置为 375KHz。

注意事项 4: 使用 ADC 转换通道 19 (VREF1V) 时, ADC 时钟速率需小于 16KHz, 采样周期需大于 1ms, 否则会影响 ADC 转换精度, 推荐将 ADC 时钟设置为 128 分频 (寄存器 ADC_CCR 的 CKDIV=111), 采样时间设置为 15 个周期 (寄存器 ADC_SMPTx 的 CHTy=0011)。则当 PCLK 频率为 2MHz 时, 可满足上述 ADC 时钟速率和采样周期的要求; 当 PCLK 频率大于 2MHz 时, 可在 ADC 转换过程中, 通过反复开关 ADC 模块外设时钟的方法来降低 ADC 时钟速率, 并满足采样周期的要求: 例如先设置 CMU_APBENR.ADCEN=0, 延时约 12us, 再设置 CMU_APBENR.ADCEN=1, 然后继续设置 CMU_APBENR.ADCEN=0, 如此循环, 直到 ADC 转换完成。

1. 16 DMA模块

注意事项 1: 外设与 SRAM 间传输数据, 建议 DMA 配置 single 方式。

注意事项 2: 同一 DMA 通道两次请求间隔时间过短时, 会出现“DMA 状态 6 死锁”现象, 此时 DMA_STATUS.STATUS 值保持 6, DMA 工作失效。为确保不会出现“DMA 状态 6 死锁”现象, 同一 DMA 通道两次请求间隔需大于 28 个系统时钟周期。

若因不可控因素发生了“DMA 状态 6 死锁”问题, 则需在程序主循环中重启外设对应的 DMA 请求使能, 如此, 可清除二次 DMA 请求, 使得 DMA 退出状态 6, 重新回归空闲态, DMA 功能恢复。如: GP16C2T1 通道捕获/比较 1 请求 DMA, 则在主循环中轮询 DMA_STATUS.STATUS, 若检测到该寄存器值为 6, 则将 GP16C2T1_DMAEN.CH1DE 位先清 0 再写 1。

DMA 模块有寄存器 DMA_CHSTA（地址：0x4000CE18，该寄存器暂未在参考手册中开放）可用于检测处于工作状态的通道编号，程序主循环检测到 DMA 状态 6 时，可读取该寄存器判断当前通道号，进而定位到导致 DMA 发生状态 6 的请求源，重启该请求源即可。

1.17 低功耗模式

注意事项 1： 芯片执行 WFI 进入低功耗模式前：

- 如果 SysTick 为使能状态，需要临时关闭 SysTick 计数器使能位(ENABLE@SysTick->CTRL)；
- 如果有非唤醒源的中断为使能状态，需要临时禁止对应的 IRQ 中断 (NVIC_DisableIRQ(XXX_IRQn))。

第2章 硬件开发注意事项

2.1 外部晶振

外部高速振荡器的典型应用连接：

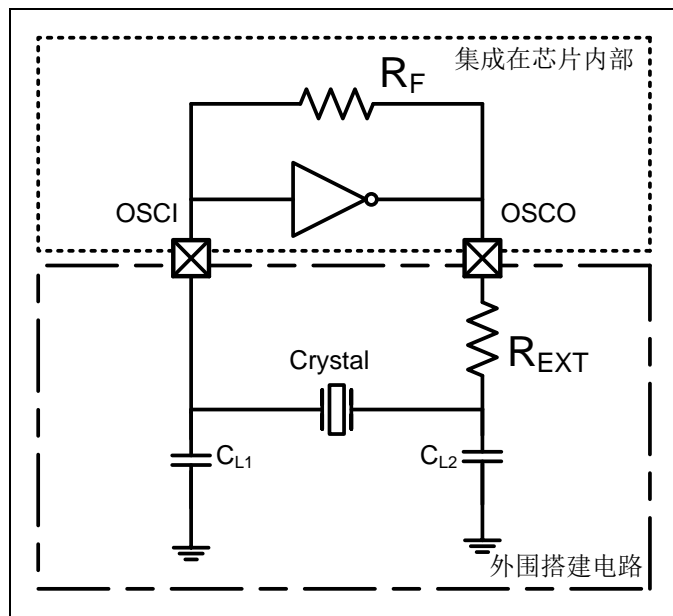


图 2-1 外部高速振荡器连接图

注意事项 1： R_{EXT} 阻值取决于晶振规格特性；

注意事项 2： 当晶振频率不低于 16MHz 时，需选用 ESR 不高于 30Ω 的晶振，且 C_{L1} ， C_{L2} 容值需小于 5pF。

2.2 GPIO PA7/8/9

如果应用系统存在较强的负电压脉冲信号干扰，务必对 PA7，PA8 和 PA9 这 3 个 IO 端口分别串接 100 欧姆以上的防护电阻以及对地并接防护二极管，所接电阻、二极管的位置需尽可能靠近芯片，可增强系统的抗干扰能力；同时这 3 个 IO 端口不建议用作过零检测功能。

当 PA7，PA8 和 PA9 这 3 个 IO 端口用作输出驱动时，建议每个端口的拉电流(I_{OH})不超过 10mA。