

瑞萨 RA 系列

RA0 快速设计指南

简介

本文档回答了常见的问题，并指出了 RA0 MCU 的一些细节之处，除非通篇阅读硬件手册，否则可能会忽略这些细节。本文档并不适合取代硬件手册，而是对手册的一种补充，重点介绍大多数工程师在开始自己的设计时需要的一些关键项目。本文档还从应用的角度探讨了一些设计决策。

目标 MCU

RA0 MCU 系列

目录

1	电源.....	4
1.1	参考文档.....	4
2	仿真器支持.....	5
2.1	SWD 接口.....	5
3	MCU 运行模式.....	6
4	选项设置区.....	7
4.1	选项设置区寄存器.....	8
5	时钟电路.....	9
5.1	复位条件.....	10
5.2	时钟频率要求.....	10
5.2.1	对 ROM 或数据闪存进行编程和擦除的要求.....	10
5.3	降低时钟生成电路（CGC）功耗.....	10
5.4	写入系统时钟控制寄存器.....	11
5.5	时钟设置示例.....	12
5.6	HOCO 精度.....	12
5.7	闪存接口时钟.....	12
5.8	电路板设计.....	12
5.9	外部晶振选择.....	13
5.10	外部时钟输入.....	14
6	复位要求和复位电路.....	15
6.1	引脚复位.....	15
6.2	上电复位.....	15
6.3	独立的看门狗定时器复位.....	15
6.4	电压监控复位.....	16
6.5	软件复位.....	16
6.6	其他复位.....	16
6.7	冷/热启动的测定.....	16
6.8	确定复位源.....	17
7	存储器.....	18
7.1	SRAM.....	19
7.2	外设 I/O 寄存器.....	19

7.3	片上闪存.....	19
7.3.1	后台运行.....	20
7.3.2	ID 代码保护.....	20
7.4	片上闪存.....	21
8	寄存器写保护.....	22
9	I/O 端口配置.....	23
9.1	多功能引脚选择设计方案.....	23
9.2	设置端口用作 GPIO.....	23
9.2.1	内部上拉.....	24
9.2.2	开漏输出.....	24
9.3	设置和使用端口外设功能.....	24
9.4	设置和使用 IRQ 引脚.....	25
9.5	未使用的引脚.....	26
9.6	不存在的引脚.....	27
9.7	电气特性.....	27
10	模块 STOP 功能.....	28
11	中断控制单元.....	29
12	低功耗.....	31
13	外部总线.....	35
13.1	总线错误监测.....	35
13.1.1	总线错误类型.....	35
13.1.2	总线错误发生时的运行情况.....	36
14	通用布线建议.....	37
14.1	数字域与模拟域.....	37
14.2	高速信号设计注意事项.....	37
14.3	信号组选择.....	38
15	参考文献.....	39

1 电源

RA0 MCU 有数字电源和模拟电源。电源使用了以下引脚。

表 1. 数字电源

符号	功能名称	描述
VCC	电源	电源引脚。连接到系统电源。通过 VCC 引脚附近的 0.1 μF 电容，将该引脚连接到 VSS。
VSS	地	接地。
VCL	电源	通过一个 0.47 μF ~ 1.0 μF 的电容将该引脚连接到 VSS。将此电容放置在靠近 VCL 引脚的位置。

表 2. 模拟电源

符号	功能名称	描述
VREFH0	12 位 ADC 高基准电压	12 位 A/D（不作为通用输入输出 GPIO 时）的基准电压输入引脚。
VREFL0	12 位 ADC 低基准电压	12 位 A/D（不作为通用输入输出 GPIO 时）的模拟基准接地引脚。

1.1 参考文档

有关 RA0 MCU 产品电源方面的详细内容，请参阅以下文档：

- R01UH1040 RA0E1 Group, *RA0E1 Group User's Manual: Hardware*

“**Overview**”一章列出了各种封装 MCU 的电源引脚以及推荐的旁路电容。

“**Resets**”一章叙述了上电复位以及如何将其与其他复位源区分开来。

“**Low Voltage Detection**”一章详细介绍了用于监控电压的低电压检测电路。“**Option-Setting Memory**”一章还介绍了如何在启动时自动启用电压检测电路。

如果使用的是片上模数转换器（ADC），请参阅相应硬件手册“**12-Bit A/D Converter (ADC12)**”的章节了解详细信息。

表 3. RA0 系列硬件手册

章节名称	描述
Overview	列出了各种封装产品中的电源引脚，并附有有关终端和旁路的说明。
Resets	叙述了上电复位以及如何将其与其他复位源区分开来。
Low Voltage Detection	详细介绍了用于监控电源的低电压检测电路。
Low Power Modes	使用低功耗模式可以降低 MCU 的功耗。有关运行模式如何影响 MCU 中各种功能模块的电源要求的详细信息，请参阅本章。
12-bit A/D Converter	如果计划使用片上 A/D 转换器，可以参考这些章节在如何为这些外设提供滤波电源方面的详细说明。
Clock Generation Circuit	详细说明了如何配置以及如何使用可用时钟，包括 PCB 设计建议。

2 仿真器支持

RA0 MCU 的仿真器接口支持使用 SWD 通信进行调试。

SWD 仿真器接口应连接到 Arm® 标准 10 针或 20 针插座。

SWD 信号必须有上拉电阻才符合 Arm® 规范。如果没有正确的上拉电阻，接口可能无法正常运行。而 RA0 MCU 具有内部上拉电阻，且默认是在上述信号端口上使能的。当内部上拉电阻被使能时，这些信号便不再需要外部电阻。

仿真器支持对于产品开发和原型设计非常有用，但在设计进入生产阶段后可能就不需要了。如果设计不再需要仿真器支持，请根据 MCU 硬件手册的 I/O 端口一章中的“*Handling of Unused Pins*”部分配置端口。另请参阅本文档第 9.5 章节中的内容。

2.1 SWD 接口

下图显示了使用串行线调试（SWD）时调试接口的典型连接方式。

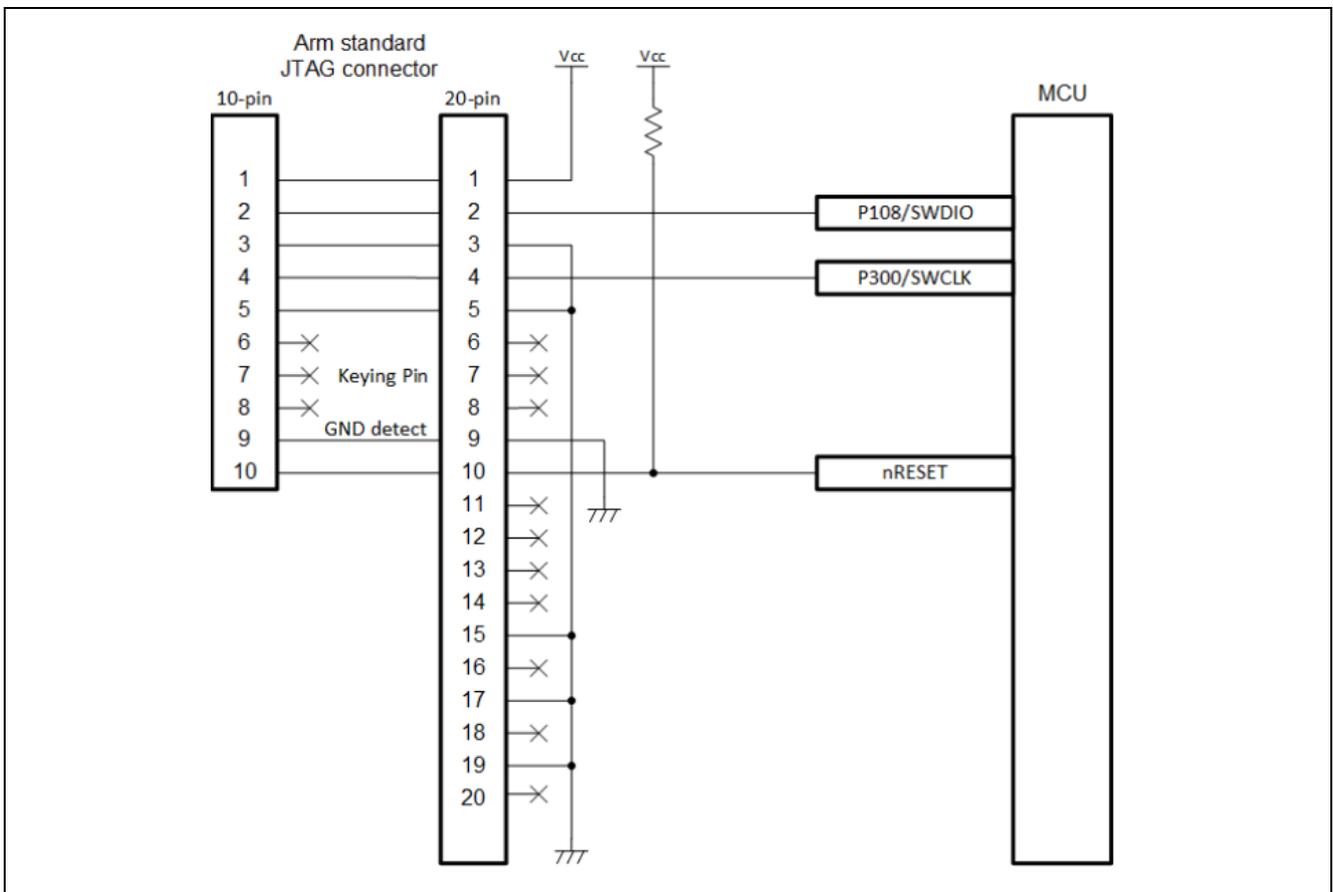


图 1. SWD 接口连接

注：用户系统的复位电路的输出必须是集电极开路。

3 MCU 运行模式

RA0 MCU 复位后进入单芯片模式。RES 引脚复位和 POR 复位时会发生此种情况。

图 2 为运行模式转换图。

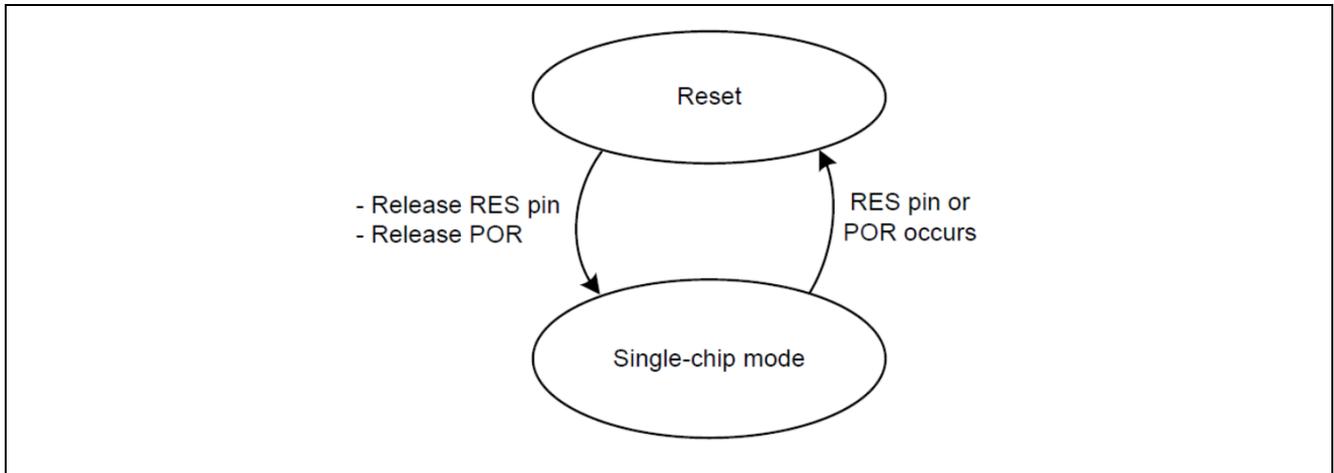


图 2. 运行模式转换

需注意：RA0 MCU 没有 MD (Mode) 引脚，这一点与瑞萨其他的 RA 产品不同。RA0 MCU 不支持 SCI Boot 模式，因此无需 MD 引脚。

4 选项设置区

选项设置区决定了 MCU 复位后的状态。它被分配在闪存的配置设置区域和程序闪存区域。这两个区域的设置方法不同。

寄存器在硬件手册的“Option Setting Memory”一章中有详细介绍。

选项设置内存寄存器占用代码闪存映射中的空间。虽然寄存器位于 RA MCU 预留的闪存中，但**一些客户可能会无意中**将数据存储到这些位置。建议用户进行检查以确保没有将不需要的数据写入这些位置，否则可能导致芯片发生故障。此外，当使用二进制文件编程时，用户必须确保内存的保留区域不会因配置设置区的地址而被编程。例如，通过设置闪存选项寄存器可以在复位后立即启用独立看门狗定时器（IWDT）。如果程序 ROM 中存储的数据无意中与选项设置区寄存器重叠，则会在不知情的情况下开启 IWDT。这可能会导致调试器与主板出现通信问题。

下图显示了选项设置区，该设置区由 RA0E1 产品的选项功能选择寄存器组成。各产品的选项设置区器可能有所不同。有关具体产品的详细信息，请查阅 MCU 用户手册。

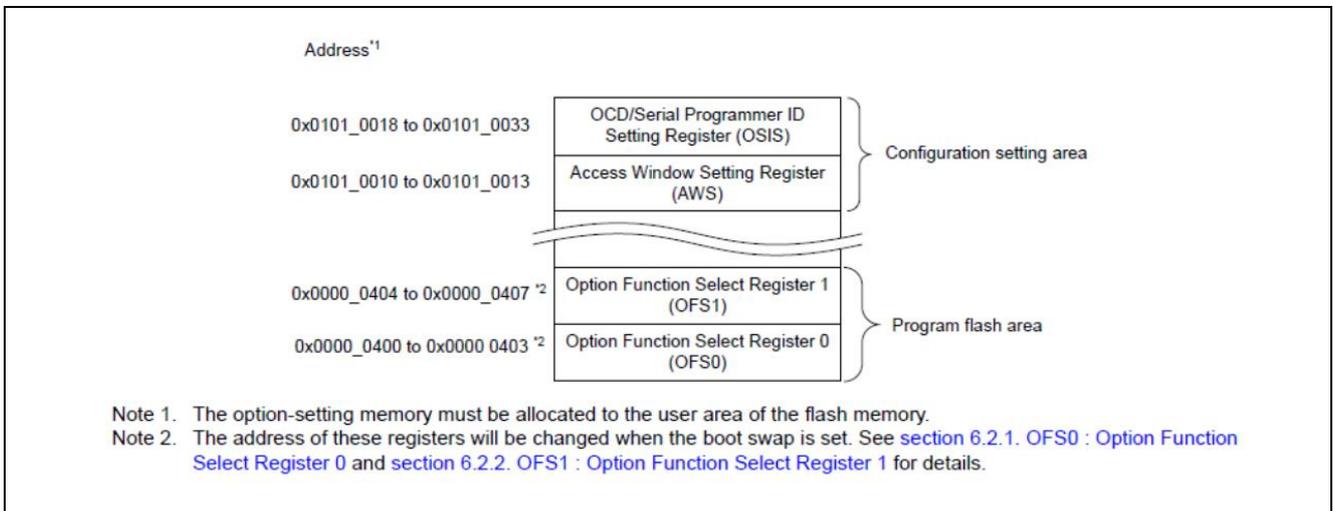


图 3. RA0E1 的选项设置区

4.1 选项设置区寄存器

以下是选项设置区寄存器的概要。首次对 MCU 进行启动编程之前，请确保它们已正确配置。

- OFS0 寄存器
 - 独立看门狗定时器 (IWDT) 自动启动
 - IWDT 超时、频率、窗口、中断类型和低功耗模式
- OFS1 寄存器
 - 复位后的 LVDAS 启动设置 (LVD0 电路启动)
 - 复位后的 HOCO 启动设置
 - P206 / RES 功能选择
 - 复位后的闪存读保护

瑞萨 FSP 配置器支持在 BSP 设置中对选项设置区进行设置，如图所示（以 RA0E1 为例）。

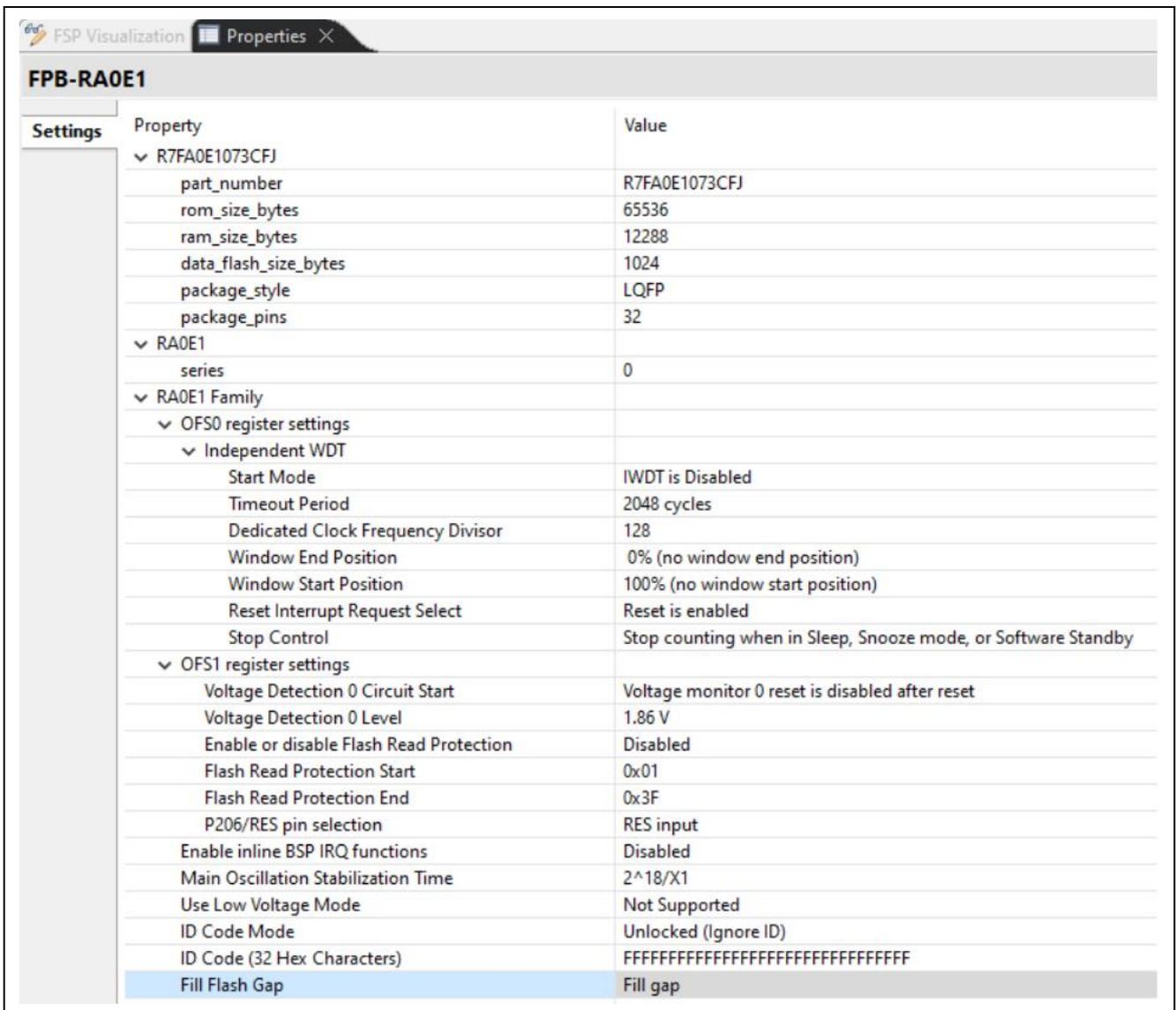


图 4. 在 FSP 中设置 RA0E1 的选项设置区

5 时钟电路

RA0 MCU 有 5 个主振荡器，均可用作主系统时钟源。典型系统中，主时钟由外部晶振或时钟驱动。它直接被作为内部选择器和分频器的输入，再被输入到主系统时钟（ICLK）、闪存时钟（FCK）、CPU 时钟以及外设模块时钟。

注：某些 RA0 产品可能会复用主时钟振荡器和副时钟振荡器的输入引脚。关于时钟生成电路框图，请参阅硬件手册“Clock Generation Circuit”一章。

每个时钟都有特定的容差和定时值。有关频率和时钟时序规格，请参阅硬件手册“Electrical Characteristics”一章中“AC Characteristics”的内容。有关各种时钟频率之间的关系，请参阅硬件手册“Clock Generation Circuit”一章。

表 4. RA0 振荡器规格

振荡器	输入源	频率	主要用途
主时钟 (MOSC)	外部晶体/谐振器	1 MHz 至 20 MHz	系统时钟 (ICLK) : CPU、Flash、SRAM、Flash-IF、外设模块时钟
	外部时钟	最高 20 MHz	
副时钟 (SOSC)	外部晶振/谐振器	32.768 kHz	实时时钟、低功耗模式下的系统时钟、PCLBUZ0、TAU 时钟、UARTA 时钟、TML32 时钟。
高速片上时钟 (HOCO)	片上振荡器	24/32 MHz	启动时的系统时钟、CPU、Flash、SRAM、Flash-IF、外设模块时钟
中速片上时钟 (MOCO)	片上振荡器	1 MHz 至 4 MHz	系统时钟 (ICLK)、外设时钟、PCLBUZ0、TAU 时钟、TML32 时钟、UARTA 时钟。
低速片上时钟 (LOCO)	片上振荡器	32.768 kHz	低功耗模式下的系统时钟、外设时钟、Systick 定时器、TAU 时钟、TML32 时钟、UARTA 时钟、PCLBUZ0、实时时钟、独立看门狗定时器时钟 ^{注1} 。

注 1：如果启用了 IWDT，不管 LOCO 寄存器 (LOCOCR) 如何设置，LOCO 都会振荡。

有的 RA0 产品可能会多路复用主时钟振荡器和副时钟振荡器输入引脚。在这种情况下，用户需仔细评测 MCU 硬件用户手册中振荡器特性部分所述的片上振荡器容差。

5.1 复位条件

复位后，RA0 MCU 以高速片上振荡器（HOCO）作为主时钟源运行。复位时，主振荡器默认停止，且引脚设为端口模式。复位后，SOSC、LOCO 和 MOCO 默认停止。IWDTC 可以开启或停止，具体取决于选项设置区的设置（参见第 4 节）。

5.2 时钟频率要求

最小频率和最大频率如下表所示。硬件手册中“Clock Generation Circuit”一章的“Overview”部分详细介绍了内外时钟源的规格。有关每个时钟输入源的容差的更多详细信息，请参阅硬件手册中“Electrical Characteristics”一章的“Oscillator Characteristics”的内容。

表 5. RA0 MCU 内部时钟的频率范围

	ICLK ^{注1}	RTCCLK	IWDTCCLK	SWCLK	CLKOUT	SYSTICCLK
最大频率 [Hz]	32M	32768	16384	12.5M	16M	32768
最小频率 [Hz]	—	128	16384	—	—	32768

注 1: RA0 MCU 对内部 ROM 或数据闪存进行编程或擦除时，ICLK 的频率必须至少为 1 MHz。

5.2.1 对 ROM 或数据闪存进行编程和擦除的要求

FCLK 必须至少为 1 MHz，才能在内部 ROM 和数据闪存上执行编程和擦除。

5.3 降低时钟生成电路（CGC）功耗

为了省电，请将任何未使用的时钟的分频器尽可能设置为更大的值。对于未使用的时钟，则通过设置适当的寄存器来确保它已停止。用于控制各时钟源的寄存器如下表所示。

表 6. 时钟源配置寄存器

振荡器	寄存器	描述
主时钟	MOSCCR	启动/停止主时钟振荡器
副时钟	SOSCCR	启动/停止副时钟振荡器
高速片上时钟（HOCO）	HOCOCCR	启动/停止 HOCO
中速片上时钟（MOCO）	MOCOCCR	启动/停止 MOCO
低速片上时钟（LOCO）	LOCOCCR	启动/停止 LOCO

时钟源启动后，在将其用作内部时钟源之前需要一个等待期。可通过 CGC 外设模块中的寄存器（如 HOCOSF 和 OSTSB）读取和控制等待期的状态。

所有内部时钟都可以输出到时钟/蜂鸣器输出引脚。这样就可以精确测量时钟频率。

使用特殊功能寄存器 HIOTRM、MIOTRM 和 LIOTRM 可以微调所有内部片上振荡器源的频率。硬件手册中“Electrical Characteristics”一章的“On-chip Oscillators Characteristics”章节中标明了微调寄存器的时钟校正分辨率以及其他适用参数。

5.4 写入系统时钟控制寄存器

写入时钟控制寄存器的各个位时应小心。请遵循硬件手册中“System Clock (ICKL)”章节内的寄存器列表、ICKL 源图和时钟源切换时序图进行操作。

当切换系统时钟（ICKL）的时钟源时，在时钟源转换期间，系统时钟周期的时间会变长。如图 5 所示。

为确保时钟频率改变后能够进行正确的处理，应首先写入相关时钟控制寄存器以改变频率，然后读取寄存器中的值，最后执行后续处理。

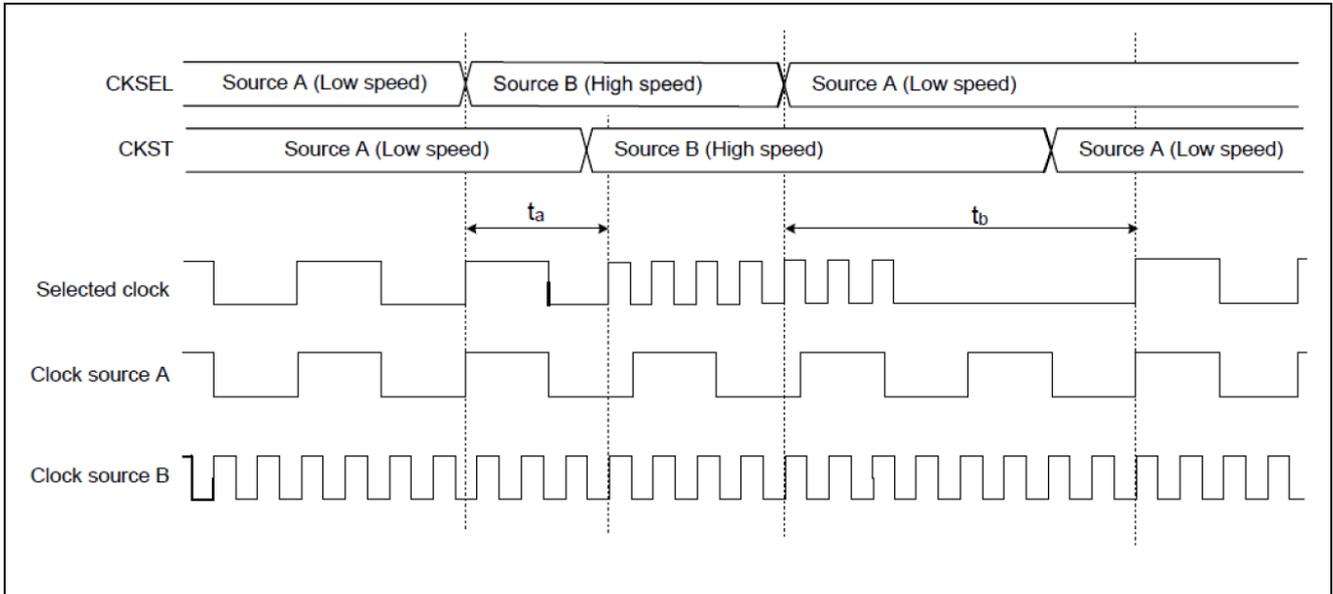


图 5. 时钟源切换时序

5.5 时钟设置示例

瑞萨 FSP 为 RA0 MCU 提供了简单、可视化的时钟配置工具。下图以 RA0E1 MCU 为例，该配置器可配置板卡支持包中的代码，便于根据用户的选择初始化时钟发生电路，并采用 MCU 硬件手册中列出的预防措施。

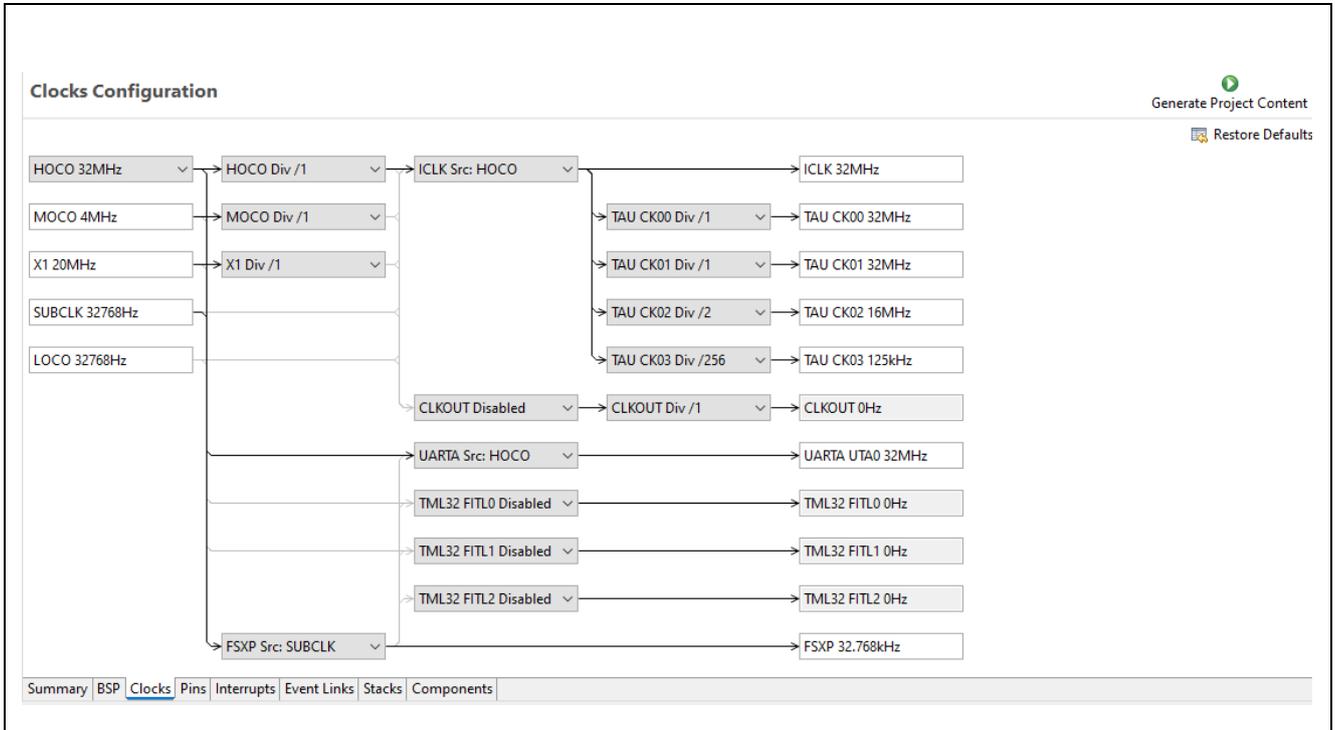


图 6. 使用 Renesas FSP 配置器进行时钟设置

5.6 HOCO 精度

RA0 MCU 内部高速片上振荡器（HOCO）的运行频率为 24MHz 或 32MHz。当 HOCO 时钟振荡稳定标志位 (HOCOSF) 为 1 时，HOCO 的典型精度将达到 ±1% 或更高。有关时钟精度规格，请参阅硬件手册中的“Electrical Characteristics”。

HOCO 可用作时钟发生电路的输入。当以这种方式使用 HOCO 时，不需要外部振荡器。如因空间限制或其他限制需要减少 PCB 设计中的元器件数量时，这便是一个优势。当然，要同时考虑时钟精度、性能权衡方面的限制，需要针对客户的应用进行评估。

5.7 闪存接口时钟

RA0 的系统时钟（ICLK）用作对内部闪存进行编程和擦除时的运行时钟。

因此频率设置将直接影响读取数据闪存所需的时间。如果用户的程序是从数据闪存读取数据，或在内部闪存上执行编程或擦除，则建议使用最大 ICLK 频率。

在烧写或擦除 ROM 或数据闪存时，系统时钟频率必须至少为 1 MHz。请注意，时钟频率对读 ROM 或读写 RAM 不产生任何影响。

5.8 电路板设计

有关使用 CGC 的更多信息和电路板设计建议，请参阅硬件手册中“Clock Generation Circuit (CGC)”章节的“Usage Notes”部分。通常而言，晶振及其负载电容应尽可能靠近 MCU 时钟引脚（X1/X2、

XCIN/XCOUT)。应避免在晶振和 MCU 之间出现任何其他信号。尽量减少每条走线上使用的连接通孔数量。

5.9 外部晶振选择

外部晶振可用作大部分 RA0 MCU 的主时钟源。外部晶振连接在 MCU 的 X1 和 X2 引脚上。其频率必须在主时钟振荡器的频率范围内。

晶振的选择很大程度上取决于具体的电路板设计。由于可用的晶振种类繁多，为 RA0 MCU 选用晶振时请仔细评估电气特性，以确定具体要求。

典型的晶振连接示例如下所示。

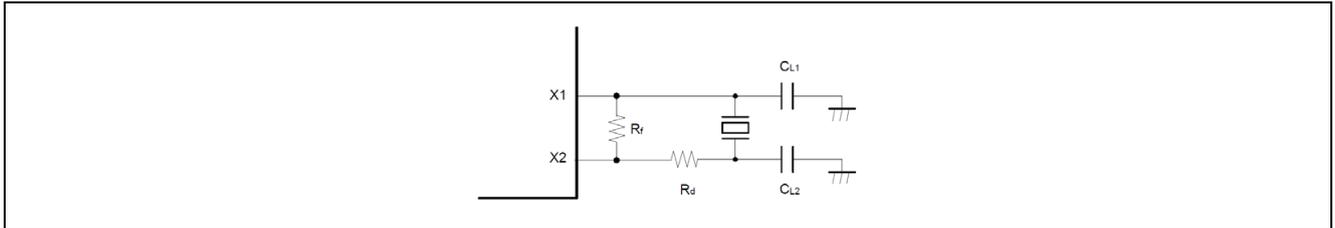


图 7. 晶振连接示例

在选择晶振和相关电容时，须进行详细评估。在有晶振厂家推荐的前提下，可添加外部反馈电阻（ R_f ）和阻尼电阻（ R_d ）。

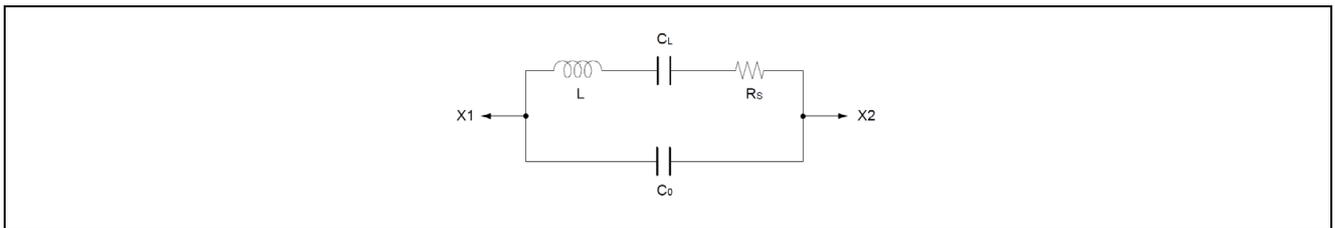


图 8. 晶振的等效电路

$CL1$ 和 $CL2$ 的电容值选择会影响内部时钟的精度。使用上图中晶振的等效电路对电路进行仿真，可了解 $CL1$ 和 $CL2$ 值的影响。如需获得更准确的结果，还要考虑与晶振元件之间的布线相关的杂散电容。

“RA Family Design Guide for Sub-Clock Circuits”详细介绍了外部晶振和副时钟振荡器的电路板设计规范。

5.10 外部时钟输入

数字时钟输入可用作大多数 RA0 MCU 的主时钟源。图 9 为连接外部时钟输入的示例。使用外部时钟信号作为时钟输入时，需将 CMC:MOSEL[1:0] 位设为“11b”。X1 引脚变为高阻抗。

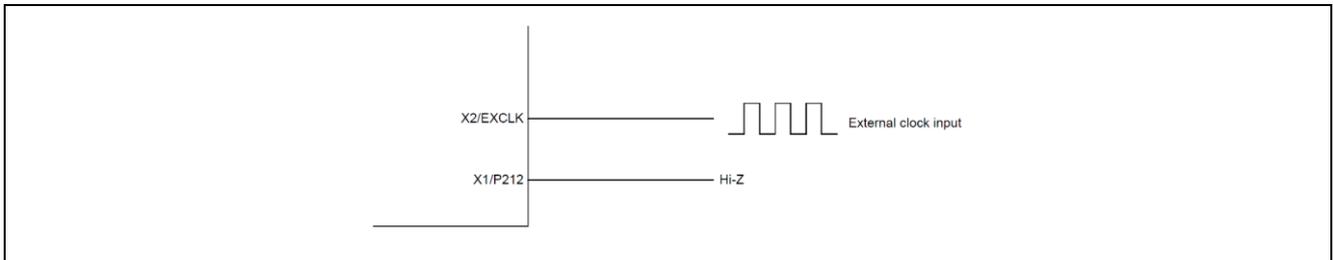


图 9. 外部时钟的等效电路

注：只有在主时钟振荡器停止时，才能更改外部时钟输入的频率。当主时钟振荡器停止位（MOSCCR.MOSTP）设为“0”时，请勿更改外部时钟输入的频率。

6 复位请求和复位电路

基于 Arm® Cortex®-M23 的 RA0 产品共有 7 种复位。

表 7. RA0 MCU 的复位

复位名称	源
引脚复位	RES#引脚被拉低
上电复位	VCC 上升（电压检测：VPOR）
独立看门狗定时器复位	独立看门狗定时器下溢，或发生刷新
电压监视器 0 复位	VCC 下降（电压检测：Vdet0）
电压监视器 1 复位	VCC 上升/下降（电压检测：Vdet1）
SRAM 奇偶校验错误复位	SRAM 奇偶校验错误检测
软件复位	寄存器设置

6.1 引脚复位

当 RES#引脚被拉低时，所有处理均中止，MCU 进入复位状态。要在 MCU 运行时复位，RES#应保持为低电平，以达到指定的复位脉冲宽度。有关更详细的时序要求，请参阅硬件手册“Electrical Characteristics”一章的“Reset Timing”部分。另请参阅本文档的第 2 节“仿真器支持”，了解有关与调试支持相关的复位电路的详细信息。

接通电源后 P206 被用作 RES 输入。选项选择寄存器 1 (OFS1) 的 PORTSELB 位定义了该端口是用作 P206 还是 RES。当该引脚设置为 P206 时，在上电复位（POR）电路复位期间以及从 POR 电路复位释放到开始正常运行期间，不要向该引脚输入低电平。如果在此期间持续输入低电平，芯片将响应外部复位而保持在复位状态。片内上拉电阻在电源接通后启用。

6.2 上电复位

有两种情况会产生上电复位（POR）：

1. 通电期间 RES#引脚处于高电平状态。
2. VCC 低于 V_{POR} 的时候，RES#引脚处于高电平状态。

当 VCC 超过上电复位电压（ V_{POR} ），且上电复位时间（ t_{POR} ）过后，芯片退出上电复位状态。上电复位时间是允许外部电源和 MCU 稳定的时间段。请参阅硬件手册的“Electrical Characteristics”一章的“POR and PVD Characteristics”，了解电压电平和时序方面的详细信息。

由于 POR 电路要满足 RES#与 VCC 同时为高电平，因此当仅使用上电复位功能时，不要在复位引脚上加电容。否则，相对于 VCC，RES#的上升时间将被减缓，从而导致 POR 电路无法正确识别上电条件。

当电源（VCC）降至或低于 V_{POR} 时，如果 RES# 引脚为高电平，则会产生上电复位。当 VCC 上升到 V_{POR} 以上并且持续 t_{POR} 时间后，芯片退出上电复位状态。

上电复位后，RSTSR0 的 PORF 位被置为“0”。为正确检测开机复位，用户需在开机复位前将该位设置为“1”。

6.3 独立的看门狗定时器复位

该复位是由独立看门狗定时器（IWDT）生成的内部复位。

当 IWDT 下溢时，可以选择生成一个独立的看门狗定时器复位（也可以改为生成 NMI），并且 IWDT 状态寄存器 IWDTSR 中的下溢位 UNDFE 被设置为“1”。经历短暂延迟后，IWDT 复位被取消。具体时序请参阅硬件手册。

6.4 电压监控复位

RA0 产品具有可使 MCU 在掉电期间避免不安全操作的电路。板载比较器依据多个基准电压（如 Vdet0、Vdet1）检查电源电压。当电源电压降至每个基准电压以下时，可能会产生中断或复位。检测电压 Vdet0 可从 OFS1 寄存器的 6 个不同级别中进行选择。检测电压 Vdet1 可从 LVD1CR 寄存器的 18 个不同的级别中进行选择。

之后当 VCC 升至 Vdet0 或 Vdet1 以上时，在稳定时间过后，解除电压监控复位。

上电复位后，低电压检测将被禁用。对于 Vdet0，可通过选项功能选择寄存器 (OFS1) 启用电压监测功能；对于 Vdet1，可通过 LVD1CR 寄存器启用电压监测功能。有关详细信息，请参阅硬件手册中“Low Voltage Detection (LVD)”一章。

LVD 复位后，RESF 的 LVIRF 位被置为“1”。

6.5 软件复位

该复位是通过将“1”写入 Arm 内核的 AIRCR 寄存器中的 SYSRESETREQ 位而生成的内部复位。生成软件复位后，当软件复位产生的内部复位时间 (t_{RESW2}) 过去时，内部复位被取消，CPU 开始复位异常处理。具体时序请参阅硬件手册。

有关 SYSRESETREQ 位的详细信息，请参阅 Arm® Cortex®-M23 Technical Reference Manuals。

6.6 其他复位

MCU 的大多数外设功能都可以在特定故障条件下产生复位。启用这些复位无需硬件配置。请参阅硬件手册中的相关章节，了解关于各个外设功能生成复位的条件的详细信息。

6.7 冷/热启动的测定

RA0 MCU 可由用户确定复位处理的起因。PORSR.PORF 标志用来显示是上电复位导致的复位（冷启动）还是操作期间输入的复位信号导致的复位（热启动）。

当上电复位发生时，该标志被置为“0”。否则，该标志不会置为“0”。当通过软件将“1”写入该标志时，该标志设置为“1”。即便写入“0”，它也不会被置为“0”。

6.8 确定复位源

RA0 MCU 可由用户确定复位信号生成源。读取 RESF 寄存器以确定哪个复位是复位源。有关根据中断源修改哪些标志，请参阅硬件手册的“Reset detect flags initialized by each reset source”内容。以下代码示例显示如何使用瑞萨 FSP 中基于 CMSIS 的寄存器结构体来判断复位到底是由软件复位还是上电复位引起的。

```
/* Power on Reset */  
  
if(1 == R_SYSTEM->PORSR_b.PORF)  
{  
  
/* Do something */  
  
}  
  
...  
  
/* Software Reset */  
  
if(1 == R_SYSTEM->RESF_b.SWRF)  
{  
  
/* Do something */  
  
}
```

7 存储器

RA0 MCU 支持 4 GB 线性地址空间，范围从 0x0000_0000 到 0xFFFF_FFFF，包含程序和数据。程序和数据存储器共享地址空间。它们使用单独的总线来访问每个总线，从而提高性能并允许在同一周期访问程序和数据。地址空间内包含片上 RAM、外设 I/O 寄存器、程序代码闪存和数据闪存区域。在执行过程中，所有保留区域都是禁止访问区域。如果访问保留区域，可能会出现未定义的行为。

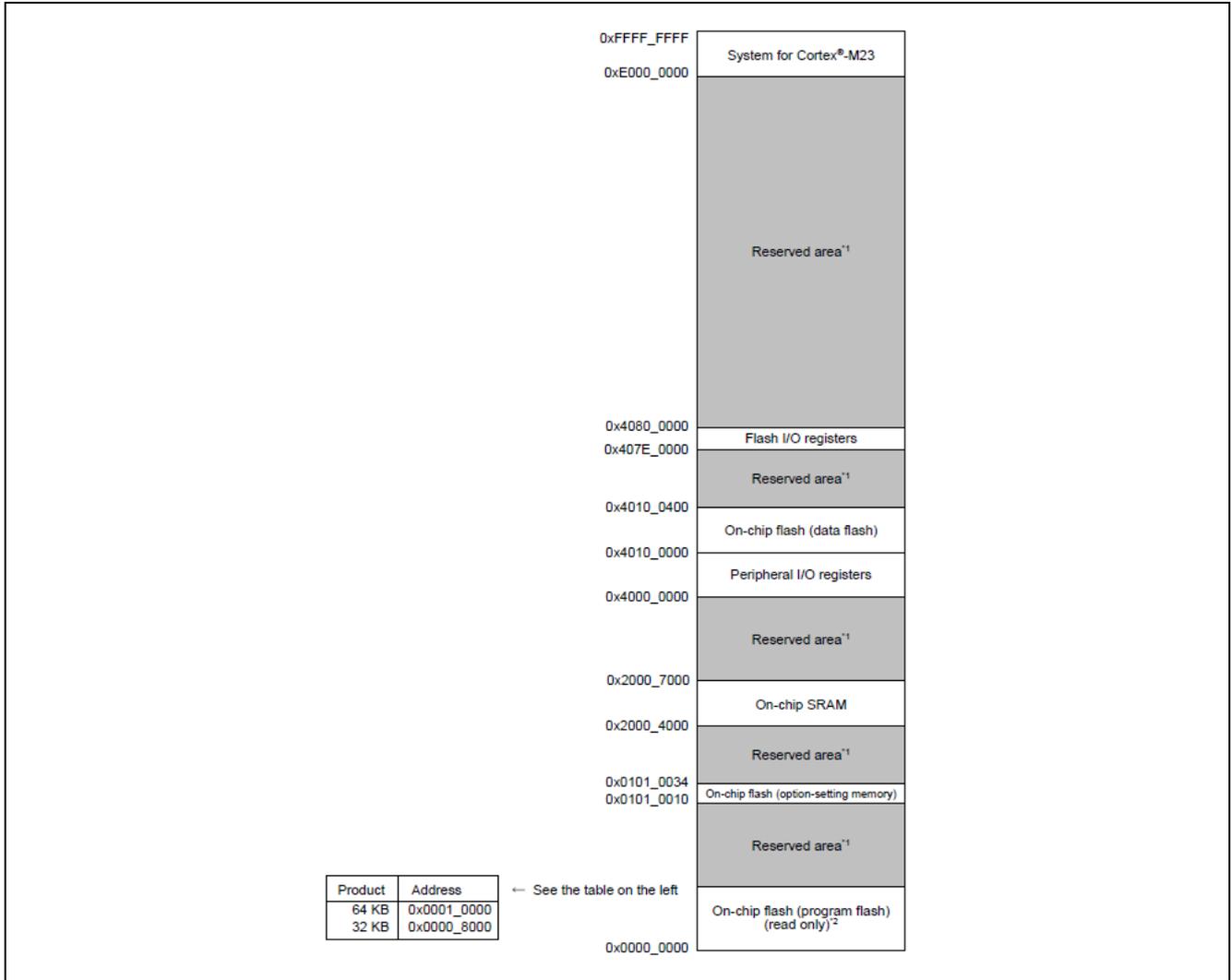


图 10. RA0E1 存储器映射图

7.1 SRAM

RA0 MCU 提供片上 SRAM 模块。下表列出了 SRAM 规格。SRAM 模块容量因产品而异。有关详细信息，请参阅硬件手册。

参数	说明
SRAM容量	SRAM0: 12KB
SRAM地址	SRAM0: 0x2000 4000 ~ 0x2000 6FFF
访问	读写均无需等待
奇偶	奇偶校验，每8位数据位配备1位偶校验
错误检测	偶校验错误检测

图 11. RA0E1 SRAM 规格示例

7.2 外设 I/O 寄存器

外设 I/O 寄存器块出现在存储器映射中的不同位置，具体取决于器件和当前的运行模式。大多数外设 I/O 寄存器占用的地址范围为 0x4000 0000h 到 0x400F FFFFh。但是，位置和大小可能会因产品而异。有关详细信息，请参见硬件手册。有关详细信息，可参见“[I/O Registers](#)”附录，也可参见每个外设功能的寄存器说明。该区域包含在所有运行模式下始终可用的寄存器。用于控制访问闪存的闪存 I/O 寄存器占据两个区域，即 407E 0000h 至 407F FFFFh。

瑞萨 FSP 包含采用 CMSIS 数据结构的 C 语言头文件，这些文件将特定器件的所有外设 I/O 寄存器映射到易于访问的 I/O 数据结构体。

7.3 片上闪存

RA0 MCU 具有两个闪存部分：代码闪存和数据闪存，各部分的大小和可编程周期因产品而异。闪存控制块 (FCBU) 控制闪存的编程和擦除。闪存应用程序命令接口 (FACI) 根据指定的 FACI 命令控制 FCU。

代码闪存旨在存储用户应用程序代码和常量数据。数据闪存旨在存储可能不时更新的信息，例如配置参数、用户设置或记录的数据。数据闪存区域中的编程和擦除单位小于代码闪存的单位。例如，RA0E1 的代码闪存的编程单位为 32 位，擦除单位为 2 KB，而数据闪存的编程单位为 8 位，擦除单位为 256B。单位大小因设备而异。有关详情，请参阅硬件手册中的“[Flash Memory](#)”一章。

数据闪存和代码闪存区域均可通过应用程序代码进行编程或擦除。这种模式称为自编程。这样，无需连接外部编程工具即可完成现场固件更新。自编程时，必须向闪存提供稳定的 HOCO 时钟，以产生编程电压和擦除电压。因此如果在选择其他时钟源作为系统时钟时停止了 HOCO，则必须在执行自编程之前启动 HOCO 运行并确保振荡处于稳定状态。

瑞萨 FSP 为代码闪存和数据闪存提供了 HAL 层驱动程序。

图 12 给出了代码闪存和数据闪存的规格。

项目	描述	
内存容量	<ul style="list-style-type: none"> 用户区域, 64-KB/32-KB 配置设定区域 (见硬件手册“6. Section-Setting Memory”) 	用户区域, 14-KB
读取周期	<ul style="list-style-type: none"> 读取操作需要2周期 	读取操作需要2周期
擦除后的值	0xFF	0xFF
编程/擦除方法	<ul style="list-style-type: none"> 通过寄存器中指定的FCB命令来实现代码和数据闪存的编程和擦除。 通过用户程序进行闪存编程 (自编程)。 	
安全功能	防止闪存数据被非法篡改或读取	
保护	防止闪存被错误重写	
后台运行 (BGO)	在对数据闪存进行编程或擦除时, 可以读取代码闪存	
编程和擦除的单位	<ul style="list-style-type: none"> 用户区域的编程单位, 32位单位。 用户区域的擦除单位, 2-KB单位。 	<ul style="list-style-type: none"> 用户区域的编程单位, 8位单位。 用户区域的擦除单位, 256字节单位。
其它功能	在自编程期间可以接受中断。 在该 MCU 的初始设置中, 可以设置选项设置存储器的区域。	
在板编程	在板调试模式下的编程: <ul style="list-style-type: none"> SWD 接口。 无需专门硬件。 在用户程序中通过代码和数据闪存编程例程进行编程: <ul style="list-style-type: none"> 允许在不重置系统的情况下对代码和数据闪存进行编程。 	
注1: HOCO需稳定振荡。见硬件手册“28.9 Self-programming”		

图 12. RA0 MCU 上的代码闪存和数据闪存规格

7.3.1 后台运行

RA0 MCU 代码闪存和数据闪存支持后台运行 (BGO)。这就意味着, 当对闪存编程或者擦除时, 用户可以继续执行和访问正在操作的内存区域以外的内存区域。例如, 当数据闪存被擦除或编程时, CPU 可以从代码闪存执行应用程序代码。

7.3.2 ID 代码保护

RA0 MCU 的选项设置区有一个 128 位存储器, 用作 ID 代码。如果 ID 代码留空 (0xFF), 则不启用保护, 允许使用片上调试器访问 MCU。如果设置了 ID 代码, 用户就可以控制对这些模式的访问。用户可以选择始终禁止连接, 也可以选择输入匹配的 ID 代码时允许连接。有关更多信息, 请参阅硬件手册中 “OCD/Serial Programmer ID Setting Register (OSIS)” 以及 “ID Code Protection” 章节。

瑞萨 FSP 为 RA0E1 提供了 ID 代码保护的选项设置, 如图 13 所示。

FPB-RA0E1		
Settings	Property	Value
	▼ R7FA0E1073CFJ	
	part_number	R7FA0E1073CFJ
	rom_size_bytes	65536
	ram_size_bytes	12288
	data_flash_size_bytes	1024
	package_style	LQFP
	package_pins	32
	▼ RA0E1	
	series	0
	▼ RA0E1 Family	
	> OFS0 register settings	
	> OFS1 register settings	
	Enable inline BSP IRQ functions	Disabled
	Main Oscillation Stabilization Time	2^18/X1
	Use Low Voltage Mode	Not Supported
	ID Code Mode	Unlocked (Ignore ID)
	ID Code (32 Hex Characters)	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
	Fill Flash Gap	Fill gap

图 13. 在瑞萨 FSP 配置器中设置 ID 代码保护

注：为 ID 代码设置保护时，必须谨慎处理，以避免发生可能无法访问 MCU 区域的错误。

7.4 片上闪存

要在 Arm® Cortex®-M 内核上执行代码，内存空间必须为小端模式。

8 寄存器写保护

寄存器写保护功能可防止重要寄存器因软件错误而被覆盖。保护寄存器（PRCR）用于设置需要受保护的寄存器。表 8 列出了 PRCR 位与要保护的寄存器之间的关联。

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0

图 14. PRCR 寄存器

表 8. PRCR 保护位

PRCR 寄存器的位	描述
PRC0	与时钟发生电路相关的寄存器： CMC、SOMRG、FOCOSCR、FMAINSR、FSUBSCR、ICLKSCR、 MOSCCR、SOSCCR、LOCOCR、HOCOCCR、MOCOCCR、OSTS、HOCODIV、 MOCODIV、MOSCDIV、OSMC、LIOTRM、MIOTRM
PRC1	与低功耗模式相关的寄存器： SBYCR、PSMCR、SYOCD CR
PRC3	与 LVD 相关的寄存器： LVD1CR、LVD1MKR、LVD1SR
PRKEY[7:0]	这些位控制着 PRCR 寄存器的写入访问。要修改 PRCR 寄存器，需一次性写入 16 位，即高 8 位写入 A5h，低 8 位写入要写的值。

注：并非全部 RA0 产品包含所有寄存器。有关详细信息，请参阅硬件手册中“Register Write Protection”章节。

瑞萨 FSP 有两个 API（R_BSP_RegisterProtectEnable 和 R_BSP_RegisterProtectDisable）分别用于启用和禁用寄存器写保护。

9 I/O 端口配置

硬件手册中“**I/O Ports**”章节描述了基于外设选择和其他寄存器设置的引脚配置。以下列举了一些通用的内容。

需要注意的是，复位后到配置前，每个引脚将处于该引脚的默认状态。RA0 MCU 的所有 I/O 引脚在复位后都会立即作为输入引脚运行。一些引脚的状态在短时间内可能与预期不符。无论使用哪种配置方法，都是这样的情况。用户需考虑这个现象可能对应用产生的影响，包括对其他系统功能的影响。

I/O 端口可通过直接写入寄存器或使用 FSP 引脚配置器来配置。

9.1 多功能引脚选择设计方案

RA0 MCU 的大多数端口都具有多种外设功能。瑞萨提供了方便为每种 RA0 MCU 选择端口的工具，如：FSP 的引脚配置器。需要多个外设功能时，请在设计时使用以下方案选择端口功能。

- 先指定那些仅有一个端口选项的外设功能。例如，调试功能中的每个调试信号只有一个端口选项。当需要此功能时，请先分配这些端口。
- 然后指定那些具有有限几个端口选项的外设功能。例如，支持 CLKOUT 功能的 RA0E1 产品通常只有三个端口选项用于 CLKOUT 信号。
- 最后指定那些具有多个端口选项的外设功能。例如，串行阵列单元（SAU）通常有多个可用的端口选项。
- RA0 硬件手册中的 Pin Lists 显示，某些外设端口功能在功能名称末尾带有后缀（如“_A”）。在为 RA0 MCU 的大多数外设分配端口功能时，可以忽略这类后缀。详情请参阅 RA0 硬件手册中“**I/O Ports section, Peripheral Select Settings for Each Product**”，以及本文档第 9.3 节的内容。

9.2 设置端口用作 GPIO

有两种方法可以设置端口用作 GPIO，一种是使用端口控制寄存器（PODRm, PDRm, PIDRm, POSRm, EORRm, EOSRm），另一种是使用 PmnPFS 寄存器。

方法 1: 端口控制寄存器

- 通过将端口方向寄存器（PDRm）的相关位（PDRn）写“1”，引脚被设为输出。
- 端口方向位（PDRn）是可读/写的。将值设置为“1”后，引脚被设为输出。I/O 端口的默认状态为“0”（输入）。通过读取 PDRn 读取端口方向。
- 相应端口输出数据寄存器（PODRm）的端口输出数据位（PODRn）是可读/写的。读取 PODR 时，读取的是输出数据锁存器的状态（而不是引脚电平）。
- 端口状态寄存器（PIDRm）的端口输入位（PIDRn）是只读的。通过读取 PIDRm 寄存器中的 PIDRn 位读取引脚状态。

方法 2: 端口 mn 引脚功能选择（PmnPFS）寄存器

- 端口模式寄存器是可读/写的，用于指定单个引脚是用作 GPIO 还是外设引脚。
- 将引脚设置为输出时，建议先将端口的所需输出值写入数据锁存器，然后将方向寄存器设置为输出。虽然并非是在所有系统中都重要，但这样做可以防止正在被设置的端口上出现意外的输出毛刺。

一般来说，用端口控制寄存器配置端口能实现更快的读写，但可用的配置功能较少。而用 PmnPFS 寄存器会有更多可用的配置功能，但读写速度较慢。

通过瑞萨 FSP 的 Pin Configuration 配置复位后的 GPIO 引脚，如图 15 所示。通过 FSP 配置器将 P008 配置为输出和低电平。配置 GPIO 后，可以使用 FSP 的 HAL 层 API 对其进行控制。

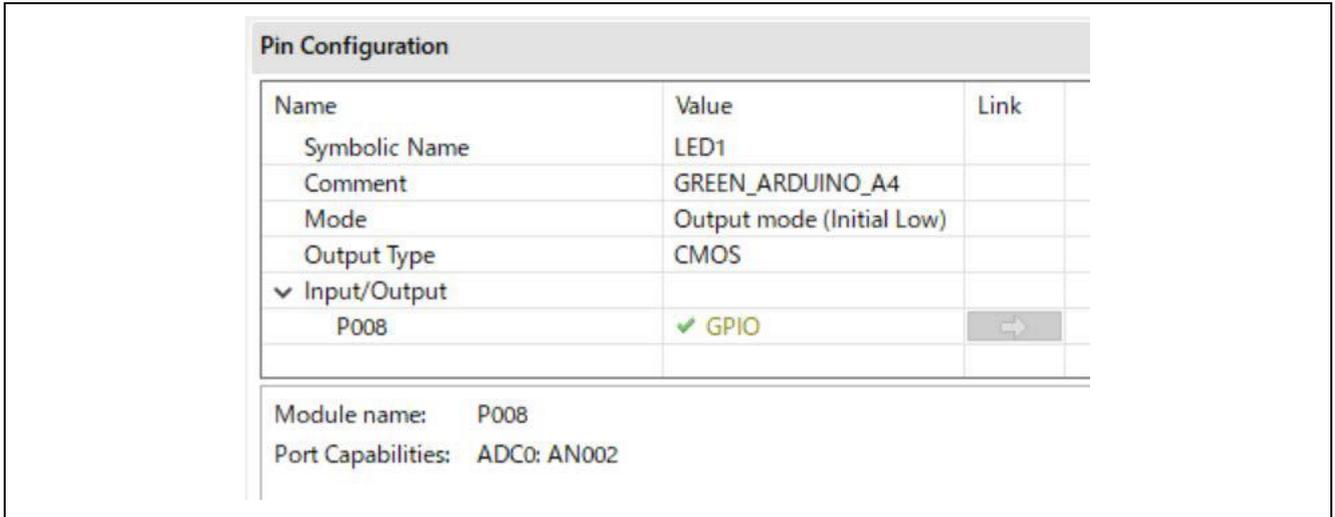


图 15. 通过 FSP 配置器将 P008 配置为输出低电平

9.2.1 内部上拉

- 大多数端口引脚都可以启用上拉电阻。上拉由每个端口 mn 引脚功能选择寄存器（PmnPFS）的上拉位（PCR）控制。每个 PmnPFS 寄存器的 PCR 位控制端口上的相应引脚。
- 必须首先通过 PmnPFS 寄存器中相关位将该引脚设置为输入。将 PCR 位设置为“1”以启用上拉，将 PCR 位设置为“0”以禁用上拉。
- 复位后，所有 PCR 寄存器被清零，因此所有上拉电阻都被禁用。关于一些例外情况，请参阅硬件手册。
- 每当有引脚被指定为外部总线引脚、GPIO 输出或外设功能输出引脚时，上拉电阻就会自动关闭。

9.2.2 开漏输出

- 配置为输出的引脚通常用作 CMOS 输出。
- 端口 0 到 9 的大多数引脚都可以配置为 NMOS 开漏输出。
- 各个端口 mn 引脚功能选择寄存器（PmnPFS）的 N 沟道开漏控制位（NCODR）控制哪些引脚在开漏模式下运行。将各 PmnPFS 寄存器的 NCODR 位设为“1”，则对应的引脚被设置为输出开漏，设为“0”，则端口被设置为 CMOS 输出。

9.3 设置和使用端口外设功能

端口 mn 引脚功能选择寄存器（PmnPFS）用于配置每个端口的特性。PSEL 位用来设置每个端口所选的外设功能。

- 由于大多数引脚具有多种功能，因此 RA0 MCU 具有引脚功能控制寄存器（PmnPFS），允许您更改分配给引脚的功能。
- 每个引脚都有自己的 PmnPFS 寄存器。

- 每个 PmnPFS 寄存器都可将引脚设为外设功能（PSEL 位）、IRQ 输入引脚（ISEL 位）或模拟输入引脚（PMC 位）。如果要将引脚用作模拟功能，请将引脚模式控制位（PMC）设为“1”，将 N 沟道开漏控制位（NCODR）设为“0”，端口方向位（PDR）设置为“0”。
- 有关各端口选项的详细信息，请参阅硬件手册中“I/O Ports”章的“Peripheral Select Settings for Each Product”一节。
- 所有 PmnPFS 寄存器均在复位时受到写保护。如需对这些寄存器进行写操作，必须首先使用写保护寄存器（PWPR）来启用写入。
- 设置 PmnPFS 寄存器时应注意，以免将单个功能分配给多个引脚。用户不会这样做，但 MCU 会允许这样的操作。如发生这种情况，引脚上的功能则不会被定义。

通过瑞萨 FSP 启用和配置 SAU_SPI1 引脚的示例如图 17 所示。

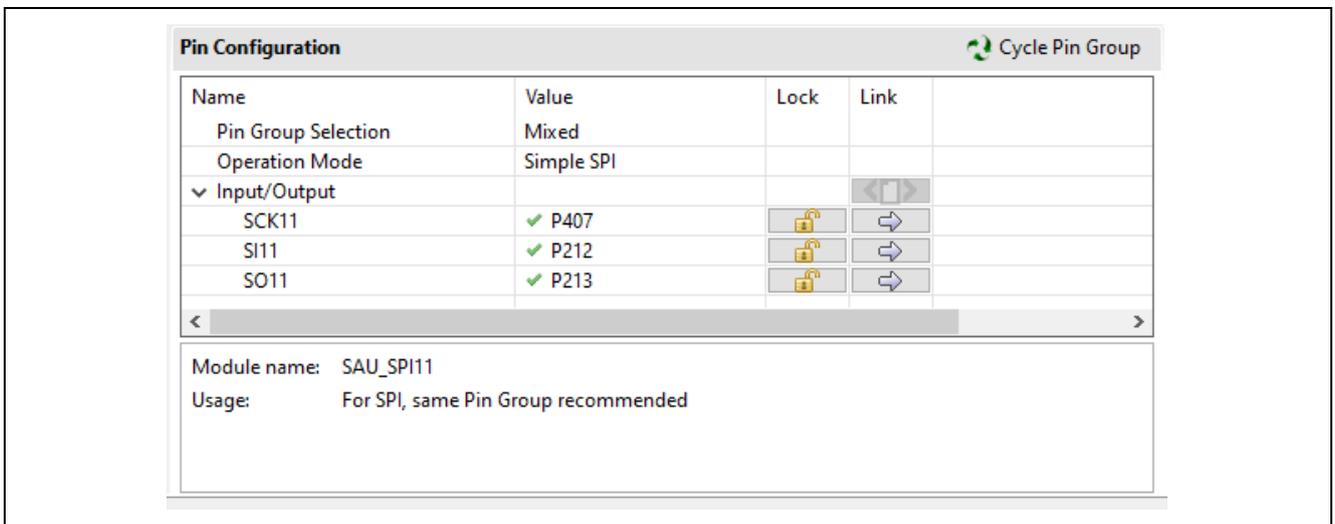


图 16. 通过 FSP 配置器启用 SAU_SPI1 引脚

9.4 设置和使用 IRQ 引脚

- 某些端口引脚可用作硬件输入中断（IRQ）。请参阅硬件手册中“I/O Ports”一章中“Peripheral Select Settings for each Product”的内容，以了解哪些引脚可用于您的 MCU。
- 要将端口引脚设置为 IRQ 引脚，必须将引脚 PFS 寄存器中的中断输入功能选择位（ISEL）设置为“1”。
- 引脚可同时用于 IRQ 和外设功能。用户应在引脚的 PFS 寄存器中设置 ISEL 和 PSEL 位。
- 相同编号的 IRQ 功能只能在一个引脚上启用。
- IRQ 引脚可在检测到以下情况时触发中断：
 - 下降沿
 - 上升沿
 - 上升沿和下降沿

使用 IRQ 控制寄存器（IRQCRi）选择相应的触发模式（下降沿、上升沿、上升沿和下降沿）。

通过瑞萨 FSP 启用和配置 IRQ 引脚的示例如图 17 和图 18 所示。

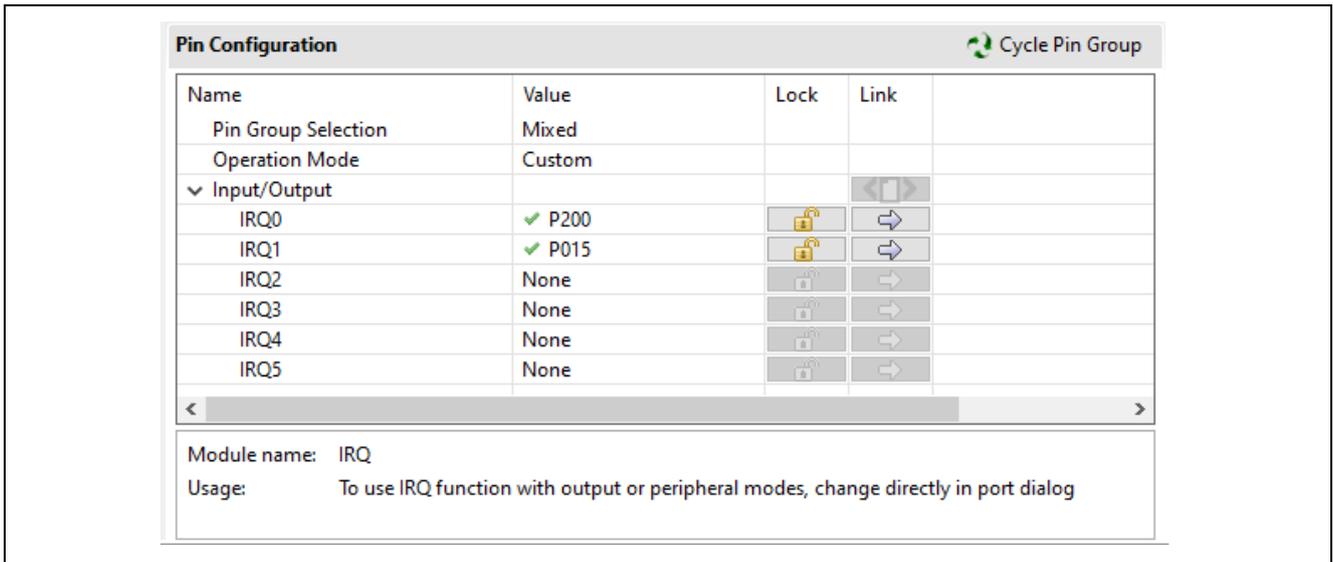


图 17. 使用 FSP 的 Pin Configurator 将 P200、P015 分别设为 IRQ0、IRQ1 输入

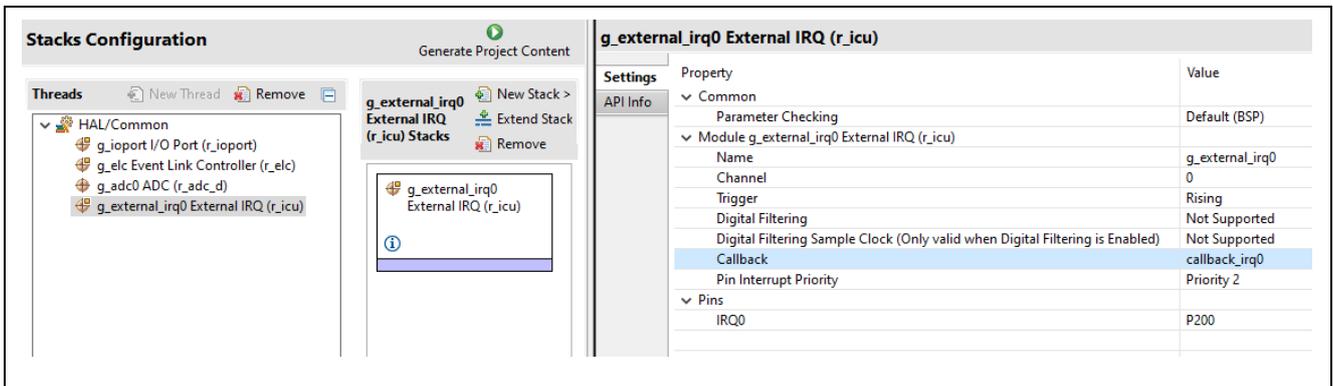


图 18. 通过 FSP 配置器配置 IRQ0

9.5 未使用的引脚

注意：某些引脚需要特定的处理：具体建议请参阅硬件手册的“Handling of Unused Pins”章节。

悬空的未使用引脚会消耗额外的功率，并使系统更易受到噪声问题的影响。使用以下方法可以处理未使用的引脚：

1. 第一种方法是将引脚设置为输入（复位后的默认状态），并使用电阻将引脚连接到 VCC 或 VSS。连接到 VCC 或 VSS 于 MCU 而言并无区别；但是，从系统噪声的角度来看，连接到 VSS 可能具有优势，所以连接到 VSS 可能是更典型的选择。请避免将引脚直接连接到 VCC 或 VSS，因为意外写入端口的方向寄存器（将引脚设置为输出）可能会造成输出短路。
2. 第二种方法是将引脚设置为输出。引脚电平是高电平还是低电平并不重要；但是，将引脚设置为输出并将输出设置为低电平将引脚在内部连接到接地层，这可能有助于解决整体系统噪声问题。将未使用的引脚设置为输出的一个缺点是端口的配置必须通过软件控制完成。当 MCU 保持在复位状态时，直到方向寄存器设置为输出时，该引脚将是悬空输入，可能会消耗额外的电流。如果在此期间可以容忍额外的电流，则此方法无需第一种方法所需的外部电阻。

3. 将引脚保留为输入并用外部电阻端接的方式，也可以通过使用许多端口上的内部上拉电阻来替代。这与将引脚设置为输出具有相同的限制（需要程序设置端口），这样做 MCU 不再会驱动引脚，从而有效防止与 GND、相邻引脚或 VCC 之间的意外短路。

9.6 不存在的引脚

每种 RA0 MCU 产品都有多种封装尺寸，分别具有不同的总引脚数。对于小于该 MCU 最大封装的任何封装（一般为 24 引脚、20 引脚或 16 引脚），请将 PDR 寄存器中不存在的端口的相应位设为“0”（输出），将 PODR 寄存器中的相应位设置为“0”。用户可以通过查看硬件手册中“I/O Ports”的“Specifications”表来查看每种 MCU 封装产品上可用的端口。请注意，不需要对不存在的引脚进行额外处理。

9.7 电气特性

普通 GPIO 端口通常需要 CMOS 电平输入（高 $\geq 0.8 \times VCC$ ，低 $\leq 0.2 \times VCC$ ）。某些 GPIO 端口具有施密特触发器输入，其输入要求略有不同。有关详细信息，请参阅硬件手册中“Electrical Characteristics”章节的内容。

10 模块 STOP 功能

为了最大限度地提高电源效率，RA0 系列 MCU 允许通过写入模块停止控制寄存器（MSTPCR_i, $i = A, B, C, D$ ）来单独停止片上外设。一旦模块停止，就无法访问该模块的寄存器。

复位后，除 DTC 外，大多数都处于模块停止状态。有关详细信息，请参阅硬件手册。

如需访问这些外设的寄存器，必须先将 MSTPCR_i 寄存器中的相应位写“0”使该外设退出停止模式，然后才可以启用。

可通过将 MSTPCR_i 寄存器中的相关位写“1”来停止该外设。

瑞萨 FSP 的 HAL 驱动程序会自动处理模块启动/停止功能。

11 中断控制单元

中断控制器单元 (ICU) 控制哪些事件信号链接到 NVIC 和 DTC 模块。ICU 模块还控制不可屏蔽中断。ICU 规格示例见图 19, ICU I/O 引脚示例见图 20。有关各 RA0 MCU 的详细信息, 请参阅硬件手册。

项目	描述	
可屏蔽中断	外设功能中断	<ul style="list-style-type: none"> 来自外设模块的中断 中断源数量:33
	外部引脚中断	<ul style="list-style-type: none"> 在下降沿、上升沿、上升沿和下降沿时检测中断。 每个中断源都可利用其中一种检测方法来设置 6 个中断源, 中断来自 IRQi (i=0 ~ 5) 引脚
	对 CPU 的中断请求(NVIC)	<ul style="list-style-type: none"> 36 个中断请求输出至NVIC
	DTC 控制	<ul style="list-style-type: none"> DTC 可以通过中断源来激活 选择中断源的方法与向NVIC 发出中断请求的方法相同
不可屏蔽中断注 ²	NMI 引脚中断	<ul style="list-style-type: none"> 来自 NMI 引脚的中断 下降沿或上升沿中断检测
	IWDT 下溢/刷新错误注 ³	向下计数器下溢或刷新错误时产生中断
	电压监测1中断注 ³	电压监测1电路 (LVD_LVD1) 的电压监测1中断
	RPEST	SRAM 奇偶校验错误中断
低功耗模式	<p>Sleep模式: 返回由非屏蔽中断或任何其他中断源使能</p> <p>Software Standby模式: 由不可屏蔽中断使能返回。中断可在 SBYEDCRn 寄存器中选择。</p> <p>Snooze 模式: 由不可屏蔽中断使能返回。可在 SBYEDCRn 寄存器中选择中断。</p> <p>请参见硬件手册中以下章节的内容:</p> <p>“11.2.14. SBYEDCR0 : Software Standby/Snooze End Control Register 0” ;</p> <p>“11.2.15. SBYEDCR1 : Software Standby/Snooze End Control Register 1”.</p>	
<p>注 1. 有关 DTC 激活源, 请参见硬件手册 Table 11.5。</p> <p>注 2. 不可屏蔽中断只能在复位后启用一次。</p> <p>注 3. 这些不可屏蔽中断也可用作可屏蔽中断。在用作可屏蔽中断时, 不要改变 NMIER 寄存器的复位值。要启用电压监测1中断, 请将 LVD1CR1.IRQSEL 位设为 1。</p>		

图 19. RA0E1 ICU 规格示例

引脚名称	I/O	描述
NMI	输入	不可屏蔽中断引脚
IRQi (i=0 ~ 5)	输入	外部中断请求引脚

图 20. RA0E1 ICU I/O 引脚示例

图 21 是使用 FSP 配置器启用和配置使用中断示例。ICU 和中断通过 FSP 配置为 HAL 驱动程序配置的一部分。

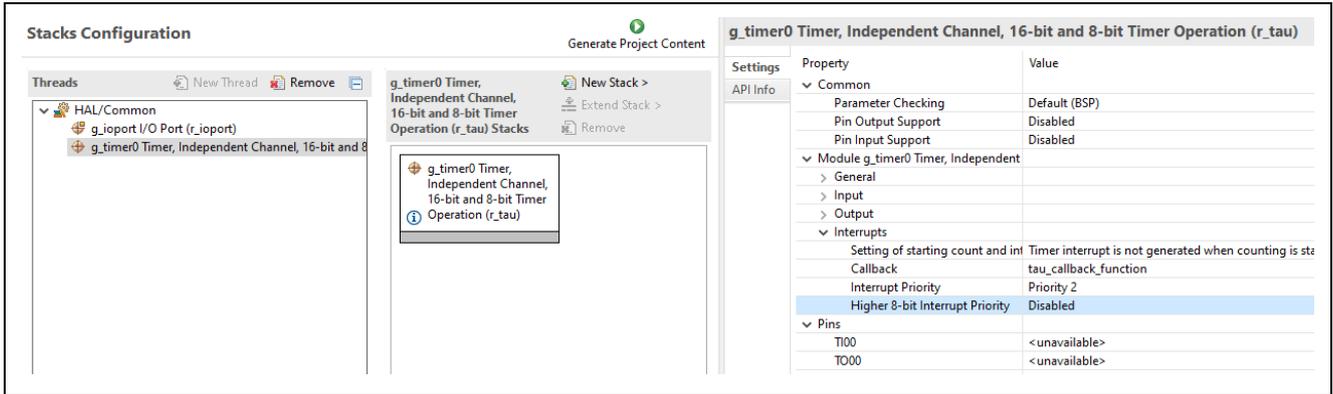


图 21. 启用 TAU 中断并设置中断服务程序将调用的用户回调函数

12 低功耗

RA0 产品具有多种降低功耗的功能。其中包括设置时钟分频器、停止模块、Normal 模式下选择运行功率控制模式以及过渡到低功耗模式。有关详细信息，请参阅硬件手册中的“Low Power Modes”章节的内容。

RA0 MCU 支持三种不同类型的低功耗模式（LPM）：

- Sleep 模式 - CPU 停止运行，但其他外设功能和时钟源还在运行（最快唤醒）
- Snooze 模式 - CPU 停止运行，但某些外设功能和时钟源可根据需要选择是否运行
- Software Standby 模式 - CPU 以及大部分外设功能和时钟源均停止（最省电）

下表概述了可用于降低功耗的功能。

表 9. 低功耗模式的功能规格

项目	规范
通过修改时钟信号来降低功耗	HOCO、MOCO 和 MOSC 的分频比可以独立选择。 ^{注1}
模块停止	每个外设模块的功能可以独立停止。
低功耗模式	<ul style="list-style-type: none"> • Sleep 模式 • Snooze 模式 • Software Standby 模式
电源控制模式	<p>在 Normal、Sleep 以及 Snooze 模式下，通过根据运行频率选择合适的运行功率控制模式，可以降低功耗。</p> <p>有四种运行功率控制模式：</p> <ul style="list-style-type: none"> • 高速模式 • 中速模式 • 低速模式 • 副振荡器模式

注：1. 有关详细信息，请参阅硬件手册中的“Clock Generation Circuit”章节的内容。

表 10 列出了进入到低功耗模式的条件、CPU 和外设模块的状态以及退出每种模式的方法。

表 10. 低功耗模式

运行状态 ^注	Sleep 模式	Software Standby 模式	Snooze 模式
进入条件	设置 SBYCR.SSBY=0 之后，执行 WFI 指令。	设置 SBYCR.SSBY=1 之后，执行 WFI 指令。	软件待机模式下的 Snooze 请求触发器。 SBYCR.SSBY=1。
退出方式	所有中断。 该模式下可用的任何复位。	该模式下可用的中断。 该模式下可用的任何复位。	该模式下可用的中断。 该模式下可用的任何复位。
通过中断取消后的状态	程序执行状态（中断处理）	程序执行状态（中断处理）	程序执行状态（中断处理）
通过复位取消后的状态	复位状态	复位状态	复位状态

注意：1. 有关更多详细信息，请参阅硬件手册中的“Operating conditions of each low power mode”。

RA0 产品可通过寄存器设置使 MCU 在 Normal 模式、Sleep 模式和 Snooze 模式下以较低的功耗运行。这些模式通过 FLMODE 寄存器降低功耗。

这些模式称为运行功率模式（Operating Power Control Modes）。

以下是运行功率模式和每种模式下可用的最大时钟和电压水平。

表 11. 每种运行功率模式下的可用振荡器

模式	振荡器				
	高速片上振荡器	中速片上振荡器	低速片上振荡器	主时钟振荡器	副时钟振荡器
高速	可用	可用	可用	可用	可用
中速	可用	可用	可用	可用	可用
低速	可用	可用	可用	可用	可用
副振荡器速度	不可用	不可用	可用	不可用	可用

虽然 FLMODE 寄存器中的值可设置为任何低功耗运行模式，但时钟频率必须设置为满足所需模式的要求。否则 FLMODE 寄存器中的设置不会对降低功耗产生任何作用。

要实现最低的功率值，请在时钟生成电路中使用最大可能的分频器。

下图以 FLMODE 寄存器中的 MODE 位和 ICLKSCR 寄存器中的 CKST 位设置的每种运行电源控制模式的相对功耗为例进行说明。

工作功率控制模式	MODE[1:0]位	ICLKSCR.CKST 位	功耗
高速模式	11b	0	高
中速模式	10b	0	↓
低速模式	01b	0	↓
副振荡器速度模式	xxb	1	低

图 22. 运行功率模式

低功耗模式可由各种中断源（如 RES 引脚复位、上电复位、电压监视器复位和外设中断）取消。请参阅 MCU 硬件手册中的低功耗模式部分，了解不同 LPM 的中断源列表。

只有 Snooze 模式是由 Snooze 请求触发，即从 Software Standby 模式进入 Snooze 模式。其他的低功耗转换则是通过执行 WFI 指令并在待机控制寄存器 (SBYCR) 中进行相应设置来完成。

瑞萨 FSP 提供低功耗模式 (LPM) 驱动器和驱动器配置器，用于设置低功耗模式、唤醒源/取消源等。

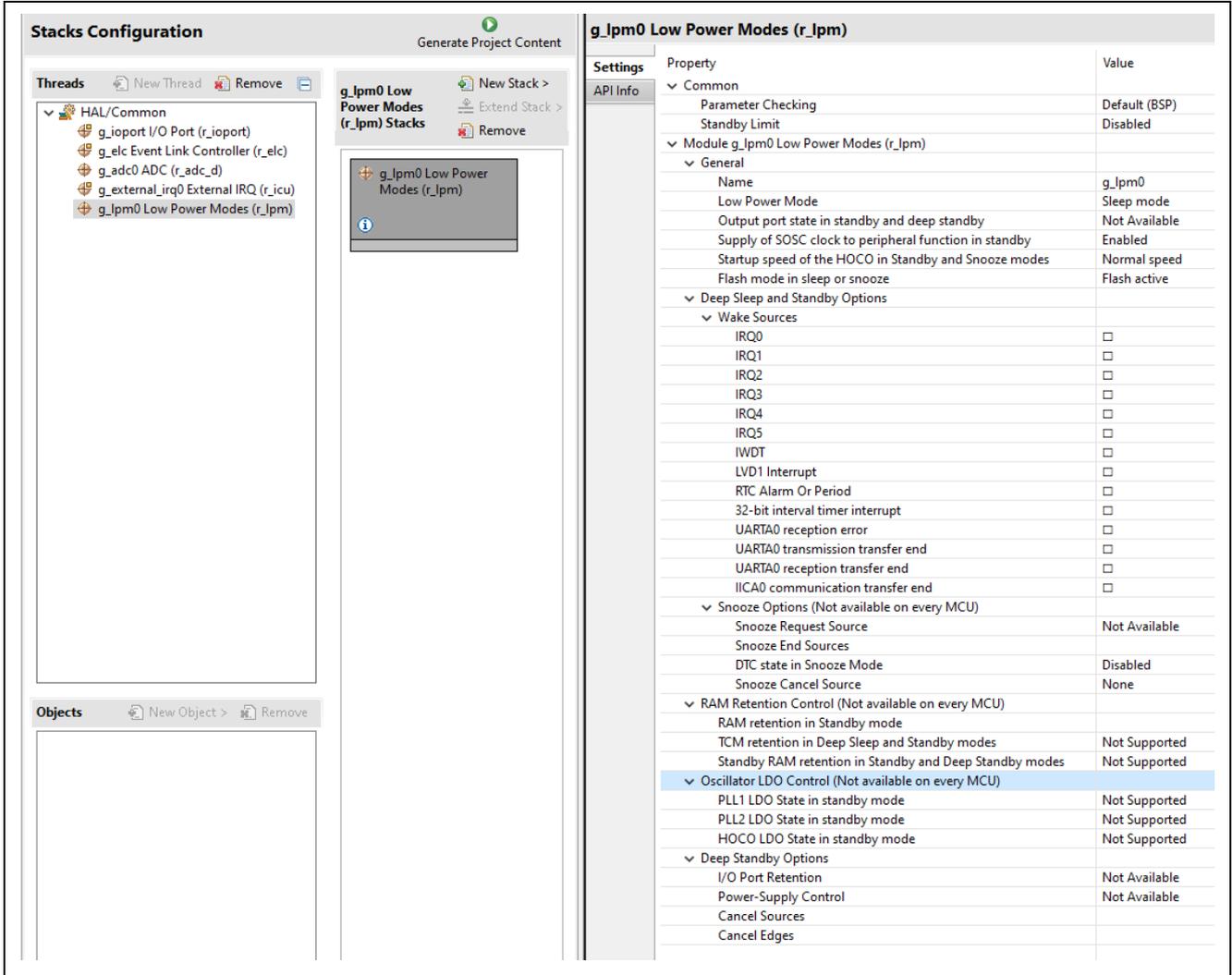


图 23. 使用 FSP 配置器设置低功耗模式

FSP 配置器设置特定 LPM 模式后，LPM 驱动程序的 API 可用于初始化 LPM 驱动程序并将 MCU 置于配置的 LPM 模式：

```
/* Open LPM driver and initialize LPM mode */
err = R_LPM_Open(&g_lpm_ctrl_instance_ctrls[g_lpm_transition_pos],
                &g_lpm_ctrl_instance_cfgs[g_lpm_transition_pos]);
/* Handle error */
if (FSP_SUCCESS != err)
{
return (err);
}
/* Transition to configured LPM mode */
err = lpm_mode_enter(g_lpm_transition_sequence[g_lpm_transition_pos]);
/* Handle error */
if (FSP_SUCCESS != err)
{
return (err);
}
```

13 外部总线

RA0 MCU 的总线由主总线和从接口组成。主总线与从接口列表见图 24。总线配置见图 25。

注：便于执行 Arm® Cortex®代码，内存空间必须为小端。

总线类型		描述
主总线	系统总线 (CPU)	<ul style="list-style-type: none"> 与 CPU 连接 与片上存储器和内部外设总线连接
	DMA 总线	<ul style="list-style-type: none"> 与 DTC 连接 与片上存储器和内部外设总线连接
从接口	内存总线1	与代码闪存连接
	内存总线4	与SRAM0连接
	内部外设总线1	与外设模块相关的系统控制连接
	内部外设总线3	<ul style="list-style-type: none"> 与外设模块 (ELC、IWDT、MSTP 和 CRC) 连接 与外设模块 (I/O 端口、ADC12、SAU0、SAU1、TAU、RTC、IICA、UARTA、TML32、PCLBUZ 连接
	内部外设总线7	与 TRNG 连接
	内部外设总线9	连接至代码闪存 (P/E (编程/擦除) 内)、数据闪存

图 24. RA0 总线规格

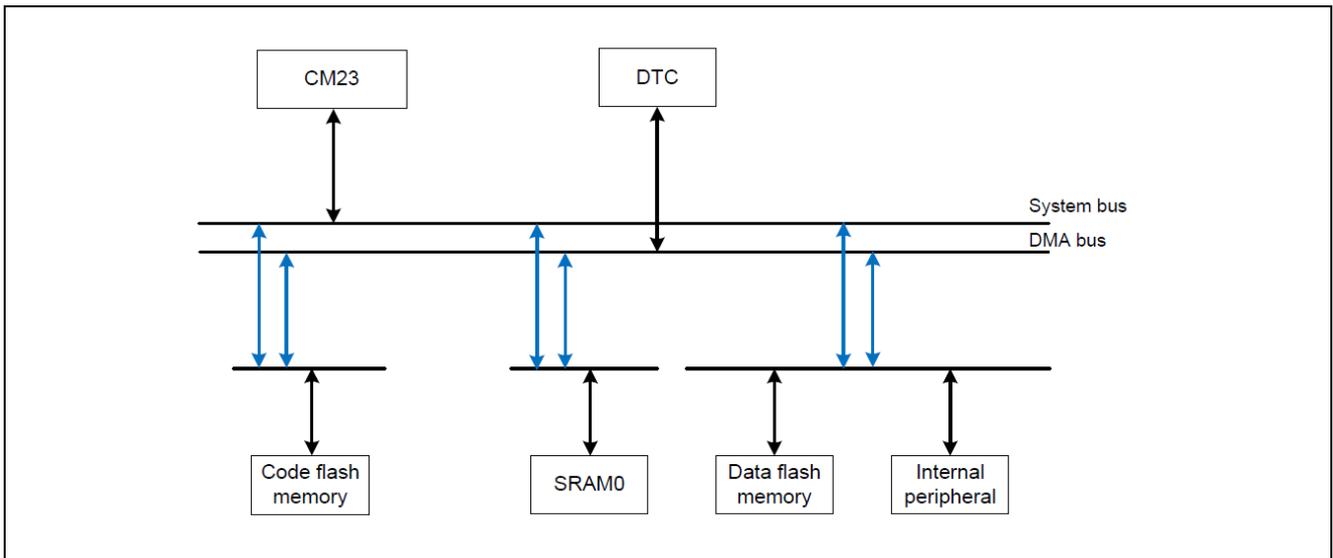


图 25. RA0 MCU 简易总线配置

13.1 总线错误监测

监测系统监控每个单独的区域。一旦检测到错误，便根据 AHB-Lite 错误响应协议将错误返回给请求的主 IP。

13.1.1 总线错误类型

各总线上可能发生的错误类型：

- 非法地址访问

13.1.2 总线错误发生时的运行情况

发生总线错误时，运行无法保证，错误将返回给请求的主 IP。每个主机发生的总线错误都存储在 BUSnERRADD 和 BUSnERRSTAT 寄存器中。这些寄存器只能通过复位清除。有关详细信息，请参阅硬件手册中的“Bus Error Address Register (BUSnERRADD)”和“Bus Error Status Register (BUSnERRSTAT)”。

注意：DTC 不会接收总线错误，因此它的运行不受总线错误的影响。

14 通用布线建议

14.1 数字域与模拟域

瑞萨 RA0 MCU 具有三种主要的引脚功能：电源、数字和模拟。

通常，电源引脚专用于电压和基准输入，不具有多种功能。电源引脚通常专用于 MCU 中的特定部分或域。例如，MCU 的主电源电压将为数字内核、许多数字外设功能和许多数字 I/O 引脚供电。数字域可以定义为数字电路、数字 I/O 引脚和相关电源引脚。指定用于模拟功能的电源引脚（如 AVCC0 和相关的 AVSS0）在 MCU 内提供特定的模拟电路，该电路与数字域电路分开。模拟域可以定义为模拟电路、模拟 I/O 引脚和相关电源引脚。

数字信号通常是与时钟相关的周期性重复开关模式的信号。数字信号的过度过程或者转换过程往往是相对尖锐的边缘，转换之间具有稳定的高电平或低电平。在指定的时间范围内，每个信号必须稳定在可接受的电压电平（称为逻辑状态）。信号状态通常以预定的时钟间隔进行采样，使用时钟的边沿转换来评估相关的数据信号。只要电压电平保持在指定范围内，数字信号电压电平的微小变化通常是可以接受的。然而，对数字信号的较大外部影响会对数字信号产生严重影响，这可能导致在对数据进行采样时出现不正确的逻辑状态。

模拟信号通常完全不同。模拟信号可能是周期性的，但模拟信号的评估通常是对一定范围内的电压测量，而不是逻辑状态。模拟信号的电压电平根据特定的触发事件进行采样，并使用 MCU 中的模拟电路处理测量结果。模拟测量的精度与采样电压电平的精度直接相关。任何可能改变模拟输入信号电压电平的不必要的外部影响，即使很轻微，也会影响测量的准确性。

由于瑞萨 RA0 MCU 的 I/O 引脚具有高度多路复用特性，因此许多 I/O 引脚可用于模拟或数字功能。这可能导致数字和模拟功能可能重叠并导致数据错误的情况。

为了最大程度地减少数字和模拟信号域之间的潜在问题，请考虑以下策略。

- 分配 I/O 引脚功能时，请选择引脚功能，以便模拟引脚和数字引脚尽可能在物理上分开。
- 每个模拟信号应尽可能与所有其他信号分开。
- PCB 布线应尽可能隔离每个模拟信号。避免在同一区域出现任何其他信号，无论是模拟信号还是数字信号。
- 确保模拟电源电压和模拟基准电压具有适当的交流滤波器。这可以是位于 MCU 电压引脚附近的推荐电容或适当的电感滤波器的形式。目标是提供电压纹波很小或没有电压纹波的电压电源和基准电压。
- 在 PCB 设计中使用专用电源层时，应避免在模拟电压区域出现数字信号，并避免在数字电压区域出现模拟信号。

对于高度敏感的应用，强烈建议使用仿真工具评估特定设计，以了解电路设计对性能的影响。这也包括精密传感器设计或超高速图总线接口等应用。有关每个外设功能的具体要求，请参阅硬件手册中的“Electrical Characteristics”一章。

14.2 高速信号设计注意事项

随着数字信号时钟速度的提高，外部刺激对这些信号的影响会变得更加显著。一些外设功能可归类为高速数字信号。对于高速数字信号，应考虑其他设计因素。

串扰是指一个信号的转换对另一个附近信号产生感应影响的情况。当这种串扰效应足够强时，第一个信号可能会导致第二个信号出现误差。为了减少串扰的影响，请使用以下通用 PCB 布线指南。

- 在同一布线层上的信号之间提供足够的间隔。通常，同一组数字信号之间至少保留 1 个走线宽度间隔，不同组的数字信号之间至少保留 3~5 个走线宽度间隔。
- 在同一布线层上的时钟信号和数据信号之间提供额外的间隔。通常，在时钟和任何其他数字信号之间至少保持 3~5 个走线宽度间隔。
- 避免在任何相邻的布线层上并行走数字信号。如果信号必须在相邻信号层上，请尽可能仅使用正交交叉的方式。

如果可能，在信号层之间使用电源层或接地层将 PCB 信号层分开。电源层或接地层的实心铜可以充当数字信号的“屏蔽层”。

每个标准化接口都有特定的要求。为确保 PCB 的设计能够避免信号串扰问题，强烈建议参考设计中每个接口的相关标准。

14.3 信号组选择

某些引脚名称添加了 `_A`、`_B`、`_C`、`_D`、`_E` 或 `_F` 后缀来指示信号组。RA0 产品引脚（SAU 和 IICA 除外）在分配功能时可忽略这些后缀。对于 SAU 和 IICA，除 SCL11 和 SCK11 外，只能选择后缀相同的信号。对于这两个信号组，禁止同时使用具有不同后缀的同一信号。对于所有其他信号组，可以为每个功能信号选择最方便的引脚分配。

如果外设功能的引脚名称没有信号组后缀，则可以安全地为每个功能信号选择最方便的引脚分配。

请参阅硬件手册“[I/O Ports](#)”一章中的“[Peripheral Select Settings for each Product](#)”和“[Notes on the PmnPFS Register Setting](#)”部分。

15 参考文献

制作本快速设计指南时参考了以下文档。请访问 [Renesas 官网](#) 以获取最新资料。

参考资料	文档编号	说明
1	R01AN6277	Renesas RA Family Design Guide for Sub-Clock Circuits
2	R01UH1040	Renesas RA0E1 Group, User's Manual: Hardware

网页和支持

可从以下链接了解更多关于 RA 系列的要点，下载组件和相关文档，以获得支持。

RA 产品	www.renesas.com/ra
RA 产品支持论坛	www.renesas.com/ra/forum
RA FSP	www.renesas.com/FSP
瑞萨技术支持	www.renesas.com/support

更新履历

版本号	日期	描述	
		页数	概述
1.00	2024 年 9 月 24 日	-	初版发布

注意

1. 本文件中电路、软件和其他相关信息的描述仅用于说明半导体产品的操作和应用示例。用户应对产品或系统设计中电路、软件和信息纳入或任何其他用途承担全部责任。对于您或第三方因使用这些电路、软件或信息而引起的任何损失和损害，Renesas Electronics 不承担任何责任。
2. Renesas Electronics 特此声明，对于因使用本文件中所述的 Renesas Electronics 产品或技术信息（包括但不限于产品数据、图纸、图表、程序、算法和应用示例）而引起的侵权或与第三方有关的专利、版权或其他知识产权的任何其他索赔，概不承担任何责任和赔偿。
3. 对 Renesas Electronics 或其他公司的任何专利、版权或其他知识产权均不授予任何明示、暗示或其他形式的许可。
4. 您应负责确定需要从任何第三方获得哪些许可，并在需要时为合法进口、出口、制造、销售、使用、分销或以其他方式处置包含 Renesas Electronics 产品的任何产品获得此类许可。
5. 不得对 Renesas Electronics 产品的全部或部分进行更改、修改、复制或逆向工程。对于因更改、修改、复制或逆向工程而导致您或第三方蒙受的任何损失或损害，Renesas Electronics 不承担任何责任。
6. Renesas Electronics 产品根据以下两个质量等级进行分类：“标准”和“优质”。Renesas Electronics 每种产品的预期应用取决于产品的质量等级，具体如下所示。
“标准”：计算机、办公设备、通信设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备、工业机器人等
“优质”：运输设备（汽车、火车、轮船等）；交通管制（交通信号灯）；大型通信设备；关键金融终端系统；安全控制设备等
除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中明确指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 产品不适合或不授权用于可能对人类生命构成直接威胁或造成人身伤害（人造生命支持设备或系统；手术植入物等），或者可能造成严重的财产损失（空间系统、海底中继器、核动力控制系统、飞机控制系统、关键设备系统、军事装备等）的产品或系统。对于因使用任何与 Renesas Electronics 数据手册、用户手册或其他 Renesas Electronics 文档不一致的 Renesas Electronics 产品而引起的您或任何第三方所造成的任何损坏或损失，Renesas Electronics 不承担任何责任。
7. 没有任何半导体产品是绝对安全的。尽管 Renesas Electronics 的硬件或软件产品中可能实施了任何安全措施或功能，Renesas Electronics 对因任何漏洞或入侵（包括但不限于以任何未经授权的方式访问或使用 Renesas Electronics 产品或使用 Renesas Electronics 产品的系统）而产生的任何后果概不负责。RENESAS ELECTRONICS 不担保或保证 RENESAS ELECTRONICS 产品或使用 RENESAS ELECTRONICS 产品创建的任何系统不会被破坏，或者可免于数据损坏、攻击、病毒、干扰、黑客攻击、数据丢失或失窃或其他安全入侵（“漏洞问题”）。RENESAS ELECTRONICS 不承担任何由任何漏洞问题引起的或与之相关的任何和所有责任或义务。此外，在适用法律允许的范围内，RENESAS ELECTRONICS 不对本文件和任何相关或附带的软件或硬件提供任何和所有明示或暗示的保证，包括但不限于对适用性或特定用途的适用性的暗示保证。
8. 使用 Renesas Electronics 产品时，请参见最新的产品信息（数据手册、用户手册、应用笔记、可靠性手册中的“处理和使用半导体器件的一般说明”等），并确保使用条件符合 Renesas Electronics 在最大额定值、运行电源电压范围、散热特性和安装等方面的规定。对于因在超出上述规定范围的情况下使用 Renesas Electronics 产品而引起的任何失常、故障或事故，Renesas Electronics 不承担任何责任。
9. 尽管 Renesas Electronics 致力于提高 Renesas Electronics 产品的质量和可靠性，但半导体产品具有特定的特性，例如在特定速率下发生故障以及在某些使用条件下出现故障。除非在 Renesas Electronics 数据手册或 Renesas Electronics 其他文档中指定为高可靠性产品或用于恶劣环境的产品，否则 Renesas Electronics 的产品将不受抗辐射设计的约束。用户应负责采取安全措施，以防止人身伤害、火灾造成的伤害，和/或因 Renesas Electronics 产品发生故障或失常而对公众造成的危险，例如硬件和设备的安全设计，包括但不限于冗余、火控和故障预防、针对老化退化的适当处理或其他适当的措施。由于对微型计算机软件进行评估非常困难且无实操性，因此用户有责任评估自己生产的最终产品或系统的安全性。
10. 请联系 Renesas Electronics 销售办事处，以获取有关环境事宜的详细信息，例如每个 Renesas Electronics 产品的环境相容性。用户有责任认真、充分地研究有关纳入或使用受控物质的适用法律和法规（包括但不限于欧盟 RoHS 指令），并按照所有适用法律和法规使用 Renesas Electronics 产品。对于因您未遵守适用的法律和法规而造成的损坏或损失，Renesas Electronics 不承担任何责任。
11. Renesas Electronics 产品和技术不得被用于或纳入为任何适用的本国或外国法律、法规所禁止制造、使用或销售的产品或系统范围内。用户应遵守由对当事方或交易拥有管辖权的任何国家/地区的政府颁布和管理的任何可适用的出口控制法律和法规。
12. 应由 Renesas Electronics 产品的购买方或分销商，或者对产品进行分发、处置或以其他方式出售或转让给第三方的任何其他当事方，负责将本文中阐明的内容和条件提前通知前述第三方。
13. 未经 Renesas Electronics 事先书面同意，不得以任何形式全部或部分重印、再现或复制本文件。
14. 如果对本文中包含的信息或 Renesas Electronics 产品有任何疑问，请联系 Renesas Electronics 销售办事处。

（注 1）本文件中的“Renesas Electronics”是指 Renesas Electronics Corporation，也包括其直接或间接控制的子公司。

（注 2）“Renesas Electronics 产品”是指 Renesas Electronics 开发或制造的任意产品。

（版本 5.0-1 2020 年 10 月）

公司总部

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
<https://www.renesas.com/>

商标

Renesas 和 Renesas 徽标是 Renesas Electronics Corporation 的商标。所有商标和注册商标都是各自所有者的财产。

联系信息

有关产品、技术、文档最新版本或离您最近的销售办事处的更多信息，请访问：www.renesas.com/contact/。