

文档编号: AN2018

上海东软载波微电子有限公司

应用笔记

ES32F36xx

修订历史

版本	修订日期	修改概要
V1.0	2019-12-16	初版
V1.1	2020-09-27	1. 增加 CAN 模块应用注意; 2. 增加电源模块应用注意。
V1.2	2020-11-20	1. 修改电源模块外围参数; 2. 细化 USB 模块应用注意; 3. 添加外部晶振部分应用注意; 4. 增加 GPIO 模块应用注意; 5. 添加 UART/SPI/I2S/I2C 模块 FIFO 部分应用注意。
V1.3	2021-04-13	1. 调整文档目录结构; 2. 在 GPIO 模块中添加 IO 中断延时应用注意; 3. 增加 GPIO 边沿中断应用注意; 4. 增加 PA9/PA10 上电高脉冲应用注意; 5. 在 FLASH 模块添加误擦除应用注意; 6. 增加配置字部分应用注意; 7. 增加 UART 模块应用注意; 8. 增加 SPI 用 DMA 方式接收应用注意; 9. 增加系统时钟为 96MHz 时的应用注意; 10. 增加 ADC 部分应用注意; 11. 增加 DMA 应用注意; 12. 增加 SPI 速度表; 13. 增加系统进入低功耗时 PH2 的配置注意; 14. 增加 USB 外设不用时 USB 相关引脚的连接注意; 15. 增加 FLASH 等待时间表 16. 增加 RTC/TSENSE 模块应用注意
V1.4	2021-05-27	1. 增加 LOSC 不用时应用注意

地 址：中国上海市龙漕路 299 号天华信息科技园 2A 楼 5 层

邮 编：200235

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承担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	概述	4
1.1	开发环境	4
1.2	库函数选择	4
1.3	寄存器写保护	4
1.3.1	系统写保护	4
1.3.2	RTC 写保护	4
1.3.3	TSENSE 写保护	4
1.3.4	IWDT 写保护	4
1.3.5	WWDT 写保护	5
1.4	写 1 清零寄存器	5
1.5	位带操作	5
第 2 章	系统控制	6
2.1	系统时钟	6
2.1.1	内部高速 24MHz(默认时钟)	6
2.1.2	外部时钟 HOSC(4~16MHz)	6
2.1.3	96MHz	6
2.1.4	72MHz	6
2.1.5	系统总线默认时钟	7
2.1.6	外部低速时钟(LOSC)	7
2.2	外部晶振	7
2.3	IAP 操作程序	7
2.4	FLASH 模块	7
2.5	配置字	8
第 3 章	外设	9
3.1	GPIO 模块	9
3.2	DMA 模块	9
3.3	CAN 模块	9
3.4	UART 模块	9
3.5	SPI 模块	10
3.6	IIC 模块	10
3.7	FIFO(UART/SPI/I2S/I2C 模块)	10
3.8	TIMER 模块	10
3.9	QSPI 模块	10
3.10	ADC 模块	10
3.11	RTC 模块	11
3.12	TSENSE 模块	11
3.13	USB 模块	11
第 4 章	最小系统电路	12
4.1	LQFP100 封装芯片最小系统电路	12
4.2	LQFP64 封装芯片最小系统电路	13
4.3	LQFP48 封装芯片最小系统电路	14

第1章 概述

1.1 开发环境

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

1.2 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，提供更为人性化的 API，适合大部分用户；

MD：基本上只提供寄存器位域级别的“读”、“写”接口，适合对芯片底层较为熟悉的用户。

对于一些复杂模块，如 QSPI、USB、EBI 等，由于用户独立配置相关寄存器实现对应功能较为繁琐，故这些模块仅提供 ALD 库，不再提供 MD 库。

如果用户对速度不是要求非常严格，一般情况下推荐用户使用 ALD 库。可以减少用户学习时间，增加代码可移植性，最终缩短用户产品的开发周期。

1.3 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，GPIO，RTC，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。

1.3.1 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。

1.3.2 RTC写保护

对 RTC_WPR 寄存器以字方式写入 0x55AAAA55 会解除写保护，写入其他值使能写保护。

可通过读 RTC_WPR 寄存器确认 RTC 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 RTC 模块处于写保护解除状态。

该寄存器保护除自身外的 RTC 所有寄存器。

1.3.3 TSENSE写保护

对 TSENSE_WPR 寄存器以字方式写入 0xA55A9669 会解除写保护，写入其他值使能写保护。

可通过读 TSENSE_WPR 寄存器确认 TSENSE 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 TSENSE 模块处于写保护解除状态。

该寄存器保护除自身外的 TSENSE 所有寄存器。

1.3.4 IWDWT写保护

对 IWDWT_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 IWDWT_LOCK 寄存器确认 IWDWT 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。

1.3.5 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。

1.4 写 1 清零寄存器

中断标志寄存器都是用“写 1 清零”的方式来操作。对于“写 1 清零”的寄存器，不可使用“读-修改-写”的方式来进行“写 1 清零”，否则会引起标志误清，进而产生漏中断的后果。对该类寄存器操作需要以字方式进行写。

1.5 位带操作

位带扩展区把每个 bit 扩展为一个 32-bits 的字，通过访问这些字可达到访问原始 bit 的目的；某个 bit 所在字的地址为 A，位序号为 N($0 \leq N \leq 31$)，则该 bit 位带扩展后的地址为：

SRAM: $\text{AliasAddr} = 0x22000000 + (A - 0x20000000) \times 32 + N \times 4$

外设: $\text{AliasAddr} = 0x42000000 + (A - 0x40000000) \times 32 + N \times 4$

库函数中提供位带操作 API:

RAM 位带: `void BITBAND_SRAM(uint32_t *addr, uint32_t bit, uint32_t val);`

外设位带: `void BITBAND_PER(volatile uint32_t *addr, uint32_t bit, uint32_t val);`

第2章 系统控制

2.1 系统时钟

系统上电默认使用内部 2MHz/24MHz 高速时钟(HRC)作为系统时钟，其中 2MHz 或 24MHz 由配置字中的 HRCSF_SEL 决定。若 HRC 是 2MHz，则在 SDK 的"startup_es32f36xx.s"文件中将其切换到 24MHz，即进入 main 函数后，HRC 均是 24MHz。

可以使用 API 获取系统时钟：`system_clock = ald_cmu_get_sys_clock();`

几种常用系统时钟配置：

2.1.1 内部高速 24MHz(默认时钟)

此种系统时钟不需要用户做任何配置。

2.1.2 外部时钟HOSC(4~16MHz)

外部高速时钟要求为 4MHz 的倍数，如：4MHz、8MHz、12MHz、16MHz。

首先要确认焊接了外部高速时钟，并已知外部高速时钟的频率，假如外部高速时钟为 12MHz，则配置方式如下：

```
ald_cmu_clock_config(CMU_CLOCK_HOSC, 12000000);
```

2.1.3 96MHz

使用 HRC 倍频，配置方式如下：

```
ald_cmu_pll1_config(CMU_PLL1_INPUT_HRC_6, CMU_PLL1_OUTPUT_96M);
```

```
ald_cmu_clock_config(CMU_CLOCK_PLL1, 96000000);
```

使用 HOSC 倍频，外部高速时钟要求为 4MHz 的倍数，如：4MHz、8MHz、12MHz、16MHz。

首先要确认焊接了外部高速时钟，并已知外部高速时钟的频率，假如外部高速时钟为 12MHz，则配置方式如下：

```
ald_cmu_pll1_config(CMU_PLL1_INPUT_HOSC_3, CMU_PLL1_OUTPUT_96M);
```

```
ald_cmu_clock_config(CMU_CLOCK_PLL1, 96000000)
```

注意事项：在系统时钟为 96MHz 时 FLASH 的等待时间至少需要设置为 3 个时钟周期，即 `MSC.MEMWAIT.FLASH_W` 在系统时钟为 96MHz 时不能小于 3。对 `MSC.MEMWAIT.FLASH_W` 的设置需要放在 main 函数的第一行。但这样会降低系统的执行效率，为了解决执行效率问题，可以将核心部分代码链接到 SRAM 的最后 16K 位置。

2.1.4 72MHz

使用 HRC 倍频，配置方式如下：

```
ald_cmu_pll1_config(CMU_PLL1_INPUT_HRC_6, CMU_PLL1_OUTPUT_72M);
```

```
ald_cmu_clock_config(CMU_CLOCK_PLL1, 72000000);
```

使用 HOSC 倍频，外部高速时钟要求为 4MHz 的倍数，如：4MHz、8MHz、12MHz、16MHz。

首先要确认焊接了外部高速时钟，并已知外部高速时钟的频率，假如外部高速时钟为 12MHz，则配置方式如下：

```
ald_cmu_pll1_config(CMU_PLL1_INPUT_HOSC_3, CMU_PLL1_OUTPUT_72M);
```

```
ald_cmu_clock_config(CMU_CLOCK_PLL1, 72000000);
```

2.1.5 系统总线默认时钟

系统时钟	HCLK1	HCLK2	PCLK1	PCLK2
96MHz	1/2	1/2	1/2	1/4
72MHz	-	-	-	1/4
48MHz	-	-	-	1/2
36MHz	-	-	-	1/2

表 2-1 系统各总线默认分频

注 1: 使用 ALD 库函数时, 库函数已对上述各总线时钟进行相应分频;

注 2: 若没有特殊需求, 建议外部高速晶振使用 12MHz, 可以与 SDK 保持一致。

2.1.6 外部低速时钟(LOSC)

首先要确认焊接了外部低速时钟, 配置方式如下:

`ald_cmu_clock_config(CMU_CLOCK_LOSC, 32768);`

需要注意的是, 当系统时钟配置为低速时钟时(低于 1MHz), SysTick 中断将会被迫关闭。ALD 提供的延迟类函数禁止使用。

2.2 外部晶振

外部高速振荡器的典型应用连接:

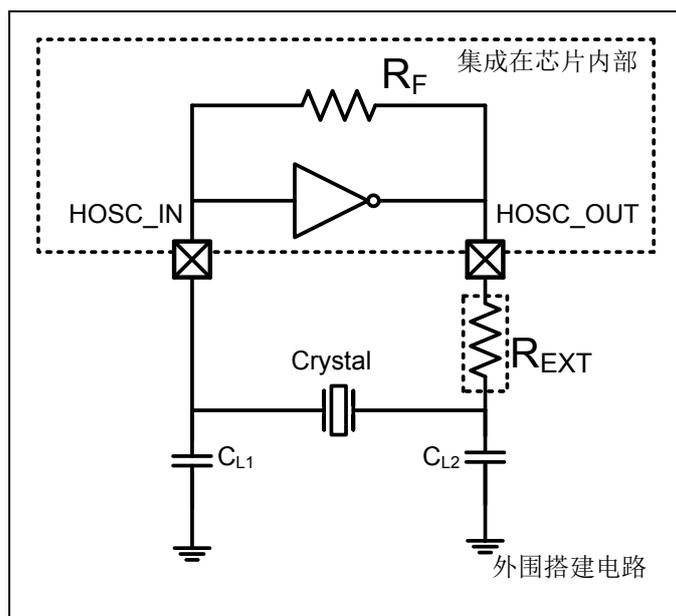


图 2-1 外部高速振荡器连接图

注 1: R_{EXT} 阻值取决于晶振规格特性;

注 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。

2.3 IAP操作程序

芯片内置 IAP 自编程固化模块, 由硬件电路实现。推荐使用 IAP 方式对 FLASH 进行擦、写操作, 可以减少用户代码量。

2.4 FLASH模块

注意事项 1: 读保护规则描述: 当 FLASH 的读保护级别设置为 level1 或 level2 时, 运行在 SRAM

上的程序不能有读 FLASH 操作。

典型应用 A: 运行的 SRAM 上的程序想读取 FLASH。将读 FLASH 操作放在 FLASH 上执行；

典型应用 B: 程序运行在 SRAM 上，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（内核寄存器 SCB->VTOR，地址为 0xE000ED08）。

注意事项 2: 芯片进行擦除过程中发生芯片复位，会导致 Page0 被误擦除。故进行 Flash 擦除操作时需要确保芯片供电稳定。在某些强干扰应用方案中，考虑 VDD 存在波动，在 Flash 执行擦、写前，可配合 LVD 模块对电压进行检测。当电压过低时，禁止执行擦、写操作。或者避开对 Page0 的使用。关于如何避开 Page0，请参考应用笔记<AN2020>。

注意事项 3: MSC_MEMWAIT 寄存器的 FLASH_W 位配置访问 Flash 时的等待系统时钟数。常用系统时钟访问 Flash 时的等待时钟数如下表所示：

系统时钟	96MHz	72MHz	48MHz	24Mhz
FLASH_W	≥3	≥2	≥1	无需等待

表 2-2 常用系统时钟访问 Flash 等待时钟数

2.5 配置字

注意事项 1: 使用 ELink II/ESLink II mini 修改芯片配置字之后，芯片需要断电后重新上电，才能正常调试。

注意事项 2: 使用 ESBurner 擦除芯片后，需要执行配置字编程，否则将导致芯片运行不正常。即点击“擦除”按钮后，需要点击“配编”按钮。

注意事项 3: MCU 工作在临界电压时，程序可能跑飞。故 BOR 配置字必须配置为 2.6V 及以上。

注意事项 4: 外部 LOSC 不使用时，配置字需关闭 LOSC 硬件强制使能。

注意事项 5: 为保证配置字的正常 load，HRCSF_SEL 必须选择 2MHz。如下图所示：



图 2-2 BORVS 和 HRCSF_SEL 选择

第3章 外设

3.1 GPIO模块

注意事项 1: 未使用的 IO 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: 备份域 IO(PC13/PC14/PC15)禁止高速翻转，翻转频率不能大于 1Hz。

注意事项 3: 最大支持 2 个 IO 同时强驱动(20mA)。

注意事项 4: 外部中断的有效电平宽度要大于 100uS。

注意事项 5: IO 的中断延时约 200uS，使用 IO 中断唤醒低功耗模式(STOP1/STOP2/STANDBY)时，其唤醒时间也需要增加 200uS 左右。

注意事项 6: 为进一步降低功耗，系统进入低功耗前，PH2 引脚可设置为数字输入。

3.2 DMA模块

注意事项: DMA 搬运次数最大可配置为 1024，对应的字节数可根据"src_size"参数确定：

src_size=0(字节)，最大搬运字节数为 1024-bytes；

src_size=1(半字)，最大搬运字节数为 2048-bytes；

src_size=2(字)，最大搬运字节数为 4096-bytes。

3.3 CAN模块

注意事项 1: 当 MCU 使用中断方式接收，通信速率是 1Mbits/s，并且帧间隔小于 30uS 时，则 MCU 的主频应不小于 48MHz，否则会有丢帧风险。

注意事项 2: 通信速率范围 10Kbits/s--1Mbits/s。当速率低于 20Kbits/s 时，要求 PCLK1 频率不超过 24MHz。

注意事项 3: 当接收到一帧数据时，需要检查 0x4000B030 地址 bit21，若该位被置位，则该帧可能为错误帧，应将对应 RXFIFO 中的数据全部丢弃。具体实现可参考 ALD 库函数中 __can_rx_check() 函数。

注意事项 4: 由于 CAN 标准中要求时钟精度在 1.58%以内，故当产品的使用环境比较恶劣时，建议使用外部高精度晶振。外部晶振的频率可以使用 4MHz、8MHz、12MHz 或 16MHz，推荐使用 12MHz(和 ES-PDS 板保持一致)。

3.4 UART模块

注意事项 1: LIN 模式下发送 Break 之后，应在 2 个 bit 位宽内发送下一 bit 数据。

注意事项 2: LIN 模式下发送数据前需要关闭 RX。

注意事项 3: UART 单线模式下最大波特率不超过 230400bps；

注意事项 4: UART 轮询接收模式，使用 STAT.RXFH 方式接收数据，建议设置 RX FIFO 为 1 字节；

注意事项 5: 通过 DMA 发送帧数据时，一帧数据的第一个字节需要直接写 TBR 寄存器，后续字节再使用 DMA 方式发送。具体实现可参考 SDK 中对应例程：

~\ES32_SDK\Projects\ES32F36xx\Examples_ALD\UART\06_send_recv_by_dma_fast\

注意事项 6: 配合 DMA 使用时，DMA 的 burst 必须为 1，R_power 必须为 0；

注意事项 7: 当发生溢出错误时，需要将 RX_FIFO 进行复位。

注意事项 8: 使用中断方式发送数据时，需要按照如下顺序进行：先清除 TXS 中断，再向 TBR 寄存器写入数据。

3.5 SPI模块

注意事项 1: 不支持 SPI 内部 CRC 功能;

注意事项 2: I2S 模块只支持主机模式, 不支持 I2S 从机模式;;

注意事项 3: SPI 通信速度(使用 API: ald_spi_send_byte_fast()/ald_spi_rcv_byte_fast()):

系统时钟 (常温)	主机		从机	
	发送	接收	发送	接收
24MHz	12Mbps	3Mbps	1.5Mbps	3Mbps
48MHz	24Mbps	6Mbps	3Mbps	6Mbps
72MHz	36Mbps	9Mbps	4.5Mbps	9Mbps
96MHz	24Mbps	6Mbps	3Mbps	6Mbps

表 3-1 SPI 模块通信速度

注意事项 4: SPI 在配置为从机且时钟相位配置为第一个边沿采样时, 要求在主机发起传送前写入数据。

注意事项 5: 单线模式下, 使用中断方式发送数据时, 需按照如下顺序操作: 1 使能 SPI; 2 使能发送空中断; 3 向 DATA 寄存器写入数据 4 等待进入中断。

注意事项 6: SPI 通过 DMA 方式接收数据时, DMA 可能会进入 Halt 状态, 导致数据无法正常接收。故不能使用 DMA 方式进行数据接收。

3.6 IIC模块

注意事项 1: 通过 DMA 方式收发数据时, 一帧的数据长度最大为 255 字节。

注意事项 2: 控制寄存器 CON1 的 DNF 位不能置位, 故数字滤波功能无法使用。

3.7 FIFO(UART/SPI/I2S/I2C模块)

注意事项 1: 向 TX_FIFO 中写入多个数据时, 每次写数据前确认 FIFO 是否满, 避免 FIFO 上溢。

注意事项 2: 在中断接收模式下, 需要在接收中断函数中将 RX_FIFO 读空;

以上情况在 ALD 库中已做了处理, 使用 ALD 库无需关注。使用 MD 库时需自行处理。

3.8 TIMER模块

注意事项 1: 使用滤波搭配 Timer clock divider(CKD)时, 会有错误情形, 建议避免使用 CKD 搭配滤波功能。

注意事项 2: 在递减模式中, 若写入 AR 寄存器的值大于当前计数值(COUNT), 则需要下次重载时才起作用。

3.9 QSPI模块

注意事项: QSPI 模块通信速度不能超过 48Mbps (即每根数据线的最大速率为 12Mbps)。

3.10 ADC模块

注意事项: ADC1 和 ADC0 共用一个参考源, 对该参考源的配置统一使用 ADC0.ADC_CCR 寄存器。ADC1.ADC_CCR 寄存器有作用的只有 3 个位域: CKDIV/DIFFEN/TRMEN。

3. 11 RTC模块

注意事项 1: 使用 RTC 模块时, APB2 总线时钟(PCLK2)不能超过 12MHz。例如: 主频为 72MHz 时, 需要对 PCLK2 进行至少 8 分频, 分频可使用如下库函数:

ALD: ald_cmu_div_config(CMU_PCLK_2, CMU_DIV_8);

MD: md_cmu_div_config(MD_CMU_PCLK_2, MD_CMU_DIV_8);

注意事项 2: RTC 模块的中断标志晚于中断信号出现, 即进入 RTC 中断后需要等待 2 个时钟周期才能正确读到中断标志。操作方式可参考例程:

~\ES32_SDK\Projects\ES32F36xx\Examples_ALD\RTC\

3. 12 TSENSE模块

注意事项: TSENSE 模块的中断标志晚于中断信号出现, 即进入 TSENSE 中断后需要等待 2 个时钟周期才能正确读到中断标志。操作方式可参考例程:

~\ES32_SDK\Projects\ES32F36xx\Examples_ALD\TSENSE\

3. 13 USB模块

注意事项 1: USB 模块的推荐系统时钟为: 36MHz/48MHz/72MHz/96MHz;

注意事项 2: USB 模块 PHY 时钟固定为 12MHz, 推荐使用外部 12MHz 晶振[与示例代码一致]。

注意事项 3: USB 模块引脚说明

Pin	LS	FS	HS	描述
USB_DM	✓	✓	✓	USB 差分数据线, 高速模式下板级走线注意防护
USB_DP	✓	✓	✓	USB 差分数据线, 高速模式下板级走线注意防护
VDD33_USB	✓	✓	✓	USB 模块电源, 需连接到 3.3V 电源上
USB_REXT	-	-	✓	串一个 1%精度 12.7KΩ 电阻到地, 高速模式专用
USB_ID	✓	✓	✓	输入端口, 高电平时 USB 模块工作在 Device 模式[默认], 低电平时 USB 模块工作在 Host 模式。采用 Micro-AB 接口时, Device 模式可以直接使用, Host 模式需连接 OTG 转接头, 在 OTG 转接头中会将 USB_ID 引脚拉低, 让 USB 控制器工作在 Host 模式。
USB_VBUS	✓	✓	✓	输入端口, 需接到 Micro-USB 接口的 VDD 引脚上; 注 1: 当 USB 模块工作在 Host 模式时, 此引脚电压不能低于 4.75V; 注 2: 当 USB 模块工作在 Device 模式时, 此引脚电压不能低于 2.1V; 注 3: MCU 进入低功耗模式后, 若此引脚上有电压, 则会产生 17uA 左右的漏电。

表 3-2 USB 模块引脚说明

注意事项 4: 当系统运行 96MHz 时, ALD 库默认对 HCLK1 总线时钟进行 2 分频。由于 USB 内置 DMA 不支持 HCLK1 总线分频, 故当系统运行 96MHz 时, 禁止使用 USB 内置 DMA。若使用我司提供的 USB 协议栈, 当系统运行 96MHz 时, 请在工程中添加 “USB_DMA_N_SUPPORT” 宏, 此时自动使用外部 DMA 进行数据搬运。

注意事项 5: 不使用 USB 功能时, USB 相关引脚需按如下方式处理:

VDD33_USB: 悬空或接 3.3V; UAB_DM/USB_DP/USB_REXT/UAB_ID/USB_VBUS: 接地。

第4章 最小系统电路

4.1 LQFP100 封装芯片最小系统电路

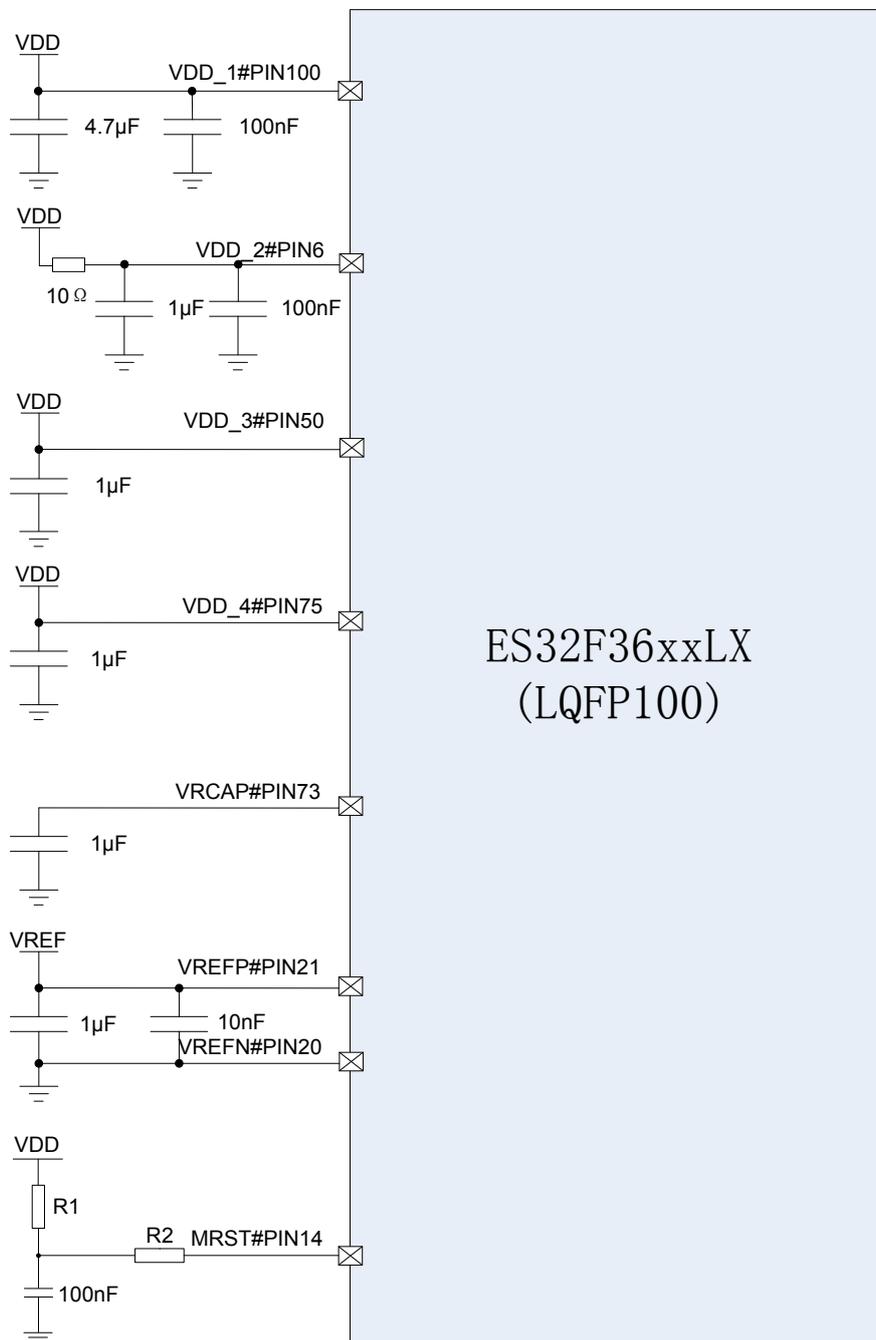


图 4-1 LQFP100 封装芯片最小系统电路

注 1: 每一组电源必须连接如图所示的陶瓷耦合电容。这些电容必须尽可能地靠近芯片的相应管脚, 才能保证芯片的运行性能。

注 2: PIN100 为芯片主电源, PIN6 为芯片备份域电源, 需特殊防护。

注 3: VREFP 和 VREFN 管脚在复用为 ADC 外部参考时才需要外接电容。

注 4: 对于支持 USB 模块的系列芯片, USB 部分推荐电路参考应用笔记 AN2021。

注 5: MRST 引脚采用 RC 复位, 其中 $47\text{K}\Omega \leq R1 \leq 100\text{K}\Omega$, 电容 $C1=100\text{nF}$, $R2$ 为限流电阻, $0.1\text{K}\Omega \leq R2 \leq 1\text{K}\Omega$ 。

4.2 LQFP64 封装芯片最小系统电路

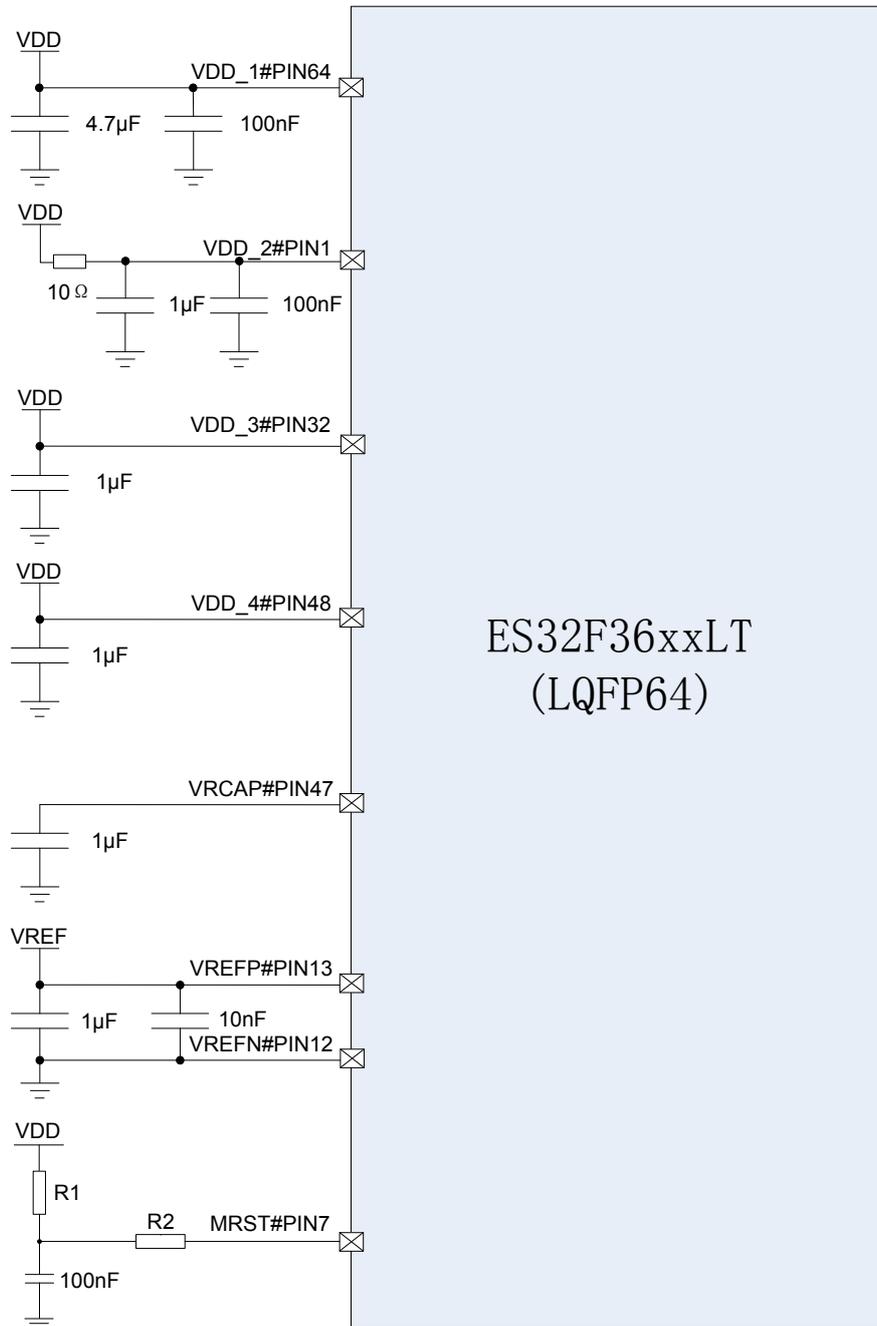


图 4-2 LQFP64 封装芯片最小系统电路

注 1: 每一组电源必须连接如图所示的陶瓷耦合电容。这些电容必须尽可能地靠近芯片的相应管脚, 才能保证芯片的运行性能。

注 2: PIN64 为芯片主电源, PIN1 为芯片备份域电源, 需特殊防护。

注 3: VREFP 和 VREFN 管脚在复用为 ADC 外部参考时才需要外接电容。

注 4: 对于支持 USB 模块的系列芯片, USB 部分推荐电路参考应用笔记 AN2021。

注 5: MRST 引脚采用 RC 复位, 其中 $47\text{K}\Omega \leq R1 \leq 100\text{K}\Omega$, 电容 $C1=100\text{nF}$, $R2$ 为限流电阻, $0.1\text{K}\Omega \leq R2 \leq 1\text{K}\Omega$ 。

4.3 LQFP48 封装芯片最小系统电路

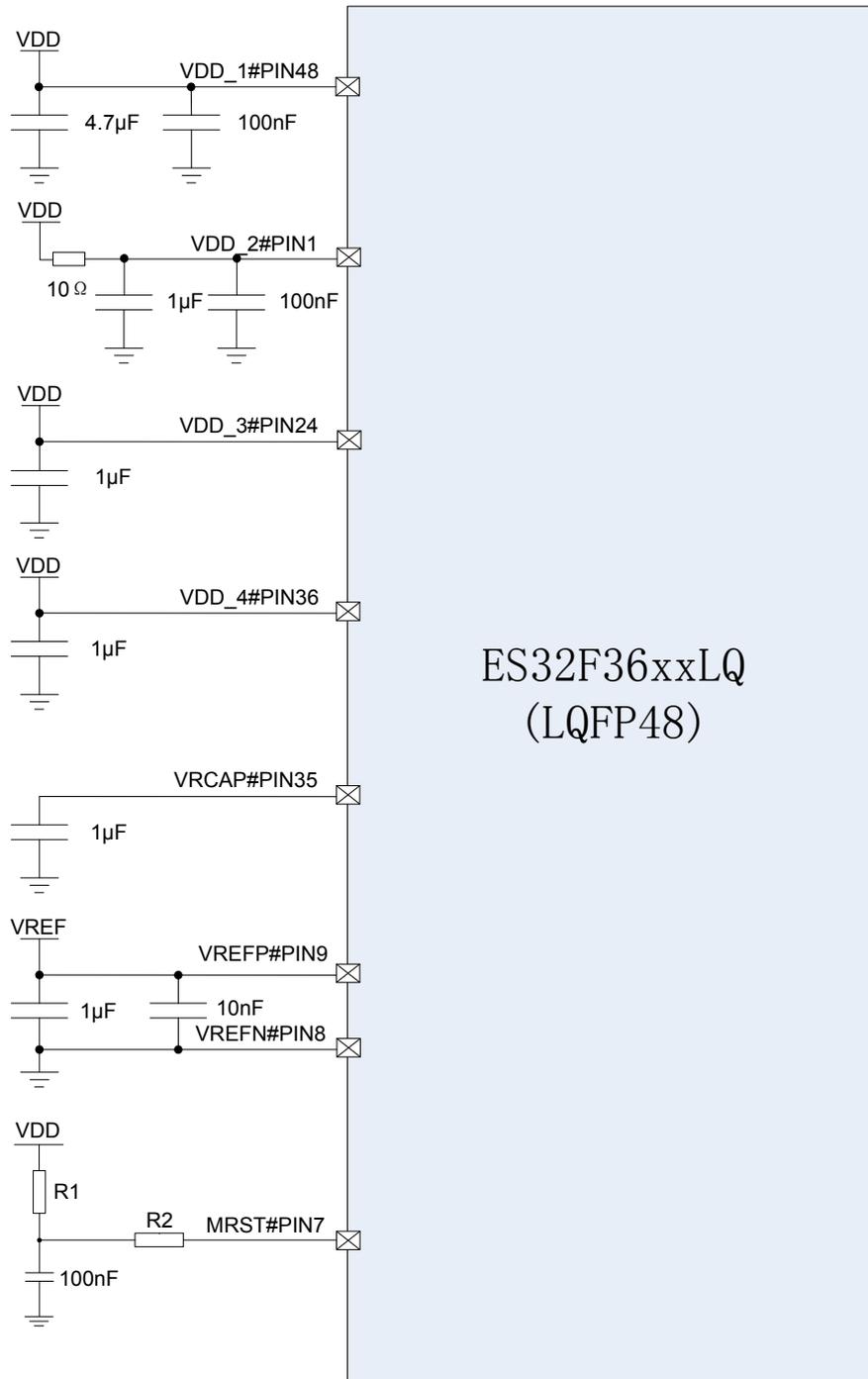


图 4-3 LQFP48 封装芯片最小系统电路

注 1: 每一组电源必须连接如图所示的陶瓷耦合电容。这些电容必须尽可能地靠近芯片的相应管脚, 才能保证芯片的运行性能。

注 2: PIN48 为芯片主电源, PIN1 为芯片备份域电源, 需特殊防护。

注 3: VREFP 和 VREFN 管脚在复用为 ADC 外部参考时才需要外接电容。

注 4: 对于支持 USB 模块的系列芯片, USB 部分推荐电路参考应用笔记 AN2021。

注 5: MRST 引脚采用 RC 复位, 其中 $47\text{K}\Omega \leq R1 \leq 100\text{K}\Omega$, 电容 $C1=100\text{nF}$, $R2$ 为限流电阻, $0.1\text{K}\Omega \leq R2 \leq 1\text{K}\Omega$ 。