

文档编号: AN2046

上海东软载波微电子有限公司

应用笔记

ES32F0943

修订历史

版本	修改日期	更改概要
V1.00	2024-09-11	初版发布

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技术有限公司

本资料内容为上海东软载波微电子技术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技术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技术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技术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技术有限公司联系

目录

第 1 章	概述	5
1.1	开发环境	5
1.2	库函数选择	5
1.3	寄存器写保护	5
1.3.1	FC 写保护	5
1.3.2	IWDT 写保护	6
1.3.3	GPIO 写保护	6
1.3.4	CMP 写保护	6
1.4	写 1 清零寄存器	6
第 2 章	系统控制	7
2.1	系统时钟 RCU	7
2.1.1	HOSC 时钟(HOSCCLK).....	7
2.1.2	HRC 时钟(HRCCLK).....	8
2.1.3	PLL 时钟(PLLCLK).....	8
2.1.4	LOSC 时钟(LOSCCLK).....	8
2.1.5	LRC 时钟(LRCCLK).....	8
2.1.6	时钟安全系统(CSS)	8
2.1.7	特殊区块 IP	9
2.1.8	MCO 时钟 (MCOCLK).....	9
2.2	低功耗唤醒流程	9
2.3	FLASH 模块.....	9
2.4	配置字.....	10
2.5	HRC 透过 LOSC 校准方式说明	10
2.5.1	时钟设定.....	10
2.5.2	脚位设定.....	10
2.5.3	校准初始配置	10
2.5.4	MCO 配置.....	10
2.5.5	GP32C4T1 配置.....	10
2.5.6	主程序计算方式.....	11
第 3 章	外设	12
3.1	GPIO 模块	12
3.2	UART 模块.....	12
3.3	I2C 模块.....	13
3.4	SPI 模块.....	13
3.5	RTC 模块	14
3.6	模拟电源	15
3.7	模拟斩波器	15
3.8	CMP 模块	15
3.9	IA 模块	16
3.10	ADC 模块.....	16
3.11	温度感测模块.....	16
3.12	OPAMP 模块.....	16

3.13	LCD 模块	17
第 4 章	系统电路与版图注意事项	19
4.1	最小系统电路	19
4.1.1	LQFP80 封装芯片最小系统电路	19
4.1.2	LQFP64 封装芯片最小系统电路	20
4.2	外围电路	21
4.2.1	外部晶振 HOSC 与 LOSC	21
4.2.2	ESLINK II 与调试接口	21
4.2.3	UART-BOOT 刻录接口	21
4.3	ADC 量测技巧与输出速率	22
4.3.1	ADC 量测技巧	22
4.3.2	配置与输出速率之间的关系	23
4.3.3	ADC 数据建立时间	23
4.4	利用 OPAMP 产生压力传感器所需恒流源	23

第1章 概述

1.1 开发环境

推荐用户使用 Keil5、IAR8.11 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，适合大部分使用者。

MD

基本上只提供寄存器位域级别的“读”、“写”接口，适合对晶元底层较为熟悉的使用者。

如果用户对速度不是要求非常严格，一般情况下推荐使用者使用 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

为了防止程序异常导致系统出现错误，芯片提供了写保护寄存器，用于防止对被保护的寄存器误操作。FC、IWDT、GPIO 等模块都支持寄存器写保护功能，用户在写入被保护寄存器之前需要先解除写保护状态(允许写)，否则无法对该寄存器进行写入。写入操作完成后，再使能写保护(禁止写入)。在库函数中，都提供了相应的宏定义，方便使用者进行解除保护和使能保护的操作。

1.3.1 FC 写保护

闪存控制器 FC 提供 FC_CMD 和 FC_CTL.OPRLD 寄存器，用于闪存的写访问操作，这些操作是受保护的，即用户无法直接编程或擦除闪存，以避免意外修改闪存数据。

如果用户需要对闪存执行编程或擦除，必须先进行解锁流程。

- 解锁流程需要连续输入 2 组解锁密钥，否则解锁失效，寄存器保持上锁状态。
- 解锁成功后，用户可以通过 FC_STA.CMDULK 位来判定是否成功解锁。
- 当此位状态为 1 时，表示成功解锁，可以进行寄存器写访问操作。

为了保护闪存数据的安全，用户在执行闪存编程或擦除操作前，应该严格遵守解锁流程，避免误操作导致数据丢失。库函数中提供了相应的宏定义，方便用户进行解锁和锁定操作。

解锁流程如下所示：

1. 检查 FC_STA.CMDULK 为 0，确认目前处于锁定状态。
2. 将第一组密钥 0x00112233 写入 FC_UL 寄存器。
3. 将第二组密钥 0x55667788 写入 FC_UL 寄存器。
4. 检查 FC_STA.CMDULK 为 1，确认解锁成功。

进行闪存编程与擦除操作后，建议重新启用锁定功能以保护闪存数据不被意外擦除或覆盖。要执

行此操作，需使用 FC_UL 寄存器填入非密钥值(例如 0x00000000)以重新上锁闪存控制器。

1.3.2 IWDT写保护

IWDT 寄存器、IWDT_BKPR、IWDT_BKRLR 和 IWDT_BKWINR 都是受保护的写访问操作。

解锁方式是对 IWDT_BKCR 写入 0x00005555 解锁码，即可解除写保护功能。如果对 IWDT_BKCR 写入非密钥值，则会重新启用寄存器写访问保护机制。

由于 IWDT 寄存器存放在备用寄存器中，因此写入寄存器的访问需等待 33 个 PCLK 时钟进行更新，您可以读取 IWDT_BKFR.BUSY 位以判断寄存器是否已完成更新。

1.3.3 GPIO写保护

透过 GPIOx_LCK 寄存器，可锁定 GPIO 寄存器的写访问功能，包括 GPIOx_MOD、GPIOx_OT、GPIOx_PUD、GPIOx_AFL 和 GPIOx_AFH 寄存器。

锁定的过程需要进行 IO 锁定流程，需对 GPIOx_LCK 寄存器进行 2 次相同数值的写入，每次写入需要 32 位数值，并且 GPIOx_LCK[31:16]必须是 GPIOx_LCK[15:0]取反值。写入完成后，可读取 GPIOx_LCK.LCKK 位来判别锁定流程是否正确开启了 IO 锁定功能，当 GPIO 寄存器被锁定后，无法取消锁定，只能透过系统复位来清除锁定。

1.3.4 CMP 写保护

完成 CMP 配置后，您可以将 CMP_CFG1.LOCK 位设置为 1，这样可以使得 CMP_CFG1 和 CMP_CFG2 寄存器都变成只能进行读取操作，包括 LOCK 位。当 CMP 寄存器被锁定后，无法取消锁定，只能透过系统复位来清除锁定。

1.4 写 1 清零寄存器

中断标志寄存器都采用"写 1 清零(C_W1)"的方式来操作，这意味着只有在写 1 时，对应的位才会被清除为 0。对于"写 1 清零(C_W1)"的寄存器，不建议使用"读-修改-写"的方式进行操作，因为容易导致标志误清，进而导致中断丢失的问题。建议在中断处理程序中，使用专门的写操作来清除相应的标志位，确保中断能够正确触发并被处理。

第2章 系统控制

2.1 系统时钟 RCU

系统时钟(SYSCLK)提供以下来源:

- HRC (High Speed Internal RC Oscillator) - 内部高速 16 MHz RC 振荡器
- HOSC (High Speed External Oscillator) - 外部高速时钟振荡器
- PLL (Phase Locked Loop) - 锁相环
- LRC (Low Speed Internal RC Oscillator) - 内部低速 32 kHz RC 振荡器
- LOSC (Low Speed External Oscillator) - 外部低速 32.768 kHz 时钟振荡器

系统重置后, 默认使用内部 16 MHz 高速 RC 振荡器(HRC)作为系统时钟。透过 RCU_CFG.SW 寄存器, 可切换不同的系统时钟源。在切换系统时钟源之前, 必须确保目标时钟源已经开启并且时钟已经稳定, 才可进行更换系统时钟。

读取 RCU_CFG.SWS 寄存器, 可确认当前系统时钟源是否已更换完成。需要注意的是, 无法透过 RCU_CON 或 RCU_LCON 寄存器来关闭当前的系统时钟源。当误操作时, RCU_CON 或 RCU_LCON 寄存器也不会有任何反应。此外, 当 PLL 选用于系统时钟源时, PLL 所采用的参考时钟(HRC、HOSC、LRC 或 LOSC)也无法透过 RCU_CON 或 RCU_LCON 寄存器关闭。

当该时钟源不使用时, 可以独立设置其开启或关闭, 以优化系统功率消耗。此外, 该系统还提供了除频电路, 依据应用场景与功耗需求, 配置 AHB 与 APB 操作频率。AHB 与 APB 最高可配置为 48MHz。所有周边外设时钟状态都依据所属的总线(Bus)时钟而定。如 AHB 总线时钟为 HCLK, APB 总线时钟为 PCLK。以下列出使用特殊时钟源的区块:

- STCLK 时钟
- RTCCLK 时钟
- IWDTCCLK 时钟
- ADCCLK 时钟
- CHOPCLK 时钟
- VLCDCLK 时钟
- LCDCLK 时钟

2.1.1 HOSC时钟(HOSCCLK)

外部时钟源 (HOSC Bypass)

使用外部有效的时钟源, 需先设定 RCU_CON 缓存器中的 HOSCBYP 和 HOSCON 启用 HOSC Bypass 功能。接着, 外部时钟源必须输入至 HOSCI 引脚。当外部时钟源被启用后, 内部的 HOSC 将停止振荡, 系统将依据外部时钟源提供的频率运作。

外部石英晶体振荡器 (HOSC Crystal)

HOSC 时钟讯号是由外部高速晶体振荡器产生。配置 RCU_CON.HOSCON 位控制 HOSC 开启与关闭; HOSC 时钟状态可藉由 RCU_CON.HOSCRDY 标志位确认, 当 HOSC 时钟稳定时, 标

志位将被硬件自动配置为高电位。

2.1.2 HRC时钟(HRCCLK)

HRC 时钟讯号是由内部高速 16 MHz RC 振荡器产生。透过配置 RCU_CON.HRCON 位，可以控制 HRC 开启或关闭；而 HRC 时钟状态可以透过检查 RCU_CON.HRCRDY 标志位来确认。当 HRC 时钟稳定时，标志位将被硬件自动配置为高电位。

2.1.3 PLL时钟(PLLCLK)

锁相回路(Phase-Locked Loop, PLL)可透过 RCU_CFG.PLLSRC 选择 HRC、HOSC、LRC 或 LOSC 作为参考时钟频率，并且透过 RCU_CFG1.PLLFREQ 选择要将参考时钟进行倍频以输出所需的时钟频率。根据参考时钟频率的来源不同，倍频的大小也会有所差异：

- 若选择 HRC 或 HOSC 作为输入参考时钟频率，可透过设定 RCU_CFG.PREDIV 确保输入频率为 4 MHz，接着透过 RCU_CFG1.PLLFREQ 选择 8 或 12 倍频。例如，当输入参考时钟频率除频至 4 MHz，经过 8 倍频的选择，PLL 输出时钟频率为 32 MHz。
- 若选择 LRC 或 LOSC 作为输入参考时钟频率，则透过 RCU_CFG1.PLLFREQ 选择 1024 或 1536 倍频。例如，当输入参考时钟频率为 32 kHz，经过 1024 倍频的选择，PLL 输出时钟频率为 32.768 MHz。

2.1.4 LOSC时钟(LOSCCLK)

外部时钟源 (LOSC Bypass)

由外部提供有效的时钟源，并设定缓存器 RCU_LCON.LOSCBYP 与 RCU_LCON.LOSCON 启用 LOSC Bypass 功能；外部时钟源必须由 LOSCI 引脚输入。

外部石英晶体振荡器 (LOSC Crystal)

LOSC 时钟是由外部 32.768kHz 石英晶体振荡器或一个稳定低速时钟源提供，可用于实时时钟(RTC)计数时间与日历。配置 RCU_LCON.LOSCON 位控制 LOSC 开启与关闭；LOSC 时钟状态可藉由 RCU_LCON.LOSCRDY 标志位确认，当 LOSC 时钟稳定时标志位将被硬件自动配置为高电位。

2.1.5 LRC时钟(LRCCLK)

LRC 内部低速 RC 振荡器，是一个低功耗时钟源，并可以在低功耗模式下让外设独立看门狗(IWDT)和实时时钟(RTC)持续运行；时钟频率大约 32kHz。配置 RCU_LCON.LRCON 位控制 LRC 开启与关闭。LRC 时钟状态可藉由 RCU_LCON.LRCRDY 标志位确认，当 LRC 时钟稳定时标志位将被硬件自动配置为高电位。

2.1.6 时钟安全系统(CSS)

当系统时钟选择外部石英晶体振荡器时，时钟安全系统(Clock Security System, CSS)是一个重要的系统安全机制。当系统时钟选用 HOSC 或是 PLL 使用 HOSC 倍频时，为了避免 HOSC 发生故障导致系统停机或陷入不安全的状态，可以透过启用或关闭 RCU_CON.HOSCCSSON 来启动或停用 HOSC 时钟安全系统。系统只有在 HOSC 时钟开启且讯号稳定时才能启用时钟安全系统，

并且在侦测到 HOSC 停止时会自动关闭时钟安全系统并切换成 HRC。同样地，当系统时钟选用 LOSC 或是 PLL 使用 LOSC 倍频时，可以选择启用或关闭 RCU_LCON.LOSCCSSON 来启动或停用 LOSC 时钟安全系统。在侦测到 LOSC 停止时，系统会自动关闭时钟安全系统并切换成 LRC。

2.1.7 特殊区块IP

以下这些特殊区块只能支持特定时钟源

Source IP	AHB	APB	HRC	HOSC	PLL	LOSC	LRC
ADC 时钟(ADCCLK)			⊙	⊙	⊙		
Chopper 时钟(CHOPCLK)			⊙	⊙	⊙		
VLCD 时钟(VLCDCLK)			⊙	⊙	⊙		
LCD 时钟(LCDCLK)						⊙	⊙
RTC 时钟(RTCCLK)						⊙	⊙
IWDT 时钟(IWDTCLK)							⊙
System Tick 时钟(STCLK)	⊙						

表 2-1 Clock source

2.1.8 MCO 时钟 (MCOCLK)

微控制器提供时钟输出功能，允许用户将时钟讯号输出到 MCO，不仅可以用于检查，还可当作其他装置或模块的输入时钟源。用户可透过配置 RCU_CFG.MSW 来选择输出时钟，并根据需求设定 RCU_CFG.MPRE 进行除频，可选择 2、4、8、16、32、64 或 128 倍的除频倍率。使用者可以自由选择适合的时钟输出。

2.2 低功耗唤醒流程

注意事项 1: 从低功耗模式唤醒后，必须检查/清除唤醒旗标。

低功耗模式(STOP0/STOP1/STANDBY/SHUTDOWN)中，当经由引脚 WKUPx 唤醒后，必须清除唤醒旗标 SYSCFG_WKSR.WKCLR，并检查唤醒旗标 SYSCFG_WKSR.FLAG 是否已清除。

注意事项 2: 从低功耗模式(STOP0/STOP1)唤醒后，需检查系统时钟来源是否稳定。

低功耗模式(STOP0/STOP1)唤醒后，系统时钟配置保持睡前状态，但为了确保时钟稳定，请务必确认时钟来源的 RDY 旗标；可由 RCU_CFG.SWS 取得当前的时钟源，在判定相对应的稳定旗标 RDY。

2.3 FLASH模块

注意事项 1: FLASH 读保护(RP)配置字

- 配置保护等级为 Lv0 等级，默认不保护。可修改保护等级提升至 Lv1 或 Lv2。在提升保护等级的过程中，程序区内所储存的信息仍会保留。
- 配置保护等级为 Lv1 保护，无法进行程序下载以及数据读出，但可进行调试。可修改保护等级，将保护等级降回到 Lv0 保护，这个过程会触发程序区全清除。将保护等级提升至 Lv2 保护，程序区所储存的信息仍然会保留。
- 配置保护等级为 Lv2 保护，将无法进行程序下载、数据读出以及调试，不可修改保护等级。

注意事项 2: FLASH 等待时钟数(Wait Cycle)

系统频率变化时, 必须确保有足够的 FLASH 等待周期 FC_CTL.WAIT, 支持最多 3 个等待周期, 如下表。

系统时钟	FC_CTL.WAIT
>48Mhz	2
48Mhz ≥ f _{SYSCLK} >24Mhz	1
24Mhz ≥ f _{SYSCLK}	0(无等待)

表 2-2 FLASH 等待时钟数(Wait Cycle)

注意事项 3: 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。

2.4 配置字

注意事项 1: 配置字修改与重置

当使用 ELink II (miin/Pro), 修改芯片配置字之后, 需要断电后重新上电, 才能正常调试。

注意事项 2: BOR 配置字

BOR 电压等级(Level), 必须设定高于 MCU 工作电压; 电压等级数值, 请参考 ES32F0943_Datasheet.pdf

注意事项 3: IWDG 配置字

IWDG 配置字开启后, 软件无法关闭, 预设 1 秒发生 IWDG 复位, 支持软件修改复位时间。

2.5 HRC透过LOSC校准方式说明

2.5.1 时钟设定

- ◆ RCU 始能 LOSC、GP32C4T1、GPIOA

2.5.2 脚位设定

- ◆ 设定 PA00 为 MCO

2.5.3 校准初始配置

- ◆ 设定 HRC TRIM 值为 0x80
- ◆ 设定 HRC 校准值选择为使用 HRCTRIM 校准值

2.5.4 MCO配置

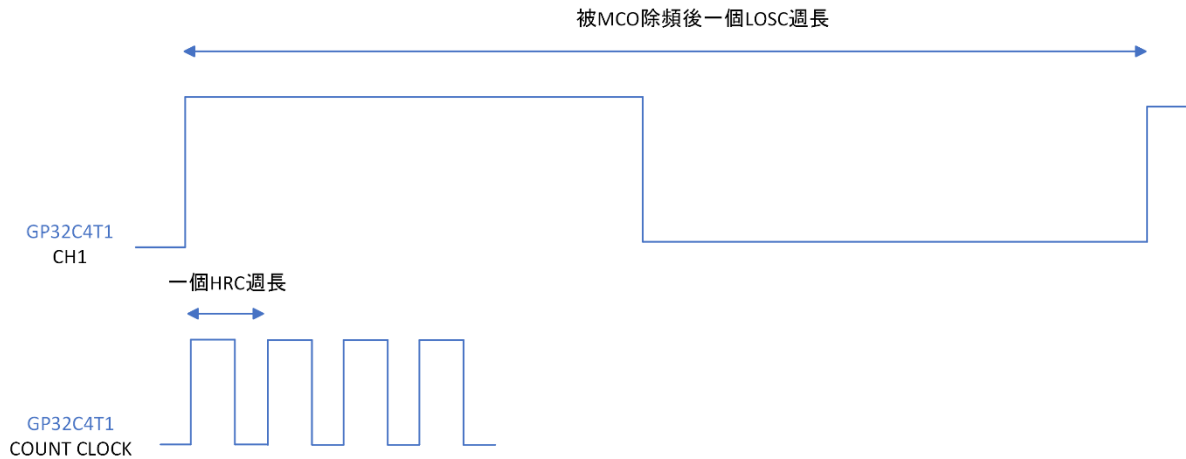
- ◆ 设定 MCO 来源为 LOSC 及设定 MCO 除频为 128

2.5.5 GP32C4T1 配置

- ◆ 设定 GP32C4T1 的除频为 0, 设定 GP32C4T1 的计数值为 0
- ◆ 设定 GP32C4T1 计数方向为上数

- ◆ 设定 GP32C4T1 的 channel1 输入捕捉为 MCO
- ◆ 设定 GP32C4T1 为输入模式，捕获源为 I1
- ◆ 设定 GP32C4T1 输入极性为上升沿
- ◆ 始能 GP32C4T1 Channl1
- ◆ 设定 GP32C4T1 为复位模式
- ◆ 设定 GP32C4T1 的触发选择为 I1 滤波后信号

2.5.6 主程序计算方式



程序逻辑:

如果 HRC 为 16M 且 LOSC 为 32.768kHz 的话，去 sample 被 MCO 除频 128 后一个 LOSC 的周期会得到 62500 个 count

如果 sample 完一个 LOSC 得到的 count 值小于 62500 代表目前的 HRC 频率小于 16M Trim 直往上加，反之 count 值大于 62500 代表 HRC 频率目前大于 16M 往下减

计算公式:

LOSC 频率=32.768kHz

MCO 除频倍数=128

所以可以先得到上图 CH1 的频率=(LOSC 频率/128)=256Hz

那么假设 COUNT CLOCK 为准的 HRC16MHz 的话

CH1 周长=count*HRC 周长

count=CH1 周长/HRC 周长=HRC 频率/CH1 频率=16000000/256=62500

第3章 外设

3.1 GPIO 模块

注意事项 1: 未使用的 GPIO 端口处理

在系统中未使用的 GPIO 引脚，建议设置 GPIOx_MOD 为模拟模式(11b)并悬空，以避免漏电情形。

如果需要设置为输入模式，需设置 GPIOx_PUD，上拉电阻接到电源或加下拉电阻接到地。

注意事项 2: GPIO 端口复用流程

当需要设置 IO 复用功能时，需要先配置 GPIOx_AFx 寄存器，选择 AFx 复用种类，再将 GPIOx_MOD 配置为复用功能模式(10b)。

注意事项 3: 低功耗模式唤醒引脚

当系统进入低功耗模式(STOP0/STOP1/STANDBY/SHUTDOWN)时，PB00 至 PB07 可以用作唤醒引脚。在设置 IO 复用功能时，需选择 AFx 复用种类为 WKUPx。此外，需开启唤醒功能 SYSCFG_WKUP.WKEN，并选择唤醒事件 SYSCFG_WKUP.WKEG，可以是上升沿或下降沿模式。

注意事项 4: 低功耗模式引脚维持现状

进入低功耗模式(STANDBY/SHUTDOWN)时，PB00 至 PB07 的引脚状态可以维持进入低功耗模式之前的设定状态，例如 GPIOx_MOD 输入或输出模式、GPIOx_PUD 上拉或下拉、GPIOx_OT 推挽或开漏和 GPIOx_DS 输出驱动。需要注意的是，引脚仅维持设定状态，而不是维持寄存器中的数值。

3.2 UART 模块

注意事项 1: 自动波特率

建议在启用自动波特率侦测功能(UART_MCON.ABREN)时，同时启用重复侦测自动波特率(UART_MCON.ABRREPT)功能，以提高侦测的准确性和稳定性。若发生侦测波特率超时(UART_RIF.ABTO)的情况，波特率开关(UART_MCON.ABREN)会自动清除，此时使用者需重新启用自动波特率侦测功能。另外，如果自动波特率侦测成功但接收到的数据与预期不符，也需要重新启用自动波特率侦测功能(UART_MCON.ABREN)。

注意事项 2: 发送空中断处理方式

UART 的 TX 数据传输缓存包括 UART_TXDATA 寄存器和 TX 位移寄存器。预设情况下，UART_TXDATA 为空，当空中断发生时，如果数据写入 UART_TXDATA，数据会立即转移到 TX 位移寄存器。此时 UART_TXDATA 仍然是空的，再次发生空中断。默认空中断和第一笔数据的空中断之间的时间太短，因此在中断处理流程中，优先清除中断再填写 UART_TXDATA，以便更好地控制数据传输。

3.3 I2C模块

注意事项 1: 当总线 SDA 遇到异常下拉的波形时，可能出现传输状态错误

发生条件为，当 I2C 配置为主机，I2C_CON2.START 写 1 时，从机输出 SDA 低电平，I2C 可能会进入从机接收模式；当从机输出不受控的 SDA 低电平时，请系统流程中加入软件超时机制，在 I2C_CON2.START 写后开始计时，若 I2C 通讯超时，则复位 I2C。复位 I2C 的方式为 I2C_CON1.PE 写 0，重新配置 I2C，然后 I2C_CON1.PE 写 1

注意事项 2: 最大传输 NBYTE

当重载模式关闭时，I2C 支持最大 $2^{16} - 1 = 65535$ Bytes 的传输数量，分别配置 I2C_CON1.NBYTES 与 I2C_CON2.NBYTES 寄存器。

当重载模式开启时，最大传输 NBYTE 为：

- 传送模式下，NBYTE 最大设定值为 65535，当传输数量需求大于 65535 时，在传输前配置 I2C_CON2.RELOAD 位，当第一笔 65535 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 讯号。
- 接收模式下，NBYTE 最大设定值为 255，当传输数量需求大于 255 时，在传输前配置 I2C_CON2.RELOAD 位，当第一笔 255 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 讯号。

注意事项 3: I2C 控制寄存器 2 (I2Cx_CON2)访问限制

I2C 控制寄存器 2 (I2Cx_CON2)，必须按字(32 位)访问。

注意事项 4: 发送空中断处理方式

I2C TX 数据传输缓存，包含 I2C_TXDATA 寄存器与 TX 位移寄存器；预设 I2C_TXDATA 为空，空中断发生，当写数据进入 I2C_TXDATA 后，数据立即被转移到 TX 位移寄存器，此时 I2C_TXDATA 为空，再次发生空中断。默认空中断与第一笔数据的空中断，过于紧凑，因此中断处理流程中，优先清除中断再填写 I2C_TXDATA。

3.4 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，务必优先清除中断再填写 SPI_DATA。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 (字节)，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, 不可在填入数据。

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 接收字节数量, 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)。

3.5 RTC 模块

注意事项 1: 设定日期与开启流程

在开启 RTC 计数之前, 需要先以 BCD 格式设定日期(RTC_CAL)与时间(RTC_TIME)。接着, 可以透过 RTC_CTRL.CKSEL 选择时钟来源为 LOSC 或 LRC, 并设置除频系数 RTC_CTRL.PSCALE 与 RTC_CTRL.SCALE 来产生准确的 1 秒(1 MHz)计数讯号。值得注意的是, 寄存器默认值为 SCALE=0xFF, PSCALE=0x7F, 在使用外部 32.768kHz 晶振时, 频率会是 $32768/(PSCALE*SCALE) = 1 \text{ Hz}$ 。当基本设定完成后, 才可开启 RTC (RTC_CTRL.RTCEN)。此时需要关闭 RTC, 才能够修改时间与日期寄存器。

注意事项 2: 读取日期与时间

由于 RTC 属于备份寄存器区, 系统读取时间或日期前, 可以检查 RTC_STA.SYNDONE 讯号为 1, 确保日期与时间已更新。

注意事项 3: 校准与时钟补偿

当使用非准确的 32.768 kHz 晶振作为 RTC 的时钟来源时, 为了确保 RTC 信号能产生准确的 1 秒, 必须使用 RTC 频率补偿方法。在进行频率补偿前, 需先选择合适的补偿数值, 其计算方式有两种:

- 方法 1: 直接纪录 RTC 计时数, 不开启 RTC 补偿功能, 之后再与标准时间的误差值(ppm)进行比较。
- 方法 2: 使用 RTC 1Hz 讯号触发定时器(Timer)计数方式, 推算出 RTC 1Hz 的误差值(ppm)。尽管此方法受到系统时钟精度的限制, 但相较于方法 1, 它可以更快地取得误差值。举例而言, 若系统频率为 8 MHz, 利用 RTC 1Hz 讯号触发定时器计数, 定时器每秒所计数的数值会接近 8000000, 因此定时器计数数值与理想值(8000000)相减, 即可推算出误差值(ppm)。

在获得误差值(ppm)数据后，需要将该值乘上 32768，再除以最小精度 1/4096，从而得出 RTC 的校准数值。此后，需根据误差值的正负值，设定 RTC_CALIB.MODE 选择正校准或负校准模式，然后将该数值填入 RTC_CALIB.CALIB 中。最后，开启硬件自动校准功能 RTC_CALIB.CALIBEN，以确保 RTC 的信号产生时维持准确

3.6 模拟电源

模拟稳压电源 VDDA，主要提供模拟/数字转换器(SD-ADC)、仪表放大器(IA)、参考电压(VREF)与轨道轨运算放大器(OPAMP)使用。VDDA 可透过 PWR_CTRL.VLDO_MODE 选择四种模式，由 VDDA 引脚输入电源电压。

注意事项 1: 电压开关流程

应用上如需开启模拟电路，电源须按照以下配置流程。首先，开启电流开关，需设置寄存器 PWR_CTRL.BISA_EN，并等待稳定时间 10 us。接着，通过设定 PWR_CTRL.VLDO_MODE 选择 VDDA 的电压模式。当选择内部稳电路提供电源时，信道 PWR_CTRL.VLDO_SEL 决定 VDDA 的电压数值。最后，等待 VDDA 的稳定时间 200 us。

注意事项 2: 参考电压(VREF)配置

在启用参考电压之前，必须先将 PA06 脚位的复用功能设置为 AF8。参考电压源可以选择使用外部电源、电阻分压电源(VDIV)或带隙电压(VBG)。如果不选择外部电源作为参考电压源，则需要 在 PA06 脚位接上电容并接地。

3.7 模拟斩波器

注意事项 1: 模拟电路的斩波器 (Chopper) 时钟 (CHOPCLK) 配置

当需要启用模拟电源或电路中斩波器功能时，必须透过 RCU_CFG2.CHOP_CKCFG 设定输入时钟的来源为 HOSC、HRC 或 PLL，并透过 RCU_CFG2.CHOP_CKDIV 设定输入时钟的除频比例，以提供斩波器所需的 4 MHz 频率。若需要更改时钟来源或除频配置，必须先将 RCU_CFG2.CHOP_CKCFG 设置为 0，再进行修改，以确保输入频率的准确性。

注：在模拟电路中，斩波器的时钟来源都使用相同的 RCU 寄存器配置。

3.8 CMP 模块

注意事项 1: 比较器(CMP)数字滤波器功能配置

可调性低通滤波器可以使用弹跳消除取样(Debounce)功能，透过时钟频率的除频和取样比较器的输出，达到防止尖峰脉冲干扰的效果。开启滤波器功能，方法是设置 CMP_CFG1.FILT，并且透过 CMP_CFG2.FILT_CNT 和 CMP_CFG2.FILT_PRE 的设置调整滤波器的参数。比较器的系统时钟时间为 T_{CMP} 。以下是弹跳消除取样的公式：

$$\text{Debounce Time} = \{(\text{FILT_PRE}+1) \times \text{FILT_CNT}\} \times T_{\text{CMP}}$$

注意事项 2: 比较器(CMP)实现低电压侦测

为了实现低电压侦测,需将正端输入缓存器 CMP_CFG1.INPSEL 设置为 VREFINT,并将负端输入缓存器 CMP_CFG.INNSEL 设置为多节点电组分压输入,同时在缓存器 CMP_CFG2.PS 中选择电阻分压的电压来源为 VDDH。根据使用者设定的分压值 CFG2.DA,比较器将判定当 VDDH 经过分压后是否小于 VREFINT,若是,比较器将转换为低电位。

用户可以透过读取缓存器 CMP_CFG1.VALUE 的数值,或根据数据表中的引脚复用功能,启用 CMP_OUT 的脚位输出,以判断低电压的触发情况。

3.9 IA 模块

注意事项 1: IA 共模电压(VCM)

IA 的共模电压(VCM)为输入讯号的参考电位。使用者需要透过 IA_CTRL.IA_VCMS 寄存器设定共模电压的值,并且需要与 ADC 选择的共模电压一致,以避免共模电压不匹配导致的测量误差。

3.10 ADC 模块

注意事项 1: ADC 时钟(ADCCLK)配置

透过 RCU_CFG2.ADC_CKCFG 设定输入时钟的来源为 HOSC、HRC 或 PLL,并透过 RCU_CFG2.ADC_CKDIV 设定输入时钟的除频比例,以提供 ADC 所需的 4 MHz 频率。若需要更改时钟来源或除频配置,必须先将 RCU_CFG2.ADC_CKCFG 设置为 0,再进行修改,以确保输入频率的准确性。

注意事项 2: ADC 共模电压(VCM)

ADC 的共模电压(VCM)为输入讯号的参考电位。使用者需要透过 ADC_CTRL0.ADC_VCMS 寄存器设定共模电压的值,并且需要与仪表放大器(IA)选择的共模电压一致,以避免共模电压不匹配导致的测量误差。

3.11 温度感测模块

注意事项 1: 温度感测配置

温度传感器输出电压 TSP 以及 TSN 有输入至 IA 以及 ADC 的通道。

3.12 OPAMP 模块

注意事项 1: 运算放大器(OPAMP)数字滤波器功能配置

可调性低通滤波器可以使用弹跳消除取样(Debounce)功能,透过时钟频率的除频和取样运算放大器的输出,达到防止尖峰脉冲干扰的效果。开启滤波器功能,方法是设置 OPA_CTRL.OPA_FILTEN,并且透过 OPA_DBC.DBCNT 和 OPA_DBC.DBPRE 的设置调整滤波器的参数。比较器的系统时钟时间为 T_{OPAMP} 。以下是弹跳消除取样的公式:

$$\text{Debounce Time} = \{(DBPRE+1) \times DBCNT\} \times T_{OPAMP}$$

3.13 LCD 模块

注意事项 1: 电荷泵时钟(VLCDCLK)配置

透过 `RCU_CFG2.VLCD_CKCFG` 设定输入时钟的来源为 `HOSC`、`HRC` 以及 `PLL`，并透过 `RCU_CFG2.VLCD_CKDIV` 设定输入时钟的除频比例，以提供电荷泵时钟所需的 `2 MHz` 频率。若需要更改时钟来源或除频配置，必须先将 `RCU_CFG2.VLCD_CKCFG` 设置为 `0`，再进行修改，以确保输入频率的准确性。

注意事项 2: LCD 时钟(LCDCLK) 配置

LCD 时钟透过 `RCU_CFG2.LCD_CKCFG` 选择时钟源为 `LRC` 或 `LOSC`，并且可透过 `RCU_CFG2.LCD_CKDIV` 设置最低分频为 `2`，最高分频为 `8`。若需要更改时钟来源或除频配置，必须先将 `RCU_CFG2.LCD_CKCFG` 设置为 `0`，再进行修改，以确保输入频率的准确性。

注意事项 3: LCD 帧幅时钟(LCD_PHY_CLK) 配置

LCD 的帧幅时钟是由 `LCD_CLK` 提供，可以透过设定 `LCD_CTRL.FRDIV` 位，来选择适当的帧幅分频器来输出 LCD 帧幅频率。要确保 LCD 的帧幅频率和操作波型能够正确显示，必须根据外接的 LCD 显示器规格，设定正确的频率

注意事项 4: LCD 模式控制(LCD_MODE)寄存器限制

若要修改 LCD 模式控制寄存器的设定，必须先确认 `LCD_CTRL.LCD_EN` 位为 `0`，也就是 LCD 控制器未被启用的状态，才能进行修改。这样可以避免在操作过程中对 LCD 显示器造成损坏或不良影响。

注意事项 5: 占空比(Duty)与 COM/MIX/SEG 的关系

在 LCD 显示器中，占空比是指像素的亮度占整个周期的比例。硬件透过 `LCD_MODE.DUTY` 的设置，来决定 `COM/MIX/SEG` 的配置和占空比的大小。`LCD_MODE.DUTY = 0`，占空比为 `1/4`，`COM` 的数量为 `4` 个，`MIX` 将配置为 `SEG0/SEG1`，LCD 最多可配置成 `4x38`。`LCD_MODE.DUTY = 1`，占空比为 `1/6`，`COM` 的数量为 `6` 个，`MIX` 将配置为 `COM5/COM6`，LCD 最多可配置成 `6x36`。

注意事项 6: IO 复用配置与 LCD 输出开关

IO 脚位作为 LCD 控制信号的输出或是 LCD 电压时，必须透过设置 `GPIOx_AFL` 或是 `GPIOx_AFH` 将对应脚位的复用功能设为 `8`，才能开启 `COMx/MIXx/SEGx` 通道对应的脚位输出。

注意事项 7: LCD 电压(VLCD)配置与流程

LCD 可以透过设置 `LCD_MODE.VLCD_SRC` 选择电压的来源，包括内部产生和外部输入。内部可调电荷泵可以透过缓存器 `LCD_MODE.VLCD_SEL` 选择电压范围 `2.5V-4.0V`，外部电压则需要保证在允许的电压范围内(详见 `datasheet`)，同时需要设置 `PB11` 脚位的复用功能为 `AF8`。在进行 LCD 电压配置时，需要注意电压稳定性，并且必须按照以下顺序进行配置，以确保正常运作。如果不选择外部电源作为 VLCD 电压源，则需要在 `PB11` 脚位接上电容并接地。

- 若使用内部可调电荷泵产生电压，则需要先设定 VLCD 时钟为 `2 MHz`，接着设定 `LCD_MODE.VLCD_SRC` 选择内部可调电荷泵，再将脚位 `PB11` 设定为 `AF8` 复用功能。接着需要等待连接在 VLCD 脚位的电容充电，如果使用的外部电容为 `1uF`，则需要等待约 `2ms`。

最后设定 LCD_MODE.VLCD_SEL 选择电压文件位，等其他 LCD 控制讯号设定完成后，才能设定 LCD_CTRL.LCDEN 开启 LCD。

- 若使用外部电压来源，则需要先设定 LCD_MODE.VLCD_SRC 选择外部电压源，接着将脚位 PB11 设定为 AF8 复用功能。最后，等其他 LCD 控制讯号设定完成后，才能设定 LCD_CTRL.LCDEN 开启 LCD。

注意事项 8: GPIO 模式

GPIO 模式可以适用于 LED 的应用。启用 COMx/MIXx/SEGx 脚位作为数字信号输出，必须将 LCD_MODE.LCD_GPIO 设置为 1，同时也需要设置 GPIOx_AFL 或 GPIOx_AFH，将对应脚位的复用功能设置为 8。如果需要提供数字输出的反转功能，可以透过设置 LCD_MODE.POLA_SEL, POLA_SEL[0]和 POLA_SEL[1]分别控制 COMx/SEGx 的极性选择。

第4章 系统电路与版图注意事项

4.1 最小系统电路

4.1.1 LQFP80 封装芯片最小系统电路

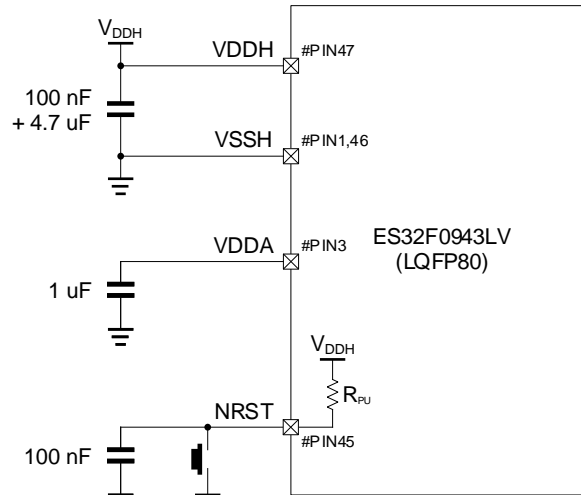


图 4-1 LQFP80 封装芯片最小系统电路

注:

1. 芯片电源请务必依照图中建议进行配置，才能保证芯片电源在任何应用场景下都能够稳定。
2. 每一组电源必须连接陶瓷耦合电容(如图式)；这些电容必须尽可能靠近芯片的相应管脚，才能保证芯片的运行性能。
3. NRST 引脚采用 RC 复位，支持内部上拉(上拉电阻值，请参考数据手册电器性章节)，电容采用 100nF。

4.1.2 LQFP64 封装芯片最小系统电路

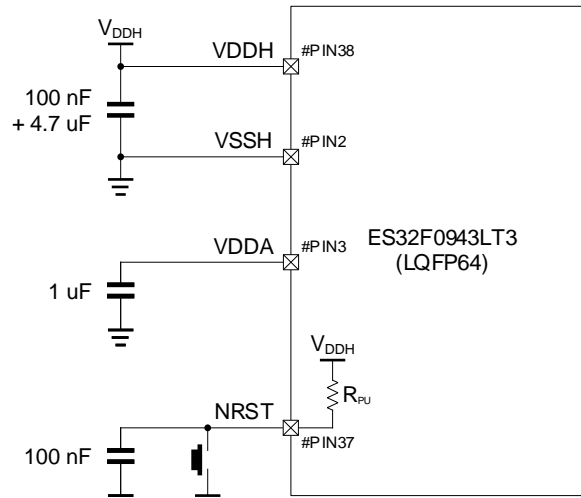


图 4-2 LQFP64 封装芯片最小系统电路

注:

1. 芯片电源请务必依照图中建议进行配置，才能保证芯片电源在任何应用场景下都能够稳定。
2. 每一组电源必须连接陶瓷耦合电容(如图式)；这些电容必须尽可能靠近芯片的相应管脚，才能保证芯片的运行性能。
3. NRST 引脚采用 RC 复位，支持内部上拉(上拉电阻值，请参考数据手册电器性章节)，电容采用 100nF。

4.2 外围电路

4.2.1 外部晶振HOSC与LOSC

注意事项 1: 外部陶瓷电容

电容建议数值，请参阅 ES32F0943 数据手册，电器特性章节。

注意事项 2: 版图参考原则

外部晶振应尽量靠近芯片，提高稳定度。

4.2.2 ESLINK II与调试接口

注意事项 1: 电路建议

PB09 与 PB08 引脚预设为 ESLINK II 与调试接口，PB09 为 SWCLK 默认为输入下拉模式，PB08 为 SWDIO 默认为输入上拉模式。当系统应用时，将此接口配置为其他 IO 功能时，如需维持 ESLINK II 与调试接口的功能，务必预留 NRST 引脚功能，才能在 NRST=0 时，恢复 SWCLK 与 SWDIO 功能。

注意事项 2: Socket 进行刻录时，芯片电源方案务必符合图 4-1 与图 4-2 的电源方案要求。

4.2.3 UART-BOOT刻录接口

注意事项 1: 开机程序默认选择

系统开机流程中，提供 BootRom 与 Main Flash 两种选择，而 ES32F0943 预设为 BootRom；此配置可透过修改配置字更换默认值。

注意事项 2: UART-BOOT 接口选用

当配置字 BOOT BYPASS 选项为 BootRom，表示系统开机从 BootRom 开始，PB00 与 PB01 引脚预设为 UART1 通信接口，PB00 为 UART1_TX 输出，PB01 为 UART1_RX 输入；由 UART 通信接口，搭配 UART-BOOT 刻录工具下载或更新程序。

当配置字 BOOT BYPASS 选项为 Main Flash，表示系统开机直接转跳至 Main 区执行，PB00 与 PB01 引脚预设为 GPIO，不具备 UART-BOOT 刻录功能。

4.3 ADC量测技巧与输出速率

4.3.1 ADC量测技巧

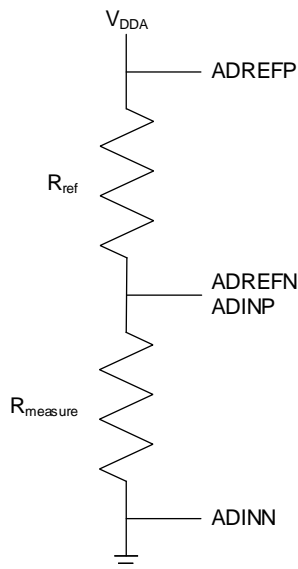


图 4-3 量测方式

可以透过串接电阻方式，透过分压电阻，提供 ADC 参考电压。这样当电压源抖动时，因为 V_{ref} 与 $V_{measure}$ 同时抖动，不会影响量测精度。

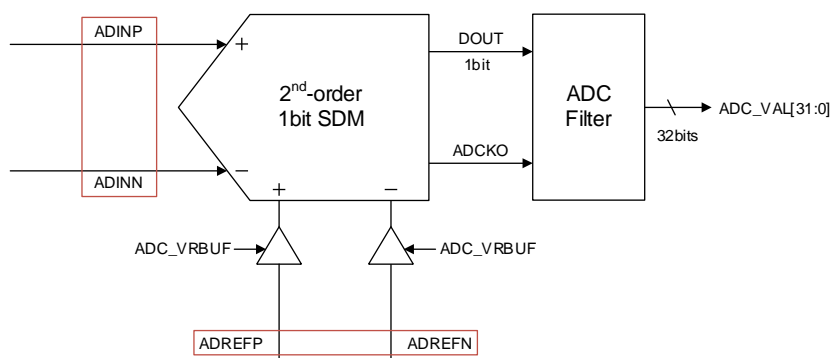


图 4-4 ADC 连接方式

ADC 参考电压可以配置 `ADC_CTRL0.ADC_VRRS=1` (请参考 RM), 选择 0.5x 参考电压, 相当于 ADC 放大倍率再增加两倍, ES32F0943 最大放大倍率可达 $(\text{ADC reference gain } 2) \times (\text{IA gain } 64) \times (\text{ADC gain } 8) = 1024$ 。

4.3.2 配置与输出速率之间的关系

- OSR 配置低, ADC 输出值稳定度相对较差, 但输出速率快。
- OSR 配置高, ADC 输出值愈稳定, 输出速率降低。
- 依用户产品应用场景做相对应适当的配置。
- 单笔转换时间 = OSR / ADC_CKO
- 数据建立时间 = $(order * OSR) / ADC_CKO$
- 斩波器(取平均) 建立时间 $(2 * order * OSR) / ADC_CKO$

4.3.3 ADC数据建立时间

当复位 ADC 时, ADC 需要一段数据建立时间才会稳定输出结果。建议复位 ADC 后等待建立时间后再开启以便 CIC Filter 皆使用已稳定的数值运算。

下列为不同条件下需要的稳定时间

- ADCKO = 1 MHz, Settle Time = 20 us
- ADCKO = 500 kHz, Settle Time = 40 us
- ADCKO = 250 kHz, Settle Time = 80 us

4.4 利用OPAMP 产生压力传感器所需恒流源

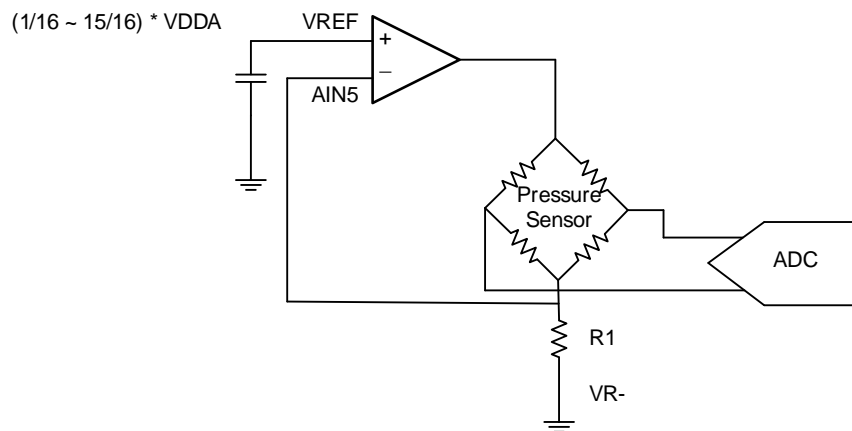


图 4-5 恒流源使用方式

- 定电流公式: $I = VREF / R1$