

文档编号: AN2045

上海东软载波微电子有限公司

# 应用笔记

---

**ES32VF2264**

## 修订历史

版本	修订日期	修改概要
V1.00	2023-12-12	初版
V1.01	2024-04-18	1. 优化中断嵌套说明； 2. 更新 CDK 下载网址； 3. 优化 I2C、UART 和 SPI 注意事项。
V1.02	2024-07-29	1. GPIO 注意事项补充 IO 高速说明； 2. SPI 注意事项补充高速 SPI 主机配置说明； 3. DMA 注意事项补充外设到外设模式配置说明。
V1.03	2024-09-11	1. IAP 操作程序小节补充 FLASH 编程注意事项
V1.04	2024-10-11	1. 补充 32pin 封装 SPI1 使用注意事项
V1.05	2024-11-29	1. 补充 DMA 存储器到外设模式注意事项
V1.06	2025-07-10	1. 新增低功耗模式注意事项； 2. DMA 模块补充 ADC 使用 DMA 的注意事项。

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

<b>第 1 章</b>	<b>嵌入式软件开发注意事项 .....</b>	<b>4</b>
1.1	配置字 .....	4
1.2	开发环境 .....	5
1.3	库函数选择 .....	5
1.4	HOSC 时钟 .....	5
1.5	寄存器写保护 .....	5
1.5.1	系统写保护 .....	6
1.5.2	IAP 操作保护 .....	6
1.5.3	IWDT 写保护 .....	6
1.5.4	WWDT 写保护 .....	6
1.6	IAP 操作程序 .....	6
1.7	GPIO 模块 .....	6
1.8	IWDT 模块 .....	7
1.9	WWDT 模块 .....	7
1.10	I2C 模块 .....	7
1.11	UART 模块 .....	8
1.12	SPI 模块 .....	8
1.13	ADC 模块 .....	9
1.14	DMA 模块 .....	10
1.15	中断 .....	11
1.15.1	中断触发方式 .....	11
1.15.2	中断嵌套 .....	11
1.16	CMU 模块 .....	11
1.17	PMU 模块 .....	11
1.18	低功耗模式 .....	11
<b>第 2 章</b>	<b>硬件开发注意事项 .....</b>	<b>13</b>
2.1	外部晶振 .....	13

## 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（上位机软件）+ES-LinkII-OB/PRO（调试器）对目标芯片进行配置字编程。

注意：调试前必须将 GBRDP 设置为“读保护等级 Level0”，否则无法正常调试，量产烧录前可根据需求自行设置 GBRDP。

**配置字设置**

**配置字**

UserID:  当前缓冲区校验码:  校验码格式:

**配置项**

配置字: 9800 0001 0001 0001 芯片未加密

BOOTADDR	<input type="text" value="0X0000_0000"/>	BORVS	<input type="text" value="2.1V"/>
WWDTEN	<input type="text" value="软件使能后可关闭"/>	IWDTEN	<input type="text" value="由软件使能"/>
WRP0_ENB	<input type="text" value="禁止"/>	WRP0_START	<input type="text" value="Flash Page0"/>
WRP0_END	<input type="text" value="Flash Page 3(默认)"/>	WRP1_ENB	<input type="text" value="禁止"/>
WRP1_START	<input type="text" value="Flash Page0"/>	WRP1_END	<input type="text" value="Flash Page 3(默认)"/>
DAFLS_ENB	<input type="text" value="禁止"/>	DAFLS_START	<input type="text" value="Flash Page0"/>
DAFLS_END	<input type="text" value="Flash Page 3(默认)"/>		
GBRDP	<input type="text" value="读保护等级Level0"/>		
PCROP0_ENB	<input type="text" value="禁止"/>	PCROP0_START	<input type="text" value="Flash Page508"/>
PCROP0_END	<input type="text" value="Flash Page511"/>	PCROP1_ENB	<input type="text" value="禁止"/>
PCROP1_START	<input type="text" value="Flash Page508"/>	PCROP1_END	<input type="text" value="Flash Page511"/>

**完整配置字**

```

9800 67FF FFFF FFFF FFFF 0000 FFFF FFFF FFFF 0000 FFFF FFFF FFFF 0000 FFFF FFFF
0001 FFFE FFFF FFFF FFFF FFFF FFFF FFFF 0001 FFFE FFFF FFFF FFFF FFFF FFFF FFFF
0001 FFFE FFFF 09A9 09AA FFFF FFFF
F656 F655 0000 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 0000 FFFF FFFF
FFFF 0000 FFFF FFFF FFFF 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF 0000 FFFF FFFF
FFFF FFFF FFFF FFFF FFFF 0000 FFFF FFFF
    
```

图 1-1 配置字界面

## 1.2 开发环境

推荐用户使用平头哥剑池集成开发环境 CDK 进行固件开发。

CDK 官方下载地址：<https://www.xrvm.cn/community/download>。搜索关键词“CDK”，找到最新版（V2.20.0 及以上）的“剑池 CDK 集成开发环境”，点击“下载”即可。注意：登录后方可下载。

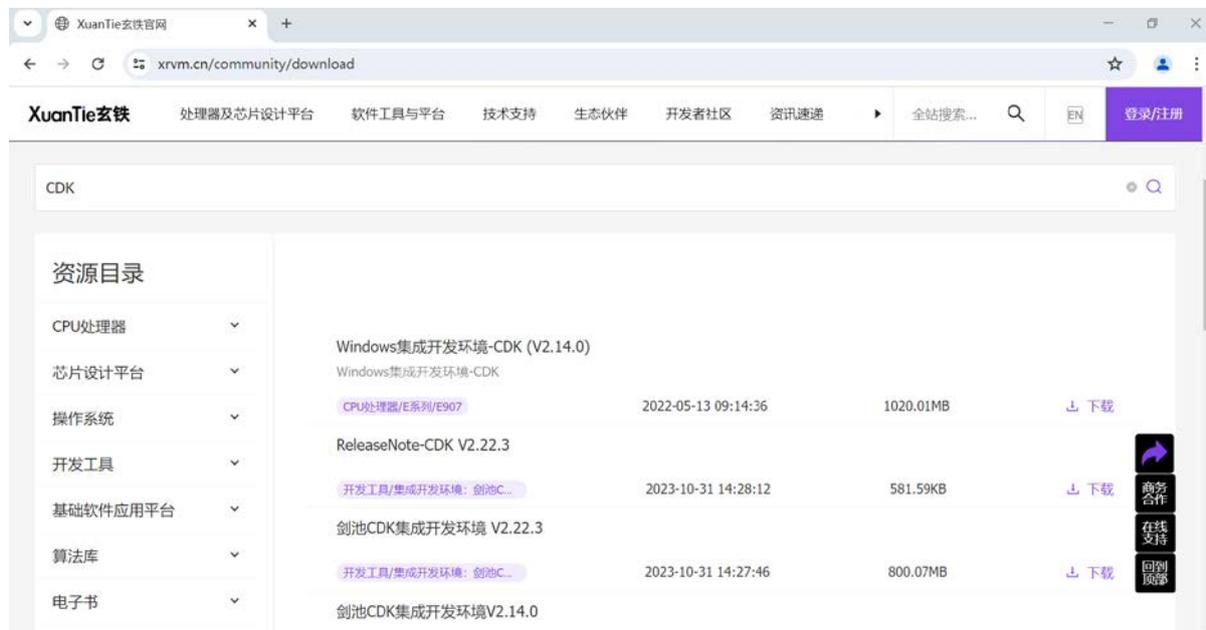


图 1-2 CDK 安装包官方下载网页

详细的 CDK 使用方法请查看《应用笔记\_CDK》。

## 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。

## 1.4 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 HOSCFBPS 位为 1，旁路外部时钟滤波器，以免时钟信号被误滤除。

## 1.5 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。

### 1.5.1 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。

### 1.5.2 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。

### 1.5.3 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。

### 1.5.4 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。

## 1.6 IAP操作程序

**注意事项 1:** IAP 操作时会占用 FLASH 总线，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。

## 1.7 GPIO模块

**注意事项 1:** 未使用的 GPIO 管脚（包括调试管脚）建议设置为输出低电平，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:** IO 高速（20MHz 以上）需设定强电流驱动模式，即相应的 GPIO\_ODRV 寄存器位设

置为 01。

## 1.8 IWDT 模块

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

**注意事项 2:** 当 IWDT 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 IWDT 进行初始化。

**注意事项 3:** IWDT 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 0.5 秒，用户可通过程序修改 IWDT\_LOAD 寄存器来调整计数周期。IWDT 模块固定为使能，IWDT 复位和中断也固定为使能，软件无法关闭，寄存器 IWDT\_CON 的 CLKS, RSTEN, IE, EN 位均无效。

**注意事项 5:** 可靠的系统在休眠下也应该保持 IWDT 处于工作状态，用户可以使用 IWDT 或 WWDT 定时唤醒后对 IWDT 进行喂狗。

## 1.9 WWDT 模块

**注意事项 1:** WWDT 禁止用户在喂狗窗口外喂狗，建议用户在 WWDT 中断服务程序内喂狗。或使用一个全局变量做为软件标志，在中断服务程序里置位软件标志，在主循环里判断软件标志，如果软件标志被置位再喂狗。

## 1.10 I2C 模块

**注意事项 1:** 最大传输 NBYTE

I2C 支持最大  $2^{16}=65536$  Bytes 的传输数量，分别配置 I2C\_CON1.NBYTES 与 I2C\_CON2.NBYTES 寄存器；当传输数量需求大于 65536 时，在传输前配置 I2C\_CON2.RELOAD 位，当第一笔 65536 Bytes 传输完后，SCL 将被拉低，此时可重新填写第二笔传输的 NBYTES 数量；当不使用 NBYTES 重载模式时，需将 I2C\_CON2.RELOAD 清除，如果开启 I2C\_CON2.AUTOEND 功能，I2C 会在 I2C\_CON2.RELOAD 清除后，自动发送 STOP 信号。

**注意事项 2:** 中断标志位处理方式

发送空中断标志位 TXE 置起后，当状态寄存器 STAT 的 TXE 位为 1 时，清除中断标志位无效。所以，TXE 置起后，必须先填写 I2C\_TXDATA，方可清除该中断标志位。

接收非空中断标志位 RXNE 置起后，当状态寄存器 STAT 的 RXNE 位为 1 时，清除中断标志位无效。所以，RXNE 置起后，必须先读空 I2C\_RXDATA 里的数据，方可清除该中断标志位。

**注意事项 3:** 当总线 SDA 遇到异常下拉的波形时，避免传输状态错误的处理方式

请开启检测停止中断功能 (I2C\_IER.STOP)，当 STOP 中断事件发生时，检查 I2C\_STAT.BUSY 是否为 1。如 BUSY 为 1，关闭 I2C\_CON1.PE，再开启 I2C\_CON1.PE。

如果上述状态发生之前，已设置 I2C\_CON2.START，则必须重新设置一次。

## 1.11 UART模块

### 注意事项 1: 自动波特率

若发生侦测波特率超时(UART\_RIF.ABTO)，波特率开关会自动清除(UART\_MCON.ABREN)，建议启用自动波特率侦测时，同时启用重复侦测自动波特率(UART\_MCON.ABRREPT)功能。若自动波特率侦测成功，但数据非预期时，用户需重新使能自动波特率侦测(UART\_MCON.ABREN)功能。

### 注意事项 2: 中断标志位处理方式

发送空中断标志位 TFEMPTY 置起后，当状态寄存器 STAT 的 TFEMPTY 位为 1 时，清除中断标志位无效。所以，TFEMPTY 置起后，必须先填写 UART\_TXBUF，方可清除该中断标志位。

接收非空中断标志位 RFNEMPTY 置起后，当状态寄存器 STAT 的 RFNEMPTY 位为 1 时，清除中断标志位无效。所以，RFNEMPTY 置起后，必须先读空 UART\_RXBUF 里的数据，方可清除该中断标志位。

## 1.12 SPI模块

### 注意事项 1: 发送 FIFO 缓存中断处理方式

SPI 发送 FIFO 中断标志分为两种，发送 FIFO 缓存空中断(SPI\_IFM.TXE)与发送 FIFO 缓存低于阈值中断(SPI\_IFM.TXTH)。当进入中断处理流程时，务必优先填写 SPI\_DATA 再清除中断标志位。根据不同的应用，为了避免填写 SPI\_DATA 时，发生 FIFO 溢出的错误操作，提供以下两种方式检查：

- 方法 1: 确认 FIFO 剩余空间，读取 SPI\_STAT.TXFLV 寄存器，表示 FIFO 已填入数据数量 (Bytes)，可判断再填入多少数据。
- 方法 2: 确认 FIFO 是否已满，读取 SPI\_STAT.TXF 寄存器，表示 FIFO 已满，不可再填入数据。

### 注意事项 2: 接收 FIFO 缓存中断处理方式

SPI 接收 FIFO 中断标志分为三种，接收 FIFO 缓存非空中断(SPI\_IFM.RXNE)、接收 FIFO 缓存满中断(SPI\_IFM.RXF)与接收 FIFO 缓存超过阈值中断(SPI\_IFM.RXTH)。FIFO 功能主要为了提升 SPI 传输效率，因此大多使用 RXF 与 RXTH 中断事件，效率比较显著；当进入中断处理流程时，务必优先读取(搬移)SPI\_DATA，避免 FIFO 满溢造成数据丢失，提供以下两种方式检查：

- 方法 1: 确认 FIFO 缓存空间，读取 SPI\_STAT.RXFLV 寄存器，表示 FIFO 接收 Bytes 数量，可判断读取多少数据。
- 方法 2: 确认 FIFO 是否已空，读取 SPI\_DATA 数据，判断 SPI\_STAT.RXNE 寄存器为 0，表示 FIFO 已空，数据已读取完毕。

**注意事项 3:** 从机模式下, 发送数据操作方法

SPI 配置为从机模式时, 填写 SPI\_DATA 数据的依据, 建议使用发送 FIFO 缓存低于阈值状态标志位(SPI\_STAT.TXTH)或中断(SPI\_IFM.TXTH)。

**注意事项 4:** 高速 SPI 主机配置

当 SPI 作为主机, SCK 达到 18MHz 以上时, 建议如下:

- ①SPI 使用时钟空闲电平为高电平、第二边沿采样;
- ②SPI 通信管脚对应的 IO 配置为强驱动模式;
- ③SPI 通信管脚的负载电容小于 30pF。

**注意事项 5:** SPI 主机单工只接收模式

SPI 单工只接收模式效率较高, 若接收数据传输不及时, 容易造成接收 FIFO 溢出, 如需使用该模式, 有如下注意事项:

- ①须配合使用 DMA 搬运接收数据;
- ②如需 SCK 频率达到 PCLK 的 2 分频, 请将数据帧长度设置为 16 位 (SPI\_CON1.FLEN = 1)。但如果数据帧长度必须设置为 8 位, SCK 设置不得高于 PCLK 的 4 分频;
- ③此模式下无需读取 BUSY 标志, 因为开始通信后此标志一直为 1。

**注意事项 6:** 32 pin 封装 (ES32VF2184LK) SPI1 使用注意事项

由于封装的原因, 如需使用 ES32VF2184LK 的 SPI1 做主机, 需通过 PIS 配置 SCK、MISO 和 MOSI 引脚, 具体方法如下:

**SCK:**

- ①将某个能复用 SPI1\_SCK 但未封装出来的 IO 配置为 SPI1\_SCK (如 PB13 配成 FUNC2);
- ②通过 PIS 将 PB13 转到 PIS\_OUT 输出到任意想设置为的 SPI1\_SCK 的 IO。

**MISO:**

- ①通过 PIS 将任意 IO 转到 PIS\_OUT2 输出 (GPIO 做 PIS 生产端)
- ②通过 “PIS 消费端通道控制寄存器 1 (PIS\_TAR\_CON1)” 将 PIS\_OUT2 配置为 SPI1 的 RX, 这样上面①设置的任意 IO 便成为 SPI1\_MISO。

**MOSI:**

- ①将某个能复用 SPI1\_MOSI 但未封装出来的 IO 配置为 SPI1\_MOSI (如 PC15 配成 FUNC5);
- ②通过 PIS 将 PC15 转到 PIS\_OUT 输出到任意想设置为的 SPI1\_MOSI 的 IO。

通过上述配置后, SPI1 可做主机使用, 但相较于引脚直通模式, 通信速率有所降低, SYSCLK 72MHz 时, SPI1 全双工通信速率最高 8 分频, 仅发送的速率能达到 2 分频。如需速率达到 2 分频, 建议采用该方案配置的 SPI1 仅做发送数据使用, 如驱动屏幕。

## 1.13 ADC 模块

**注意事项 1:** 因 ADC 在使能后需要一定的建立时间, 才能正常工作, 所以在 ADCEN 使能后的第

一次转换结果可能不准确，在应用中需要丢弃该次转换结果。

**注意事项 2:** ADC 正常工作时，需使能 IREFEN=1，否则会导致 ADC 工作异常；在 STOP 模式下 ADC 不工作，需设置 IREFEN=0，否则会增大芯片功耗。

**注意事项 3:** ADC 工作时，需设置 PWRMODSEL=0，选择为高速模式，提高转换结果精度。

**注意事项 4:** 采样 AIN19 通道（内部参考电压 VREF1.2V），ADC 时钟频率需小于 565KHz，采样时间需设置为 18Tadclk。

## 1.14 DMA 模块

**注意事项 1:** DMA 无法直接搬运 Flash 区域的数据。

**注意事项 2:** DMA 控制器工作在外设到外设模式时，需将仲裁前的 DMA 传输次数设置为 1，即设置 DMA\_CONn 寄存器的 MAX\_BURST<3:0>=0，并且对所涉及的每个外设，只能使能该外设的一个 DMA 请求功能。

**注意事项 3:** DMA 存储器到外设模式，在 DMA 通道传输过程中，每次对任意 APB 外设的特殊功能寄存器进行读写操作后，需额外执行 2 个连续的 NOP 指令。为满足上述要求，DMA 存储器到外设模式的应用场景，对 APB 外设寄存器的写读操作，只能采用以下两种方法，且不支持 ALD 库。

① 读写 APB 外设寄存器的操作，调用 MD 库内联函数实现；（建议采用该方法）

② 如需直接写读 APB 外设寄存器，可按以下方式操作：

写 APB 外设特殊功能寄存器，例如，CUART1->BRR 写入 0x271：

```
*(volatile uint32_t *) (0x00080000) = (CUART1->BRR = 0x271);
```

读 APB 外设特殊功能寄存器，例如，读 CUART1->BRR 的值：

```
g_uart_brr = ((CUART1->BRR) | (*(volatile uint32_t *) (0x00080000)));
```

注：\*(volatile uint32\_t \*) (0x00080000) 为固定代码，请勿改动。

### 注意事项 4:

当 ADC 使用 DMA 传输数据时，其他 DMA 通道均需保持空闲，否则可能导致 ADC 数据丢失。

如果一定需要 DMA 传输 ADC 数据的同时，使用其他 DMA 通道传输数据，则必须按如下方法配置 ADC 及其对应的 DMA 通道：

① 标准转换配置单次转换结束时将 STAT.NCHE 置 1，ADC->CON2.NCHESEL=1；

② 将 ADC 转换完成信号通过 PIS 连接到 AD16C4T 的 ITR 端；

③ AD16C4T 的捕获比较更新，配置为当 TI 边沿上升时，均会被更新，

AD16C4T->CON2.CCUSEL=1；

④ 使能 AD16C4T COM 访问 DMA，AD16C4T->DMAEN.COMDS=1；

⑤ 使用 AD16C4T 的 COM 请求 DMA 传输。

注意：按照上述方法配置后，AD16C4T 不能再被用作从机配置，但作为主机，PWM、计数、输入/捕获功能不受影响。且该方案不支持 ADC 连续转换+DMA 循环模式。

### 注意事项 5:

ADC 连续转换+DMA 循环模式，SRAM 转存的 ADC 数据需要被随时读取，则采用如下方案：禁

止其他 DMA 通道，仍使用 ADC 的原始 DMA 请求，ADC 通过一个 DMA 通道将数据先搬运到 CRC->DATA 寄存器，通过 PIS -> AD16C4T ITR -> AD16C4T COM -> DMA，触发另一个 DMA 通道，将 CRC 的 DATA 搬运到 SRAM 指定位置。

注意：该方案会影响 CRC 模块，且不允许其他 DMA 通道工作。

## 1.15 中断

### 1.15.1 中断触发方式

**注意事项 1:** ADC 中断源信号是 pulse 高脉冲，而其他中断源信号是电平。所以 ADC 的中断触发方式应配置为上升沿触发（CLICINTATTR[2:1] = 1）；而其他中断触发方式应配置为电平触发（CLICINTATTR[2:1] = 0）。SDK 中的中断配置函数 `ald_mcu_irq_config` 和 `md_mcu_irq_config` 已按照上述规则区分 ADC 中断配置和其他中断配置，用户直接调用即可。

### 1.15.2 中断嵌套

**注意事项 1:**

欲实现中断嵌套，需满足以下几点：

- ①使能全局中断；
- ②确保需要响应的中断使能；
- ③确保需要响应的中断产生；
- ④确保需要响应的中断优先级高于当前中断（CLICINTCTL 配置的值越大，优先级越高）。

如需中断嵌套，请参考 SDK 里的 CLIC 文件下的例程 `01_nest_interrupt`，工程配置 `Compiler` 标签 `Define` 栏需包含 `NEST_INT_ENABLE`。

## 1.16 CMU模块

**注意事项 1:** 若要进行系统时钟切换，待切换的时钟需先使能并 `ready`，避免切换异常。

**注意事项 2:** 在软件进行系统时钟切换到新的时钟源时，需先将系统时钟 `SYSCCLK` 设置为 1 分频（`CMU_CFGR` 寄存器的 `SYSDIV` 位为 0），然后再执行系统时钟切换命令，否则可能会导致原时钟源无法被关闭。

## 1.17 PMU模块

**注意事项 1:**

关于 PMU 唤醒时间寄存器 `TWUR` 使用建议如下：

- 1.浅睡眠 `SLEEP`：可以设置 `TWUR = 0`；唤醒时间约 7 个系统时钟周期。（从中断源产生到 CPU 开始运行程序，实际应是“系统时钟周期\*（`TWUR+7`）”）。
- 2.深睡眠 `STOP`，对应唤醒时间为 `12*LRC+系统时钟周期*TWUR`。
  - 2.1 系统时钟为外振 `HOSC` 或 `PLL` 时钟，建议设置 `TWUR = 0xFFF`（最大值）；
  - 2.2 系统时钟为内振 `HRC48M` 或 `HRC4M` 时钟，建议设置 `TWUR = 0x3FF`；
  - 2.3 系统时钟为内振 `LRC` 时钟，建议设置 `TWUR = 0xF`。

## 1.18 低功耗模式

**注意事项 1:** 芯片执行 `WFI` 指令进入低功耗模式前：

- 如果内核计时器为使能状态，需要临时关闭 `CLINT` 中断（`csi_vic_disable_sirq(CLINT_IRQn)`）；

- 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（csi\_vic\_disable\_sirq (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:

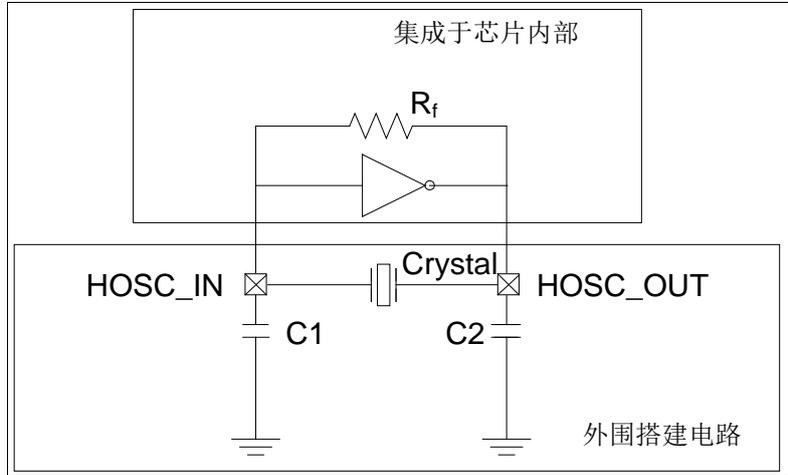


图 2-1 外部高速振荡器连接图

**注意事项 1:** 晶体振荡器与芯片配合使用时, 为达到理想的晶振起振和稳定工作状态, 对 1~8MHz 晶振参数选型时, 推荐  $ESR \leq 200 \Omega$ ,  $C_L \leq 16pF$  (晶振的负载电容参数  $\leq 16pF$ ); 对 9~24MHz 晶振参数选型时, 推荐  $ESR \leq 50 \Omega$ ,  $C_L \leq 16pF$  (晶振的负载电容参数  $\leq 16pF$ )。