Renesas RA 产品家族

RA6 快速设计指南

简介

本文档回答了常见的问题，并指出了单片机 (MCU) 系统设计的一些细节之处，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注意的一些关键项目。本文档还从应用的角度探讨了一些设计决策。

目标器件

RA6系列MCU

目录

1. 电源 4
1.1 参考资料 5
2. 仿真器支持 6
2.1 SWD 接口 6
2.2 JTAG 接口 7
2.3 使用 SCI 的串行编程接口 8
2.4 使用 SCI 的串行编程接口：支持 TrustZone® 的器件 9
2.5 复用仿真器接口 10
2.6 仿真器连接的软件设置 10
2.6.1 SWD 和 JTAG 接口 10
2.6.2 跟踪端口 11
3. MCU 工作模式 11
4. 选项设置存储器 12
4.1 选项设置存储器寄存器 13
5. 时钟电路 14
5.1 复位条件 14
5.2 时钟频率要求 14
5.2.1 USB 通信要求 15
5.2.2 以太网控制器要求 15
5.2.3 ROM 或数据闪存的编程和擦除要求 15
5.2.4 SDRAM 控制器的要求 15
5.3 降低时钟生成电路 (CGC) 的功耗 16
5.4 写入系统时钟控制寄存器 16
5.5 时钟设置示例 16
5.6 HOCO 精度 17
5.7 闪存接口时钟 17

5.8 电路板设计 17
5.9 外部晶体谐振器选择 17
6. 复位要求和复位电路 18
6.1 引脚复位 19
6.2 上电复位 19
6.3 独立看门狗定时器复位 19
6.4 看门狗定时器复位 20
6.5 电压监视复位 20
6.6 深度软件待机复位 20
6.7 软件复位 20
6.8 其他复位 20
6.9 冷/热启动的确定 20
6.10 确定复位源 20
7. 使用TrustZone® 21
7.1 Arm® TrustZone® 技术的实现 21
7.2 支持 TrustZone 的仿真器 22
8. 存储器 24
8.1 SRAM 25
8.2 待机 SRAM 25
8.3 外设 I/O 寄存器 26
8.4 片上闪存 26
8.4.1 后台运行 27
8.4.2 ID 代码保护 27
8.4.3 器件生命周期管理 28
8.4.4 闪存块保护 28
8.4.5 内存保护单元 28
8.5 外部存储器 29
8.5.1 使用外部 16 位存储器件 29
8.5.2 SDRAM 初始化示例 29
8.6 数据对齐 31
8.7 字节顺序限制 31
9. 寄存器写保护 32
10. I/O 端口配置 33
10.1 多功能引脚选择设计策略 33
10.2 设置端口并将其用作 GPIO 33
10.2.1 内部上拉 34
10.2.2 漏极开路输出 34
10.2.3 驱动能力 34

10.3 设置和使用端口外设功能 35
10.4 设置和使用中断（IRQ）引脚 36
10.5 未使用的引脚 38
10.6 不存在的引脚 38
10.7 电气特性 38
11. 模块停止功能 38
12. 中断控制单元 38
13. 低功耗 40
14. 外部总线 43
14.1 总线宽度和复用 43
14.2 总线信号的驱动能力 43
14.3 总线错误 43
15. 一般布线实践 43
15.1 数字域与模拟域 43
15.2 高速信号设计注意事项 44
15.3 信号组选择 45
16. 参考资料 45
版本历史记录 47

1. 电源

RA 产品家族具有数字电源和模拟电源。电源使用以下引脚。

表 1. 数字电源

符号	名称	说明
VCC	电源	电源引脚。连接到系统电源。通过放置在 VCC 引脚附近的 0.1 µF 电容将此引脚连接至 VSS。
VSS	接地	接地
VCL	电源	通过 VCL 引脚附近的 0.1 µF 电容将此引脚连接至 VSS。
VCL0	电源	通过 VCL0 引脚附近的 0.1 µF 电容将此引脚连接至 VSS。
VBATT	备用电源	备用电源引脚。在没有 VCC 的情况下，为 RTC 和子时钟振荡器供电。当不使用 VBATT 引脚时，连接到 VCC 或 VSS。
VCC_USB	USB FS 电源	全速 USB 电源引脚。将此引脚连接到 VCC。通过放置在 VCC_USB 引脚附近的 0.1 µF 电容将此引脚连接至 VSS_USB。
VSS_USB	USB FS 接地	全速 USB 接地引脚。将此引脚连接到 VSS。
VCC_USBHS[1]	USB HS 电源	高速USB 电源引脚。将此引脚连接到 VCC。通过放置在 VCC_USBHS 引脚附近的 0.1 µF 电容将此引脚连接至 VSS1_USBHS 或 VSS2_USBHS。
VSS1_USBHS[1]	USB HS 接地	高速 USB 接地引脚。将此引脚连接到 VSS。
VSS2_USBHS[1]	USB HS 接地	高速 USB 接地引脚。将此引脚连接到 VSS。

注：1. 仅适用于带有 高速USB 外设的器件。

表 2. 模拟电源

符号	名称	说明
AVCC0	模拟电源	各个模块的模拟电压电源引脚。将此引脚连接到与 VCC 引脚相同的电压。
AVSS0	模拟地	各个模块的模拟地。将此引脚连接到与 VSS 引脚相同的电压。
VREFH0	12 位 ADC 高参考电压	12 位 A/D 转换器（单元 0）和 AN000 至 AN002 的采样保持电路的参考电源引脚。如果不使用这些功能，则将此引脚连接到 AVCC0。
VREFL0	12 位 ADC 低参考电压	12 位 A/D 转换器（单元 0）和 AN000 至 AN002 的采样保持电路的模拟参考接地引脚，。如果不使用这些功能，则将此引脚连接至 VSS。
VREFH	12 位 ADC 和 DAC 模拟电源	12 位 A/D 转换器（单元 1）、AN100 至 AN102 的采样保持电路以及 D/A 转换器的参考电压输入引脚，用作这些模块的模拟电源。如果不使用这些功能，则将此引脚连接到 AVCC0。
VREFL	12 位 ADC 和 DAC 模拟地	12 位 A/D 转换器（单元 1）、AN100 至 AN102 的采样保持电路以及 D/A 转换器的参考接地引脚，用作这些模块的模拟地。如果不使用这些功能，则将此引脚连接至 VSS。
AVCC_USBHS[1]	USB HS 模拟电源	高速USB 模拟电源。

符号	名称	说明
AVSS_USBHS[1]	USB HS 模拟地	高速USB 模拟接地引脚。必须短接到 PVSS_USBHS 引脚。
PVSS_USBHS[1]	USB HS PLL 电路接地	高速USB PLL 电路的接地引脚。必须短接到 AVSS_USBHS 引脚。
USBHS_RREF[1]	USB HS 参考电流	高速USB参考电流源引脚。通过 2.2 kΩ 电阻 (±1%) 将此引脚连接到 AVSS_USBHS 引脚。

注：1. 仅适用于带有 USB 高速外设的器件。

1.1 参考资料

有关 RA MCU 系列的电源的更多信息，请参见以下文档：

- R01UH0884EJ0100-RA6M1 RA6M1 系列，RA6M1 系列用户手册：硬件
- R01UH0885EJ0100-RA6M2 RA6M2 系列，RA6M2 系列用户手册：硬件
- R01UH0886EJ0100-RA6M3 RA6M3 系列，RA6M3 系列用户手册：硬件
- R01UH0890EJ0110-RA6M4 RA6M4 系列，RA6M4 系列用户手册：硬件
- R01UH0891EJ0110-RA6M5 RA6M5 系列，RA6M5 系列用户手册：硬件

Arm Cortex-M4 和 Arm Cortex-M33 器件之间的章节编号可能会有所不同。

第 1 章“**概述**”列出了每个封装中的电源引脚以及建议的旁路电容。

“**复位**”一章探讨了上电复位以及如何将其与其他复位源区分开。

“**低电压检测**”一章详细介绍了可用于监视电源的低压检测电路。“**选项设置存储器**”一章还介绍了如何在启动时自动使能低压检测 0 电路。

“**备用电池功能**”一章介绍了如何为 RTC 和子时钟振荡器提供备用电池。

如果打算使用片上模数转换器 (ADC) 或数模转换器 (DAC)，请参见“**12 位 A/D 转换器 (ADC12)**”和“**12 位 D/A 转换器 (DAC12)**”，了解有关如何为这些外设提供经过滤波的电源的详细信息。

表 3. RA6 MCU 系列用户手册：硬件

章节名称	说明
概述	列出每个封装中的电源引脚，并提供有关端接和旁路的注意事项。
复位	介绍上电复位以及如何将其与其他复位源区分开。
电压检测电路	详细介绍可用于监视电源的低压检测电路。
低功耗模式	可使用低功耗模式降低电源电压。有关工作模式对电源要求有何影响的详细信息，请参见本章。
备用电池功能	介绍如何为 RTC 和子时钟振荡器提供备用电池。
12 位 A/D 转换器 12 位 D/A 转换器	如果打算使用片上 A/D 或 D/A 转换器，可通过这些章节了解有关如何为这些外设提供经过滤波的电源的详细信息。
时钟生成电路	提供有关如何配置和使用可用时钟的详细说明，包括 PCB 设计建议。

2. 仿真器支持

RA MCU 器件支持使用 SWD 或 JTAG 通信进行调试，并可使用 SCI 通信进行串行编程。借助该仿真器，可以轻松地在调试和串行编程之间进行切换。

SWD 或 JTAG 仿真器接口应连接到符合 ARM 标准的 10 引脚或 20 引脚插座。添加了 MD、TXD 和 RXD 引脚，以使用 SCI 通信进行串行编程。

必须使用串行编程接口对 Arm® TrustZone® IDAU 边界寄存器设置进行编程。对于支持 TrustZone® 的器件，建议使用板上线“或”电路连接 P300/SWCLK/TCK 和 P201/MD 引脚，以同时使用调试和串行编程。

在开发和原型机调试过程中会经常使用仿真器，但量产后可能就再使用仿真器了。如果在量产时无需支持仿真器，请务必正确设置相关的端口，详情请参见《硬件用户手册》中的“未使用引脚的处理”部分，也可以参考本文档的10.5章节。

2.1 SWD 接口

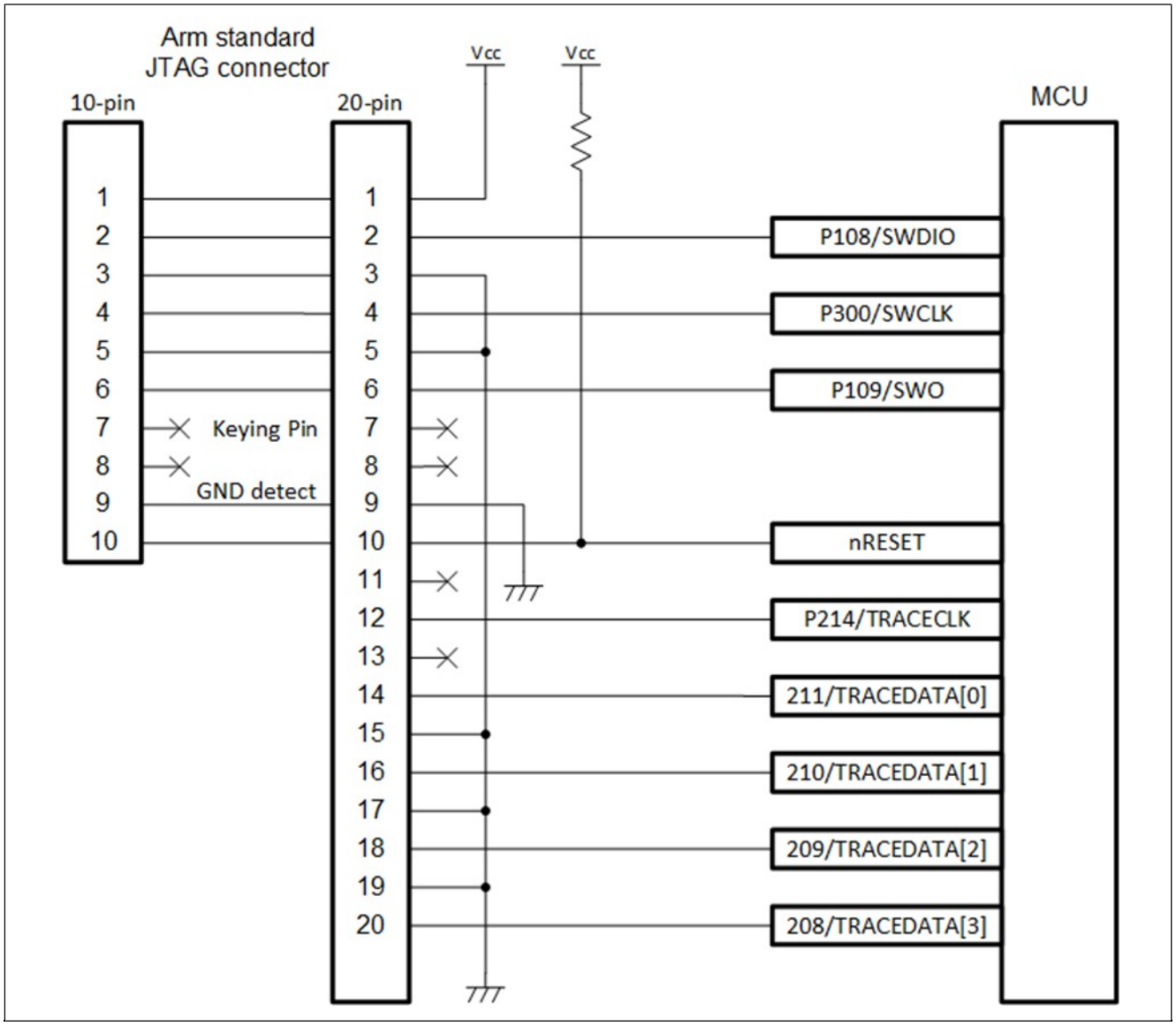

图 1. SWD 接口连接

注：1. 用户系统复位电路的输出必须为集电极开路。

2.2 JTAG 接口

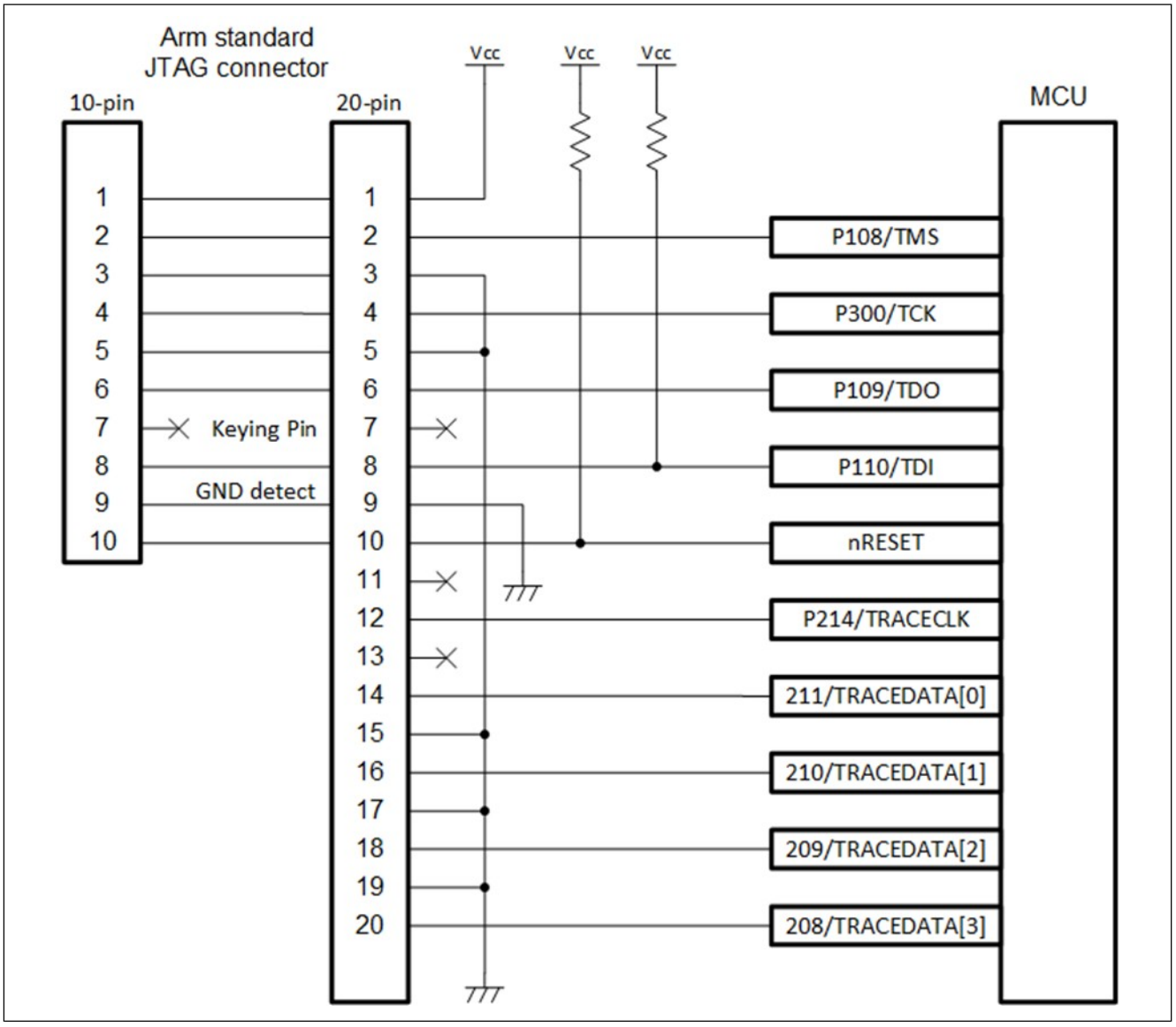

图 2. JTAG 接口连接

注：1. 用户系统复位电路的输出必须为集电极开路。
2. 在 TMS、TCK、TDO 和 TDI 上使用 4.7 K 到 10 K 的上拉电阻。

2.3 使用 SCI 的串行编程接口

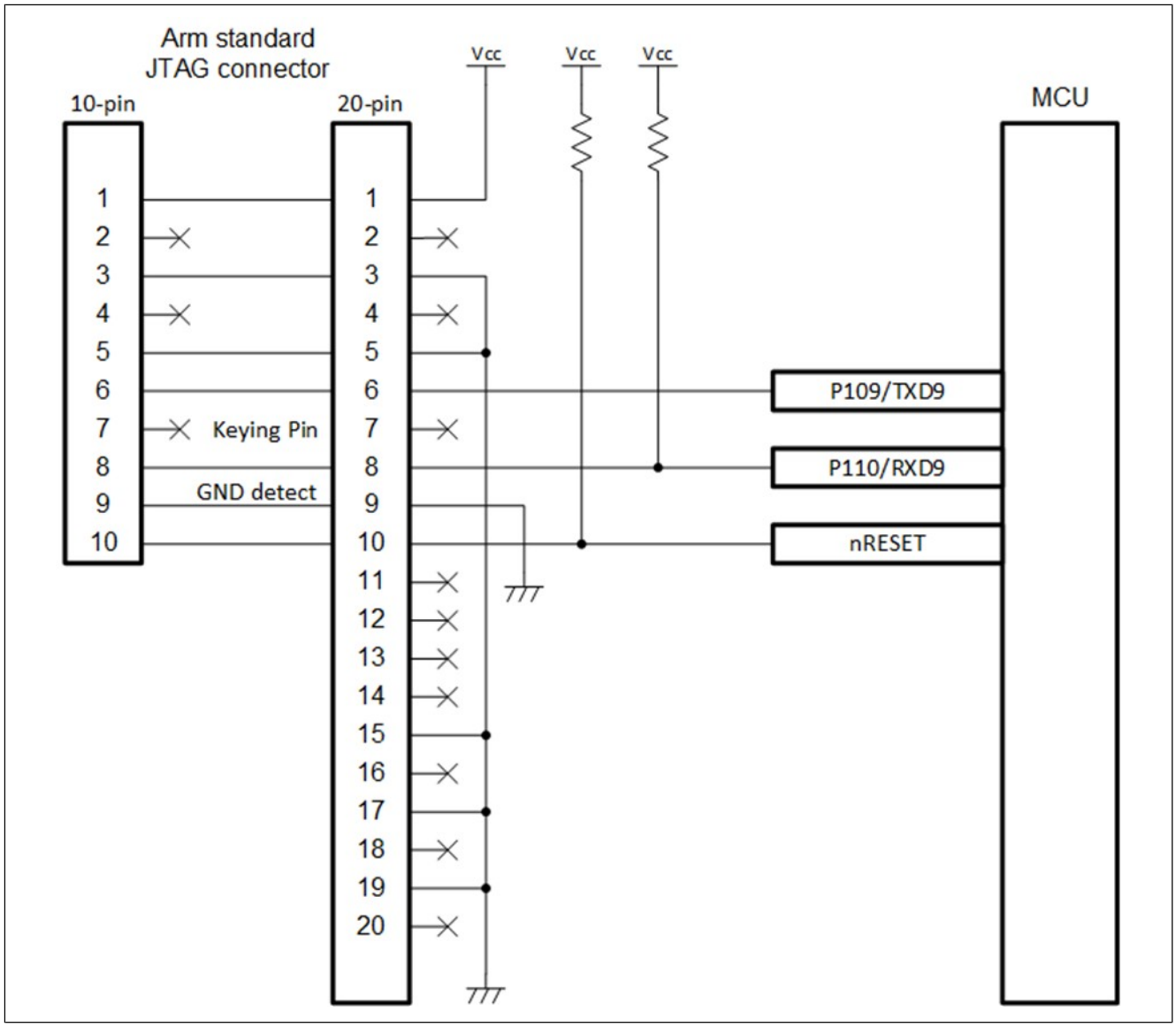

图 3. 使用 SCI 连接的串行编程接口

注：1. 用户系统复位电路的输出必须为集电极开路。

2.4 使用 SCI 的串行编程接口：支持 TrustZone® 的器件

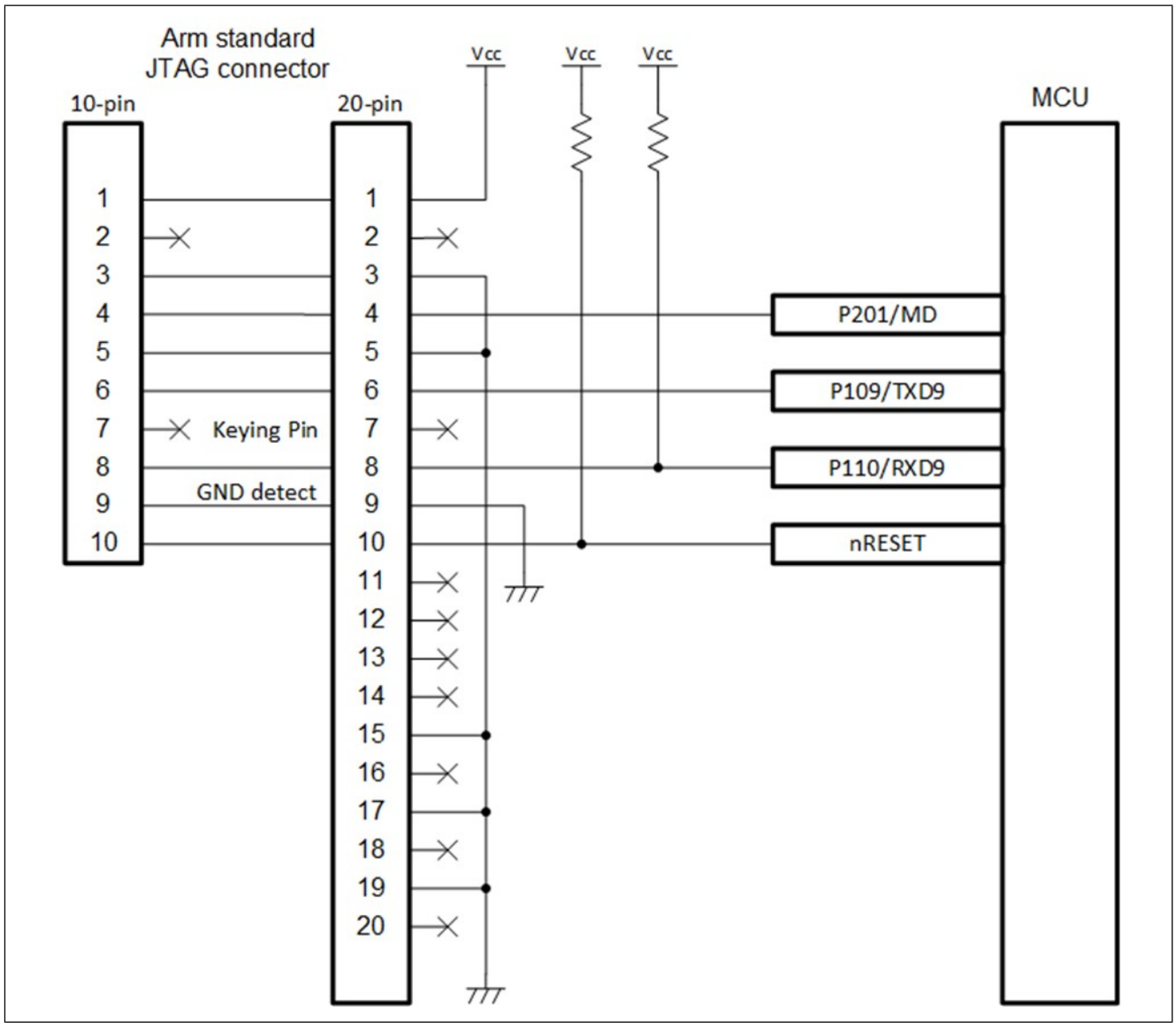

图 4. 使用 SCI 的串行编程接口：支持 TrustZone® 的器件连接

注：1. 用户系统复位电路的输出必须为集电极开路。
2. P201/MD 必须连接到符合 Arm 标准的 JTAG 连接器引脚 4，以支持对 Arm® TrustZone® IDAU 边界寄存器设置进行编程。

2.5 复用仿真器接口

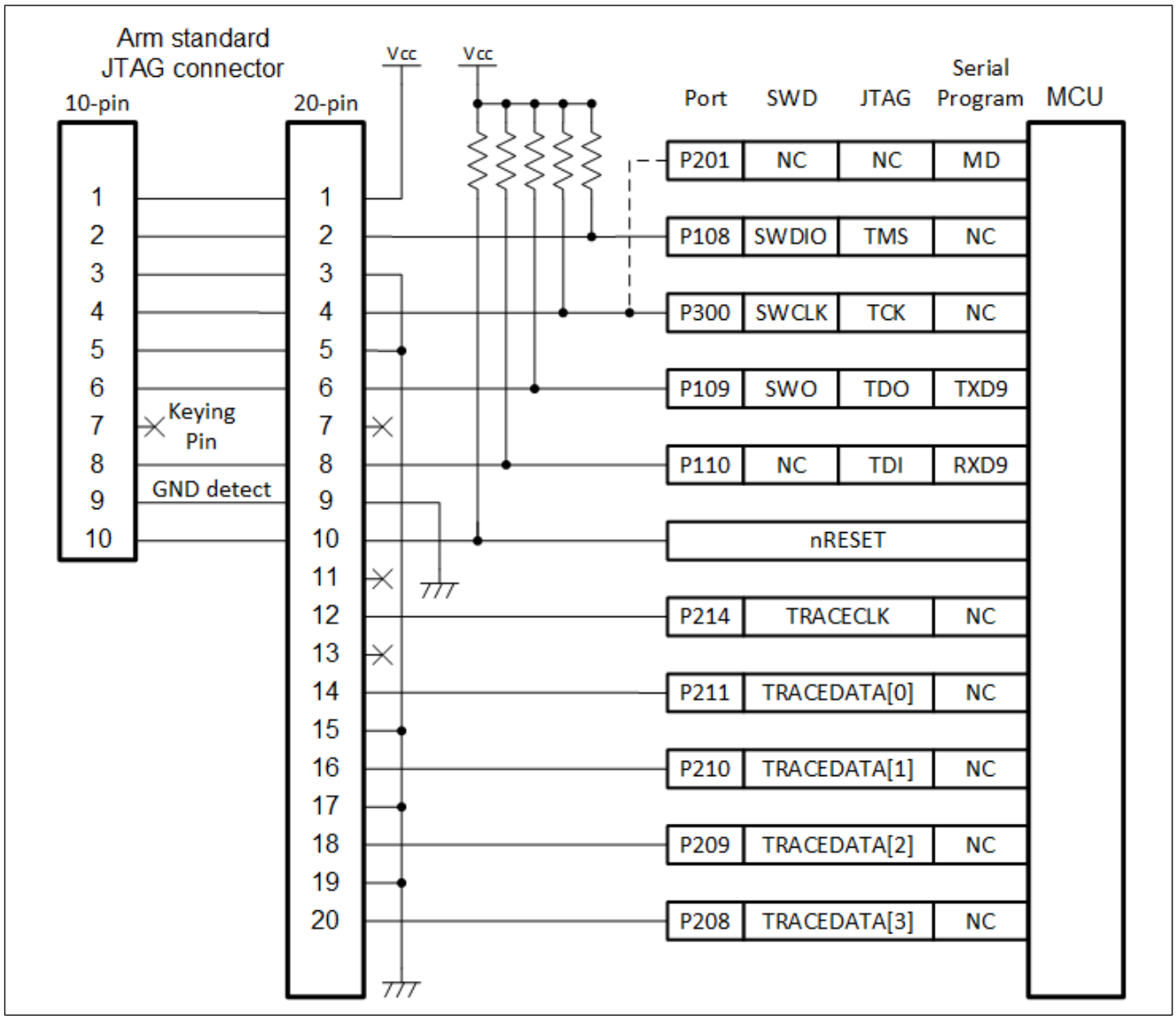

图 5. 复用仿真器接口连接

注：1. 目标器件上的复位电路必须是集电极开路。将 nRESET 信号上拉。请勿在此信号上放置电容，因为这会影响上电复位电路的运行。
2. 在 TMS、TCK、TDO 和 TDI 上使用 4.7 K 到 10 K 的上拉电阻。
3. 要在支持 TrustZone® 的器件上同时使用调试和串行编程，请使用线“或”门电路将 P201/MD 连接到 P300/SWCLK/TCK。

2.6 仿真器连接的软件设置

2.6.1 SWD 和 JTAG 接口

复位后，SWD 和 JTAG 引脚处于默认状态。

表 4. SWD/JTAG 引脚

引脚	P108	P109	P110	P300
功能	TMS/SWDIO	TDO/SWO	TDI	TCK/SWCLK

2.6.2 跟踪端口

在 RA6 器件中，4 位跟踪端口接口单元 (TPIU) 和串行线输出 (SWO) 提供跟踪输出。

必须先使能跟踪端口和时钟，调试器脚本才能使用它们。使用跟踪端口功能时，请避免将跟踪引脚用于其他功能。

表 5. 跟踪端口

引脚	P208	P209	P210	P211	P214
功能	TDATA3	TDATA2	TDATA1	TDATA0	TCLK

有关将 RA6M3 上的跟踪端口与 SEGGER J-Trace Pro 一起使用的示例，请参见以下链接：

https://wiki.segger.com/J-Link_Renesas_RA6M3

也可以在运行时通过使用 Renesas FSP 中的引脚配置器来使能跟踪端口，但是在这种情况下，某些跟踪数据可能会丢失。

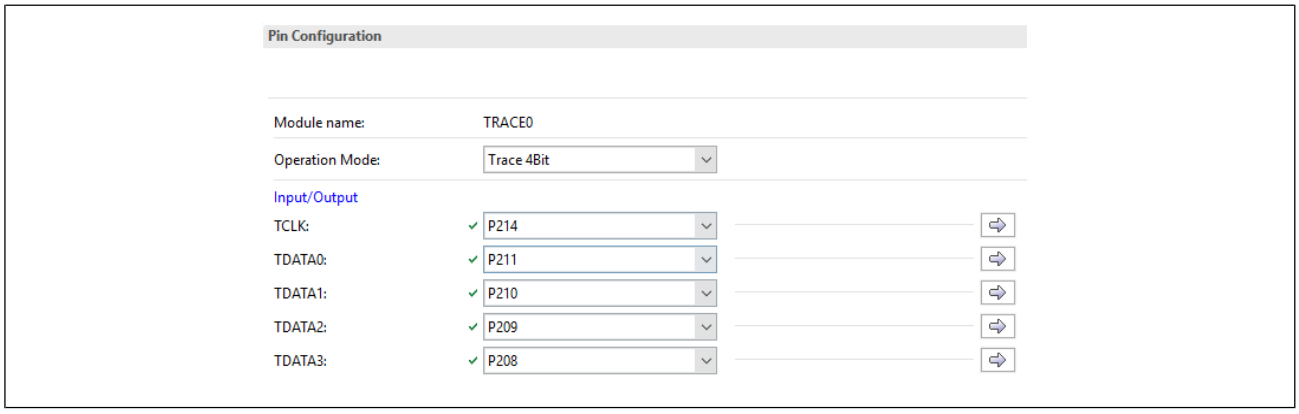

图 6. 在运行时使用 FSP 配置器使能跟踪端口

3. MCU 工作模式

复位后，RA6 MCU 系列可以进入以下两种模式之一：单芯片模式或 SCI/USB 引导模式。引导模式通过 MD 引脚来选择：

表 6. 复位时可用的工作模式

工作模式	MD	片上闪存	外部总线
单芯片模式	1	使能	禁用
SCI/USB 引导模式	0	使能	禁用

通过模式设置引脚确定的工作模式转换。

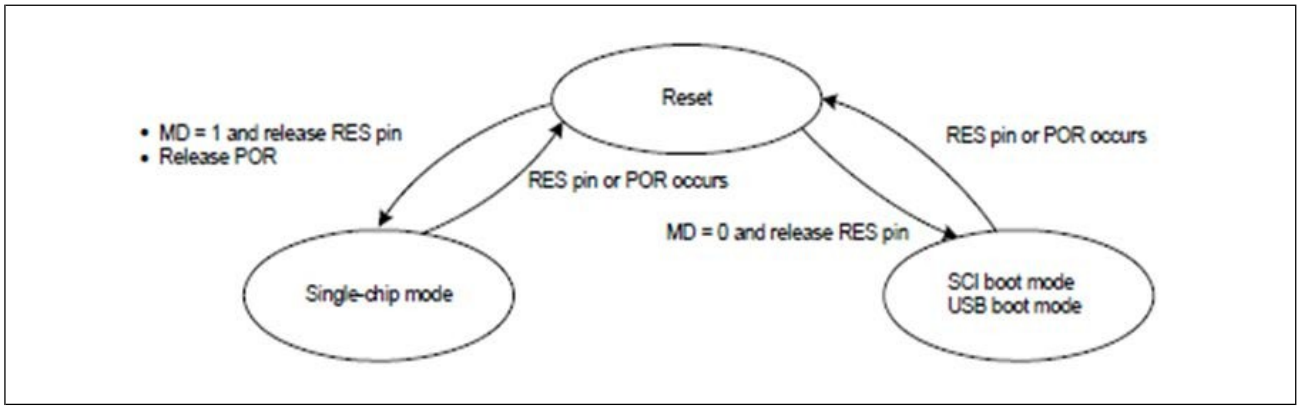

图 7. 模式设置引脚电平和工作模式

典型的 MCU 引导模式电路包括一个跳线和一对电阻器，允许选择将 MD 引脚连接到 VCC 或接地。

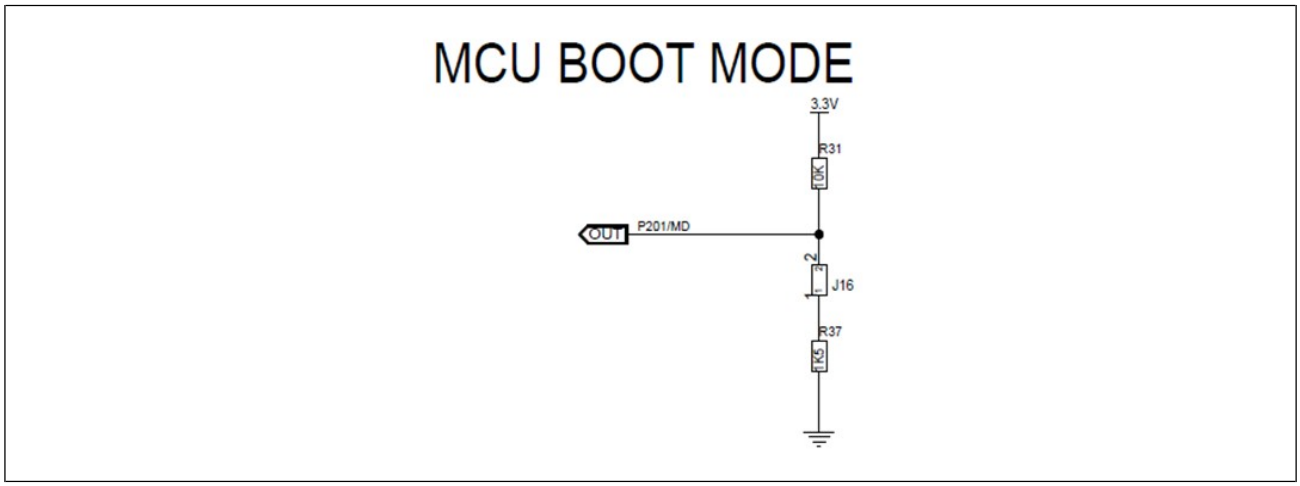

图 8. MCU 引导模式选项的典型电路

4. 选项设置存储器

选项设置存储器用于确定复位后 MCU 的状态。该存储器分配在闪存中的配置设置区域和程序闪存区域。这两个区域的可用设置方法不同。Cortex-M33内核MCU的选项设置存储器可能具有不同的内存占用和分布，有关寄存器的详细介绍，请参见《硬件用户手册》中的“选项设置存储器”一章。

闪存选项寄存器在代码闪存映射中占用一定空间。尽管寄存器位于 RA MCU 上保留闪存的一部分中，但是有些用户可能会无意中将数据存储在这些位置。用户必须进行检查，确保没有将多余数据写入这些位置，否则可能导致芯片发生意外行为。例如，闪存选项寄存器中的设置可能会在复位后立即使能独立看门狗定时器(IWDT)。如果存储在程序 ROM 中的数据无意中与选项设置存储器寄存器重叠，则有可能在不知情的情况下打开 IWDT。这将导致调试器与电路板之间的通信出现问题。

下图显示了基于Cortex-M4内核的RA6M3 MCU的包含选项功能选择寄存器的选项设置存储器。

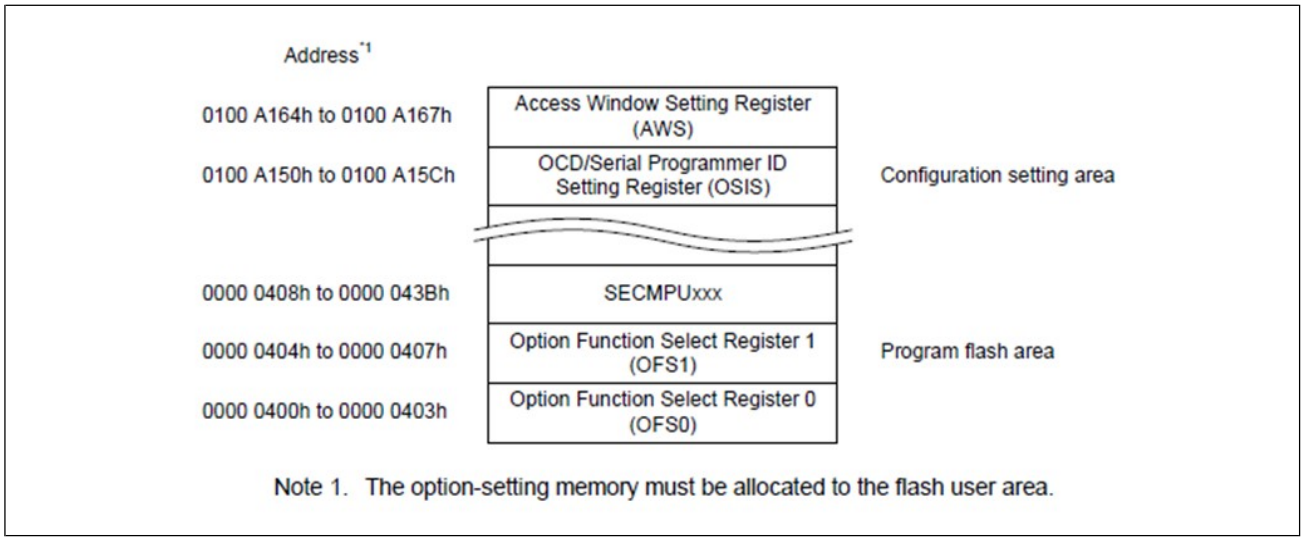

图 9. RA6M3选项功能选择寄存器

4.1 选项设置存储器寄存器

以下是选项设置存储器寄存器的概述。在启动之前，请确保已对其进行正确配置。

- OFS0 寄存器
  — 独立看门狗定时器 (IWDT) 自动启动
  — IWDT 超时、频率、窗口操作、中断类型和低功耗模式行为
  — 看门狗定时器 (WDT) 自动启动
  — WDT 超时、频率、窗口操作和中断类型
- OFS1 寄存器
  — 复位后 LVD0 使能
  — 复位后 HOCO 启动

Renesas FSP 配置器支持在 BSP 设置中设置选项存储器，如下图所示。

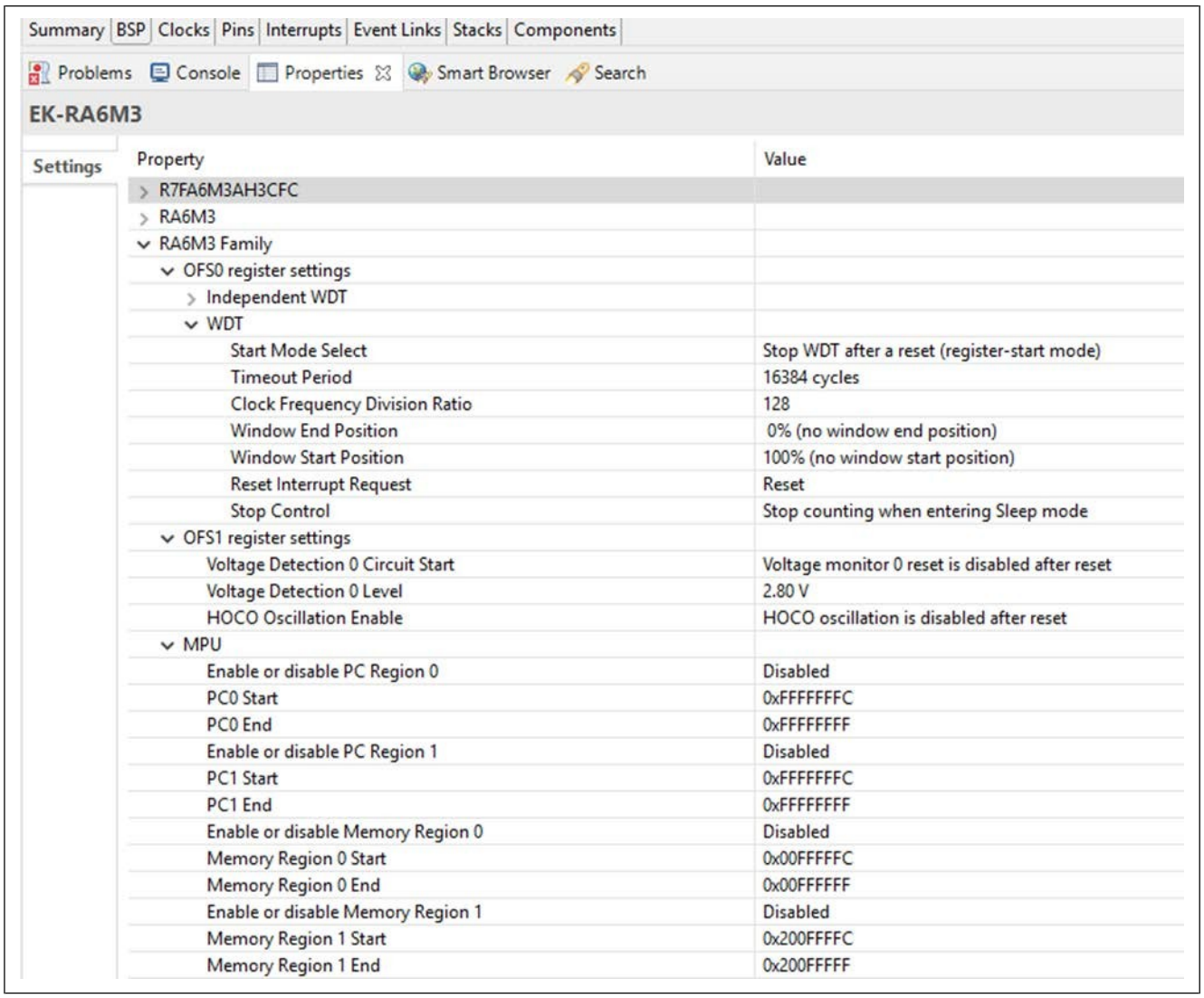

图 10. FSP 配置中的RA6M3选项存储器设置

5. 时钟电路

RA6 MCU 具有六个主振荡器。其中有五个可以用作主系统时钟源。剩下的一个振荡器专用于独立看门狗定时器。在典型系统中，主时钟由外部晶体或时钟驱动。将此输入指向 PLL，并在此经过倍频后变为 PLL 时钟，然后分频为主系统时钟 (ICLK)、闪存时钟、外设模块时钟、外部总线时钟、跟踪时钟和 USB 时钟。此外，ICLK 还用于 CPU 时钟、DMAC 时钟、内部闪存时钟和 SDRAM 时钟。有关时钟生成电路框图的信息，请参见《硬件用户手册》中的“时钟生成电路”一章。

每个时钟都有特定的容差和时序值。有关频率和时钟时序规范的信息，请参见《硬件用户手册》中“电气特性”一章的“交流特性”部分。有关各种时钟频率之间关系的信息，请参见《硬件用户手册》中的“时钟生成电路”一章。

表 7. RA6 振荡器

振荡器	输入源	频率	主要用途
主时钟	外部晶体/谐振器 -或- 外部时钟	8 MHz 至 24 MHz 最高 24 MHz	PLL 输入、主系统时钟、CLKOUT、CAN 时钟、CAC 时钟
子时钟 (SOSC)	外部晶体/谐振器	32.768 kHz	实时时钟、系统时钟（低功耗模式）、CLKOUT、AGT 时钟、CAC 时钟
高速片上振荡器 (HOCO)	片上振荡器	16/18/20 MHz	PLL 输入、主系统时钟、CLKOUT、CAC 时钟
中速片上振荡器 (MOCO)	片上振荡器	8 MHz	主系统时钟、CLKOUT、CAC 时钟
低速片上振荡器 (LOCO)	片上振荡器	32.768 kHz	系统时钟（启动时、低功耗模式下和主振荡器停止检测期间）、AGT 时钟、CAC 时钟、实时时钟
独立看门狗 (IWDT)	片上振荡器	15 kHz	独立看门狗定时器时钟

5.1 复位条件

复位后，RA6 MCU 开始以中速片上振荡器 (MOCO) 作为主时钟源运行。复位时，默认情况下主振荡器和 PLL 处于关闭状态。HOCO 和 IWDT 可能处于打开或关闭状态，具体取决于选项设置存储器中的设置（请参见第 4 节）。

5.2 时钟频率要求

ICLK 必须始终大于或等于 BCLK。最小和最大频率如下表所示。有关详细信息，请参见《MCU 硬件用户手册》中“时钟生成电路”一章的“概述”部分，其中包括外部和内部时钟源规范。更多详细信息，请参见《MCU 硬件用户手册》中“电气特性”一章的“交流特性”部分。

表 8. Arm Cortex-M4 MCU 内部时钟的频率范围

	ICLK [1]	PCLKA [1]	PCLKB	PCLKC	PCLKD
最大频率 [MHz]	120	120	60	60	120
最小频率 [MHz]	—	—	—	—	—

1. 如果使用以太网控制器，则 ICLK 和 PCLKA 频率必须相同且至少为 12.5 MHz

	FCLK [1]	BCLK	UCLK	CANCLK	SDCLK
最大频率 [MHz]	60	120	48	24	120
最小频率 [MHz]	4	—	48	8	—

1. 写入或擦除 ROM 或数据闪存时，FCLK 必须以至少 4 MHz 的频率运行。

表 9. Arm Cortex-M33 MCU 内部时钟的频率范围

	ICLK [1]	PCLKA [1]	PCLKB	PCLKC	PCLKD
最大频率 [MHz]	200	100	50	50	100
最小频率 [MHz]	—	—	—	—	—

如果使用以太网控制器，则 ICLK 和 PCLKA 频率必须相同且至少为 12.5 MHz

	FCLK [1]	BCLK	USBCLK	CANCLK	OCTACLK
最大频率 [MHz]	50	100	48	24	200
最小频率 [MHz]	—	—	48	8	—

写入或擦除 ROM 或数据闪存时，FCLK 必须以至少 4 MHz 的频率运行。

5.2.1 USB 通信要求

RA 系列的某些产品上提供的全速 USB 2.0 模块 (USBFS) 和高速 USB 2.0 模块 (USBHS) 需要 48 MHz 的 USB 时钟信号（UCLK 或 USBCLK）。当使用 USB 并选择 HOCO 作为 PLL 的时钟源时，必须使能锁频环 (FLL) 功能。

使用 USB 时，仅可选择以下主时钟振荡器频率：8 MHz、10 MHz、12 MHz、15 MHz、16 MHz、20 MHz 或 24 MHz。频率的选择取决于 USB 模块所需的时钟生成电路和 48 MHz 时钟中可用的特定分频比。

对于 Arm® Cortex-M4 器件，使用的分频器取决于 SCKDIVCR2 寄存器中 UCK 位的设置。

对于 Arm Cortex-M33 器件，使用的分频器取决于 USBCKDIVCR 寄存器中 USBCKDIV 位的设置。

5.2.2 以太网控制器要求

对于 Arm Cortex-M4 器件，当使用以太网控制器 (EtherC) 和以太网 DMA 控制器 (EDMAC) 时，PCLKA（以太网）必须与 ICLK 相同，并且两者的范围必须均为 12.5 MHz 到 120 MHz。

对于 Arm Cortex-M33 器件，当使用以太网控制器 (EtherC) 和以太网 DMA 控制器 (EDMAC) 时，PCLKH（以太网）的范围必须为 12.5 MHz 到 100 MHz。

5.2.3 ROM 或数据闪存的编程和擦除要求

FCLK 必须至少为 4 MHz 才能在内部 ROM 和数据闪存上执行编程和擦除。

5.2.4 SDRAM 控制器的要求

SDCLK 来源于 BCLK。请勿将 SDCLK 设置为高于系统时钟 (ICLK) 的频率。

5.3 降低时钟生成电路 (CGC) 的功耗

为了帮助节省功耗，请尽可能将所有未使用的时钟（例如 BCLK）的分频器设置为最大可能值。另外，如果不使用时钟，请通过设置适当的寄存器来确保时钟已停止。下表显示了用于控制每个时钟源的寄存器。

表 10. 时钟源配置寄存器

振荡器	寄存器	说明
主时钟	MOSCCR	启动/停止主时钟振荡器
子时钟	SOSCCR	启动/停止子时钟振荡器
高速片上振荡器 (HOCO)	HOCOCR	启动/停止 HOCO
中速片上振荡器 (MOCO)	MOCOCR	启动/停止 MOCO
低速片上振荡器 (LOCO)	LOCOCR	启动/停止 LOCO

5.4 写入系统时钟控制寄存器

写入系统时钟分频控制寄存器 (SCKDIVCR)、系统时钟分频控制寄存器 2 (SCKDIVCR2) 和系统时钟源控制寄存器 (SCKSCR) 中的各个位域时，应格外小心。

当将 SCKDIVCR 或 SCKDIVCR2 中的任何值从较低的分频比更改为较高的分频比时，请至少等待 750 ns，然后再更改该值。当将任何值从高分频比更改为低分频比时，请在更改值后至少等待 250 ns，然后再开始后续处理。

当将 SCKSCR 的值从 PLL 更改为其他时钟源时，请至少等待 750 ns，然后再更改该值。当将值从非 PLL 时钟源更改为 PLL 时，请在更改值后至少等待 250 ns，然后再开始后续处理。

测量等待时间的推荐方法是在软件中通过对指令周期进行计数来实现。确保考虑情况最差的使用条件，以确保会经过所需的等待时间。

5.5 时钟设置示例

Renesas FSP 为 RA6 MCU 提供了一个简单的可视化时钟配置工具，如下所示。

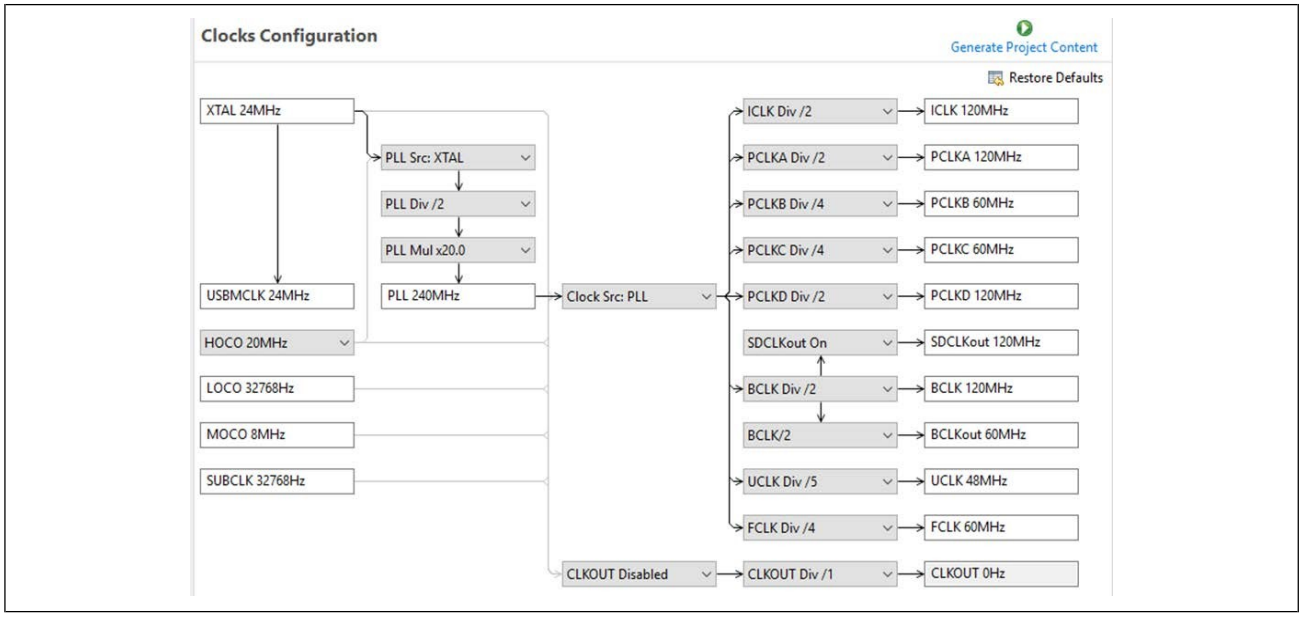

图 11. 使用 Renesas FSP 配置器进行时钟设置

5.6 HOCO 精度

内部高速片上振荡器 (HOCO) 的运行频率为 16 MHz、18 MHz 或 20 MHz，精度为 +/-2% 或更高。可以通过使能锁频环 (FLL) 功能来提高 HOCO 的精度，这可将时钟精度提高到 +/-0.3% 或更高。有关详细信息，请参见硬件手册中的“电气规范”章节。

HOCO 可以用作 PLL 电路的输入。当以这种方式使用 HOCO 时，不需要外部振荡器。当因空间限制或其他限制而需要减少 PCB 设计中的元件数量时，这可能是一个优势。不过，此时会因时钟精度问题而产生性能影响和限制，因此应针对您的应用进行评估。

5.7 闪存接口时钟

对内部闪存（ROM 和 数据闪存）进行编程和擦除操作以及从数据闪存读取数据时，闪存接口时钟 (FCLK) 用作工作时钟。因此，FCLK 的频率设置会直接影响从数据闪存读取数据所需的时间。如果用户的程序正在从数据闪存中读取数据，或者正在对内部闪存执行编程或擦除操作，则建议使用最大 FCLK 频率。

请注意，FCLK 频率对读取ROM 或对 RAM 进行读写操作没有任何影响。

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，请参见《硬件用户手册》中“时钟生成电路 (CGC)”一章的“使用注意事项”部分。

通常，晶体谐振器及其负载电容应尽可能靠近 MCU 时钟引脚（XTAL/EXTAL、XCIN/XCOUT）放置。避免在晶体谐振器和 MCU 之间连接任何其他信号走线。尽量减少每条走线上使用的连接通孔数量。

5.9 外部晶体谐振器选择

外部晶体谐振器可以用作主时钟源。外部晶体谐振器可跨 MCU 的 EXTAL 和 XTAL 引脚连接。外部晶体谐振器的频率必须处于主时钟振荡器的频率范围内。

晶体谐振器的选择在很大程度上取决于各个独特的电路板设计。由于适合与 RA6 MCU 器件配合使用的可用晶体谐振器的选择可能很多，因此请仔细评估所选晶体谐振器的电气特性，以确定具体的实现要求。

下图给出了典型的晶体谐振器连接示例。

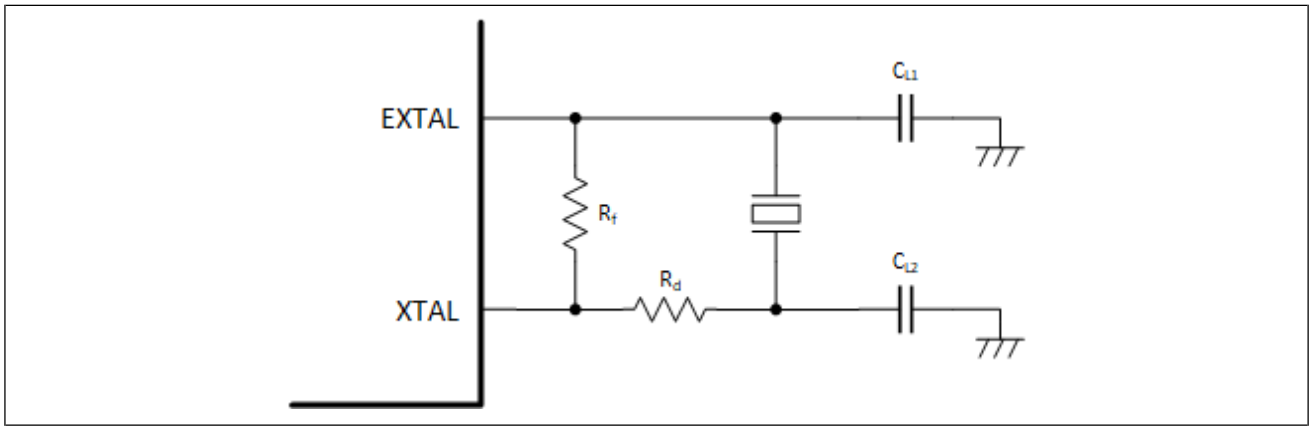

图 12. 晶体谐振器连接示例

选择晶体谐振器和相关电容时，必须仔细评估。如果晶体谐振器制造商有相关建议，可以添加外部反馈电阻 (R_f) 和阻尼电阻 (R_d)。

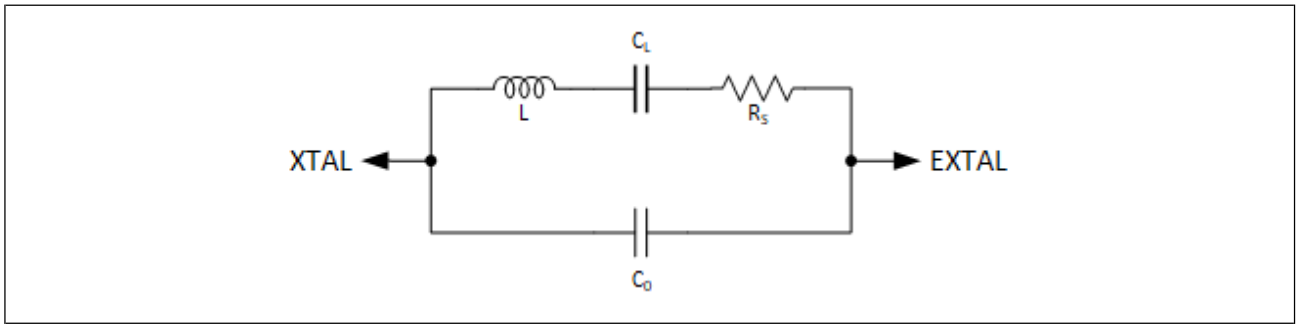

图 13. 晶体谐振器的等效电路

CL1 和 CL2 的电容值选择会影响内部时钟的精确度。要了解 CL1 和 CL2 值的影响，应使用上图中晶体谐振器的等效电路来模拟该电路。为了获得更准确的结果，还应考虑与晶体谐振器元件之间的布线相关的杂散电容。

6. 复位要求和复位电路

共有十四种类型的复位。这些复位在 Arm® Cortex-M4 器件和 Arm Cortex-M33 器件之间略有不同。

表 11. Arm Cortex-M4 器件复位

复位名称	源
引脚复位	RES# 电平被拉低
上电复位	VCC 升高（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
看门狗定时器复位	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 下降（电压检测 Vdet1）
电压监视器 2 复位	VCC 下降（电压检测 Vdet2）
SRAM 奇偶校验错误复位	SRAM 奇偶校验错误检测
SRAM ECC 错误复位	SRAM ECC 错误检测
总线主 MPU 错误复位	总线主 MPU 错误检测
总线从 MPU 错误复位	总线从 MPU 错误检测
堆栈指针错误复位	堆栈指针错误检测
深度软件待机复位	通过中断信号退出深度软件待机模式
软件复位	寄存器设置

表 12. Arm® Cortex-M33 器件复位

复位名称	源
引脚复位	RES# 电平被拉低
上电复位	VCC 升高（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
看门狗定时器复位	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 下降（电压检测 Vdet1）
电压监视器 2 复位	VCC 下降（电压检测 Vdet2）
SRAM 奇偶校验错误复位	SRAM 奇偶校验错误检测
SRAM ECC 错误复位	SRAM ECC 错误检测
总线主 MPU 错误复位	总线主 MPU 错误检测
TrustZone 错误复位	TrustZone 错误检测
缓存奇偶校验错误复位	缓存奇偶校验错误检测
深度软件待机复位	通过中断信号退出深度软件待机模式
软件复位	寄存器设置

6.1 引脚复位

当 RES# 引脚电平被拉低时，所有处理都将中止，MCU 进入复位状态。要在运行中复位 MCU，应在指定的复位脉冲宽度内将 RES# 保持为低电平。有关时序要求的更详细信息，请参见《硬件用户手册》中“电气特性”一章的“复位时序”部分。另请参见本文档的第 2 节“仿真器支持”，了解与调试支持相关的复位电路的详细信息。

无需在 RES# 线路上使用外部电容，因为 POR 电路在内部将其保持为低电平以实现良好的复位，并且需要最小的复位脉冲来启动此过程。

6.2 上电复位

有两种情况会产生上电复位 (POR)：

1. 如果 RES# 引脚在接通电源后处于高电平状态。
2. 如果 RES# 引脚在 VCC 低于 V_{POR} 时处于高电平状态。

在 VCC 超过上电复位电压 (V_{POR}) 并经过上电复位时间 (t_{POR}) 之后，芯片将从上电复位状态释放。上电复位时间是允许外部电源和 MCU 达到稳定状态的时间。有关电压大小和时序的详细信息，请参见《硬件用户手册》中“电气特性”一章的“POR 和 LVD 特性”部分。

由于 POR 电路依赖于 RES# 与 VCC 同时为高电平，因此请勿在复位引脚上放置电容。这将减慢 RES# 相对于 VCC 的上升时间，从而妨碍 POR 电路正确识别上电条件。

当电源 (VCC) 降至不超过 V_{POR} 时，如果 RES# 引脚为高电平，则会产生上电复位。在 VCC 上升到 V_{POR} 以上并且经过 t_{POR} 之后，芯片将从上电状态释放。

上电复位后，RSTSR0 中的 PORF 位置 1。引脚复位后，PORF 清零。

6.3 独立看门狗定时器复位

这是由独立看门狗定时器 (IWDT) 产生的内部复位。

当 IWDT 下溢时，可以选择产生独立看门狗定时器复位（可以改为产生 NMI），并且 RSTSR1 中的 IWDTRF 位置 1。短暂延迟（通常为 320 µs）后，将退出 IWDT 复位。

6.4 看门狗定时器复位

这是看门狗定时器 (WDT) 产生的内部复位。

当 WDT 下溢时，可以选择产生看门狗定时器复位（可以改为产生 NMI），并且 RSTSR1 中的 WDTRF 位置 1。短暂延迟（通常为 320 µs）后，将退出 WDT 复位。

6.5 电压监视复位

RA6 系列包括在欠压期允许MCU防止不安全操作的电路。板上比较器根据三个参考电压 V_{det0}、V_{det1} 和 V_{det2} 检查电源电压。当电源下降到每个参考电压以下时，会产生中断或复位。检测电压 V_{det0}、V_{det1} 和 V_{det2} 均可从 3 个不同大小的值中选择。

当 Vcc 随后上升到超过 V_{det0}、V_{det1} 或 V_{det2} 时，经过稳定时间后，电压监视复位释放将继续。

上电复位后，将禁用低电压检测。可以通过使用选项功能寄存器 OFS1 来使能电压监视。有关更多详细信息，请参见《硬件用户手册》中的“低电压检测 (LVD)”一章。

LVD 复位后，RSTSR0 中的 LVDnRF（n = 0、1、2）位置 1。

6.6 深度软件待机复位

这是在通过中断信号退出深度软件待机模式时产生的内部复位。

当退出深度软件待机模式时，将产生深度软件待机复位，并且时钟振荡开始。接收到中断后，经过深度待机退出等待时间（tDSBYWT 34 - 35 个时钟周期）后，将退出复位并开始正常处理。有关深度软件待机模式的详细信息，请参见《硬件用户手册》中的“低功耗模式”一章。

深度软件待机复位后，RSTSR0 中的 DPSRSTF 位置 1。

6.7 软件复位

这是通过将 0xA501 写入 SWRR 寄存器产生的内部复位。使用软件复位时，内部复位时间最长为 960 µs。使用软件复位时，在发出软件复位命令之前，请确保先喂狗。

产生软件复位后，RSTSR1 中的 SWRF 引脚置 1。短暂的延迟（通常为 320 µs）后，将退出软件复位。

6.8 其他复位

MCU 内的大多数外设功能都可以在特定的故障条件下产生复位。无需硬件配置即可使能这些复位。有关将为每个外设功能产生复位的条件的详细信息，请参见《硬件用户手册》中的相关章节。

6.9 冷/热启动的确定

借助 RA6 MCU，用户可以确定发生复位过程的原因。RSTSR2 中的 CWSF 标志指示是上电复位导致了复位过程（冷启动），还是操作期间输入的复位信号导致了复位过程（热启动）。

发生上电复位时，该标志置 0。否则，该标志不会置 0。通过软件向该标志写入 1 时会将其置 1。即使在写入 0 时也不会将其置 0。

6.10 确定复位源

借助 RA6 MCU，用户可以确定复位信号产生源。读取 RSTSR0、RSTSR1 和 RSTSR0，以确定哪个复位是复位源。有关流程图，请参见《硬件用户手册》中的“复位产生源的确定”部分。

以下代码示例展示了如何使用 Renesas FSP 中基于 CMSIS 的寄存器结构确定复位是由软件复位、深度软件待机还是上电复位导致的。

```
/* Deep Software Standby Reset */
if(1 == R_SYSTEM->RSTSR0_b.DPSRSTF)
{
  /* Do something */
}

/* Power on Reset */
if(1 == R_SYSTEM->RSTSR0_b.PORF)
{
  /* Do something */
}

/* Software Reset */
if(1 == R_SYSTEM->RSTSR1_b.SWRF)
{
  /* Do something */
}
```

7. 使用TrustZone®

7.1 Arm® TrustZone® 技术的实现

某些 RA6 MCU（如RA6M4和RA6M5）包含 Arm® TrustZone® (TZ) 安全功能。有关 TZ 实现的完整详细信息，请参见 Arm 文档 (https://developer.arm.com/ip-products/security-ip/trustzone) 或相应的《硬件用户手册》。

Arm TZ 技术将 MCU 划分为安全和非安全分区，因此应用程序也分为安全和非安全应用程序。安全应用程序可以访问安全和非安全存储区和资源。非安全代码访问非安全存储器和资源，也可以通过位于非安全可调用 (NSC) 区域中的一组跳板（Veneer）来访问安全资源。当从非安全分区中调用安全代码时，跳板可以确保从单个接入点访问安全代码。默认情况下，MCU 在安全分区中启动。CPU 的安全状态可以是“安全”或“非安全”。MCU 代码闪存、数据闪存和 SRAM 分为安全 (S) 和非安全 (NS) 区域。代码闪存和 SRAM 还包含一个称为非安全可调用 (NSC) 的区域。当器件生命周期为安全软件调试 (SSD) 状态时，可通过 SCI 或 USB 引导模式命令将这些存储器安全属性设置到非易失性存储器中。在应用程序执行之前，存储器安全属性就会被加载到实现定义属性单元 (IDAU) 外设和存储器控制器中，并且无法通过应用程序代码进行更新。

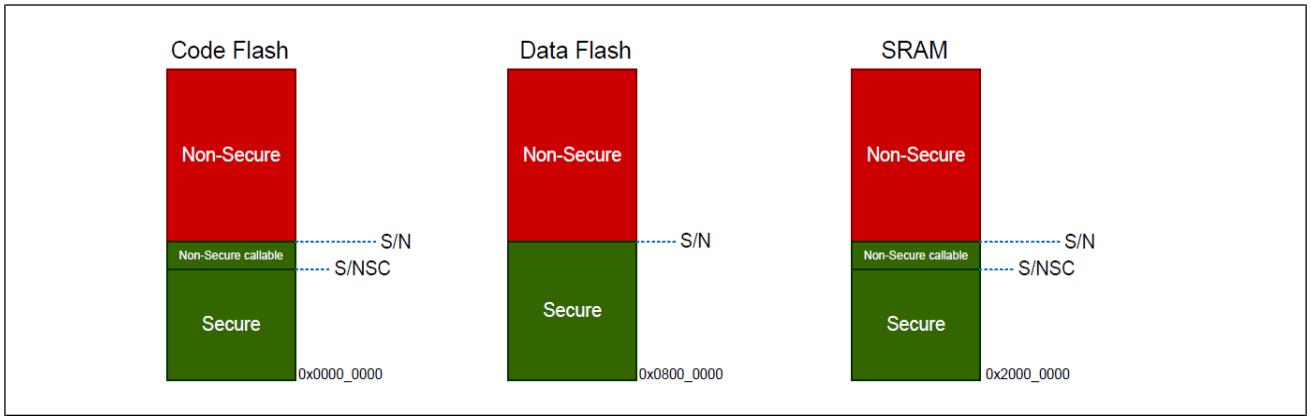

图 14. 安全和非安全区域

注：所有外部存储器访问均被视为不安全访问。

代码闪存和 SRAM 可以分为安全、非安全和非安全可调用三个区域。所有来自非安全区域的安全存储器访问均必须通过非安全可调用网关，并以特定的安全网关 (SG) 汇编器指令为目标。这将强制访问固定位置的安全 API，并防止调用子功能等。未以 SG 指令为目标将产生 TZ 异常。支持 TZ 的编译器将使用 CMSE 扩展自动管理 NSC 模板的产生。

参考文档：

- RA Arm® TrustZone® 工具入门
- 使用 Arm® TrustZone® 保护静态数据
- Arm® TrustZone® 的安全设计 - IP 保护
- 使用 SCE9 和 Arm® TrustZone® 建立和保护器件标识
- 器件生命周期管理密钥安装
- 相关的 《MCU 硬件用户手册》中的“Arm TrustZone 安全性”部分。

7.2 支持 TrustZone 的仿真器

Renesas 提供的仿真器既支持使用 SWD 或 JTAG 通信进行调试，也支持使用 SCI 通信进行串行编程。借助该仿真器，可以轻松地在调试和串行编程之间进行切换。下表显示了使用此仿真器时 10 引脚或 20 引脚插座的引脚排列。SWD 和 JTAG 的引脚排列符合 Arm® 标准，并且添加了 MD、TXD、RXD 引脚，以便使用 SCI 通信进行串行编程。

必须使用串行编程接口对 TrustZone IDAU 边界寄存器设置进行编程。建议使用板上线“或”电路连接 P300/SWCLK/TCK 和 P201/MD 引脚，以同时使用调试和串行编程。

表 13. 仿真器的引脚分配

引脚编号	SWD	JTAG	使用 SCI 的串行编程
1	VCC	VCC	VCC
2	P108/SWDIO	P108/SWDIO	NC
4	P300/SWCLK 以线“或”方式连接 P201/MD	P300/TCK 以线“或”方式连接 P201/MD	P201/MD
6	P109/SWO/TXD9	P109/TDO/TXD9	P109/TDO/TXD9
8	P110/SWO/RXD9	P110/TDI/RXD9	P110/TDI/RXD9
9	GNDdetect	GNDdetect	GNDdetect
10	nRESET	nRESET	nRESET
12	P214/TCLK	P214/TCLK	NC
14	P211/TDATA[0]	P211/TDATA[0]	NC
16	P210/TDATA[1]	P210/TDATA[1]	NC
18	P209/TDATA[2]	P209/TDATA[2]	NC
20	P208/TDATA[3]	P208/TDATA[3]	NC
3、5、15、17、19	GND	GND	GND
7	NC	NC	NC
11、13	NC	NC	NC

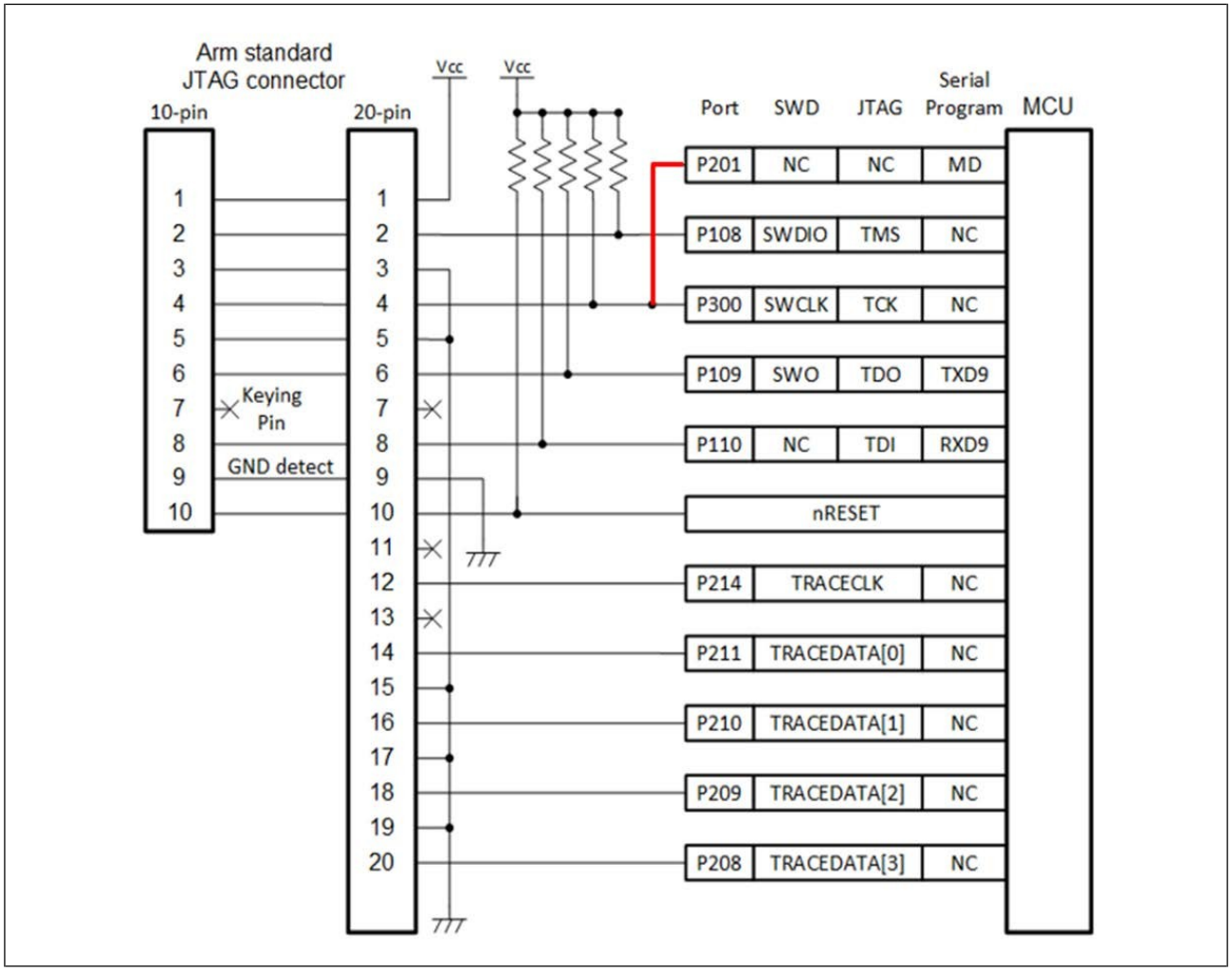

图 15. 支持 Trustzone 的 MCU 的仿真器连接

8. 存储器

RA6 MCU 支持 4 GB 的线性地址空间，范围为 0000 0000h 到 FFFF FFFFh，其中包含程序、数据和外部存储器总线。该系列的某些产品包括一个 SDRAM 控制器，可利用该控制器访问连接到外部存储器总线的 SDRAM 器件。程序和数据存储器共用地址空间；使用单独的总线分别访问这两个存储器，从而提高性能并允许在同一个周期访问程序和数据。存储器映射中包含片上 RAM、外设 I/O 寄存器、程序 ROM、数据闪存和外部存储器区域。

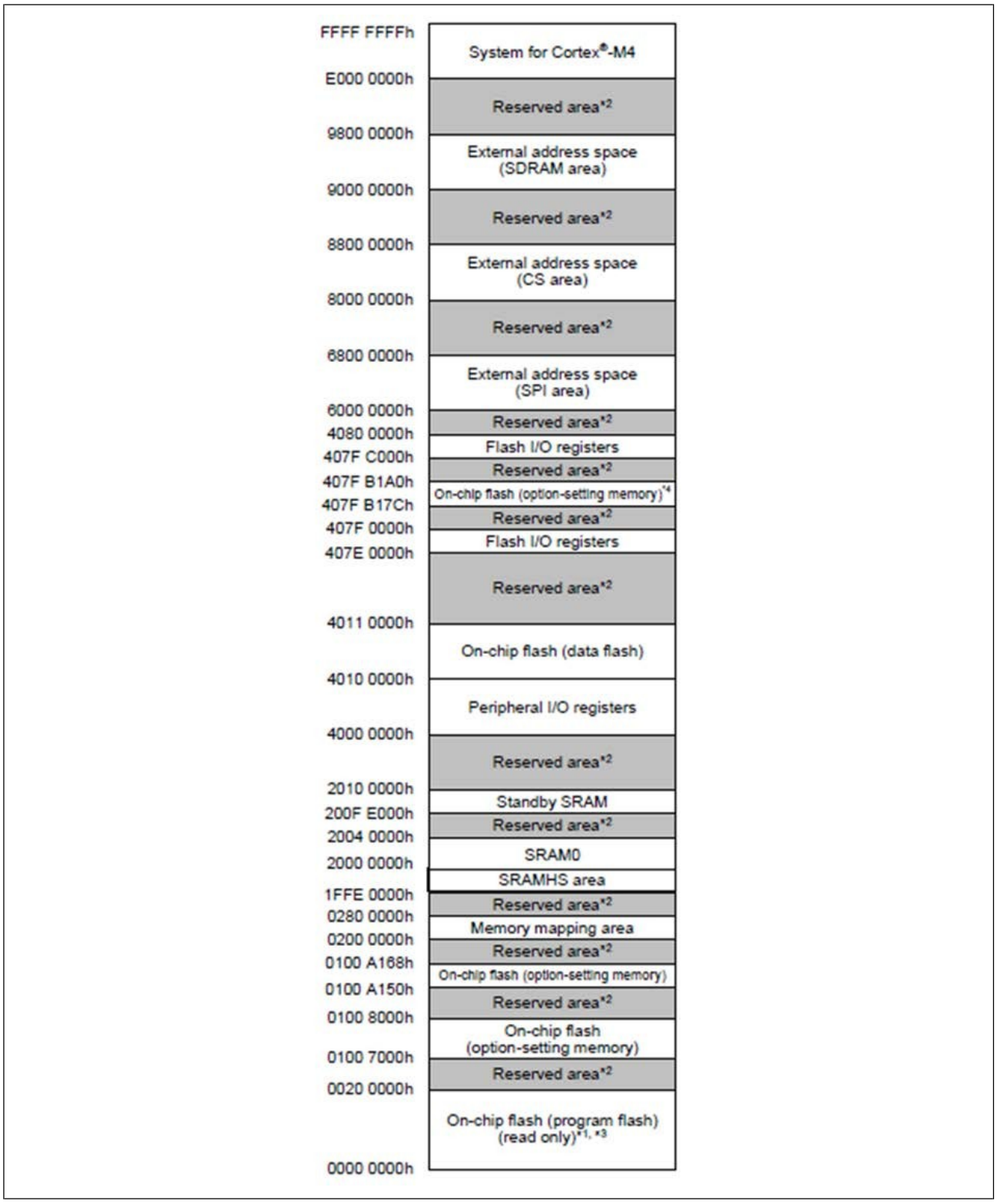

图 16. RA6M3 存储器映射

8.1 SRAM

RA6 MCU 提供带奇偶校验位或 ECC（纠错码）的片上高速 SRAM 模块。SRAM0 的前 32 KB 区域受 ECC 控制。奇偶校验在其他区域执行。下表列出了 SRAM 规范。SRAM 模块的数量和容量因器件而异。有关详细信息，请参见《硬件用户手册》。

Table 53.1 SRAM specifications

Parameter	Specifications without ECC	Specifications with ECC	SRAMHS specifications
SRAM capacity	SRAM0: 224 KB SRAM1: 256 KB	SRAM0 (ECC area): 32 KB	SRAMHS: 128 KB
SRAM addresses*2	SRAM0: 2000 8000h to 2003 FFFFh SRAM1: 2004 0000h to 2007 FFFFh	SRAM0 (ECC area): 2000 0000h to 2000 7FFFh	SRAMHS: 1FFE 0000h to 1FFF FFFFh
Access*1	Wait states are inserted into the read cycle by default. If the ICLK frequency is faster than 60 (up to 120) MHz, a wait state is required. If the ICLK frequency is 60 MHz or less, a wait state is not required. For details, see section 53.4, Usage Notes.	Wait states are inserted into the read cycle by default. If the ICLK frequency is faster than 60 (up to 120) MHz, wait state is required. If the ICLK frequency is 60 MHz or less, a wait state is not required. For details, see section 53.4, Usage Notes.	Access to the SRAMHS is always no wait state.
Data retention	Not available in Deep Software Standby mode		
Module-stop function	Module-stop state can be set to reduce power consumption		
Parity	Even-parity (data: 8 bits, parity: 1 bit)	No parity	Even-parity (data: 8 bits, parity: 1 bit)
Error checking	Even-parity error check	Detection up to 2-bit errors	Even-parity error check

Note 1. For details, see section 53.3.7, Access Cycles.
Note 2. The Cortex®-M4 processor supports Arm®v7 unaligned accesses. In this product, SRAMHS and SRAM0 are adjacent to each other and there is an access boundary between them. Therefore, when SRAMHS and SRAM0 are used as a continuous area of memory space, access that straddles the boundary must not be produced as this might lead to access of data other than what is intended. For details, see the *ARM® Cortex®-M4 Processor Technical Reference Manual*.

图 17. RA6M3 SRAM 规范示例

8.2 待机SRAM

RA6 MCU 提供了一个片上 SRAM，支持在深度软件待机模式下保留数据。下表列出了备用 SRAM 规范。

在深度软件待机模式下，待机 SRAM 的电源由 DPSBYCR.DEEPCUT[1:0] 位使能。如果 DPSBYCR.DEEPCUT[1:0] 位设置为 00b，则将在深度软件待机模式下保留待机 SRAM 中的数据。有关 DPSBYCR.DEEPCUT[1:0] 位的详细信息，请参见第 11 节“低功耗模式”。

Table 54.1 Standby SRAM specifications

Parameter	Specifications
SRAM capacity	8 KB
SRAM address	200F E000h to 200F FFFFh
Access	The number of access depends on the frequency between ICLK and PCLKB. See section 54.2.4, Access Cycle for details.
Data retention	Data can be retained in Deep Software Standby mode
Parity	Even parity (data: 8 bits, parity: 1 bit)
Module-stop function	Module-stop state can be set to reduce power consumption

图 18. RA6M3 待机 SRAM 规范示例

Renesas FSP 中的 LPM（低功耗模式）驱动程序提供了一个用于切断或保持待机 SRAM 电源的选项，如下图所示。仍然需要调用 LPM 驱动程序的 API 才能将所选设置写入 MCU 寄存器。

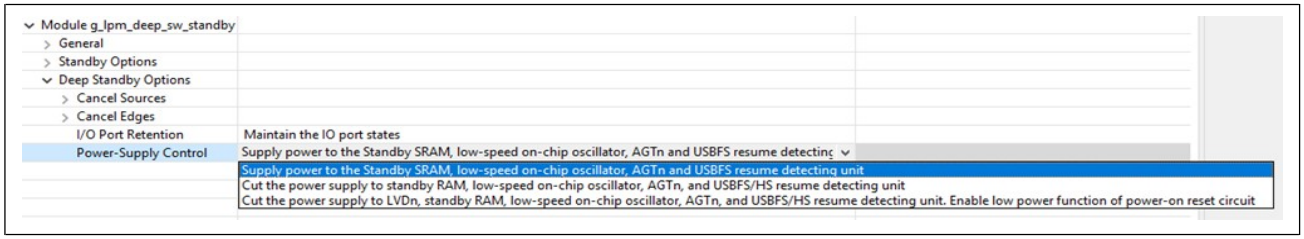

图 19. 使用 Renesas FSP 配置器使能/禁用待机 SRAM 的电源

8.3 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工作模式。大多数外设 I/O 寄存器占用的地址范围为 4000 0000h 到 400F FFFFh。但是，位置和大小可能会因器件而异。有关详细信息，请参见《硬件用户手册》。有关详细信息，可参见“I/O 寄存器”附录，也可参见每个外设功能的寄存器说明。该区域包含在所有工作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 407E FFFFh 和 407F C000h 至 407F FFFFh。

Renesas 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。

8.4 片上闪存

RA6 MCU 具有两部分闪存：代码闪存和数据闪存，各部分的大小和擦写周期数因器件而异。闪存控制单元 (FCU) 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需要不时更新的信息，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远小于代码闪存的单位（数据闪存为 2 字节，而代码闪存为 128 字节）。

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。这样，无需连接外部编程工具即可完成现场固件更新。

Renesas FSP 为代码闪存和数据闪存提供了 HAL 驱动程序。

下图给出了代码闪存和数据闪存的规范示例。

Table 55.1 Specifications of the code flash memory and data flash memory

Parameter	Code flash memory specifications	Data flash memory specifications
Memory capacity	• Up to 2 MB	64 KB
Read cycle	• 80 MHz < ICLK frequency ≤ 120 MHz: Cache hit: 1 cycle Cache miss: 3 cycles • 40 MHz < ICLK frequency ≤ 80 MHz: Cache hit: 1 cycle Cache miss: 2 cycles • ICLK frequency ≤ 40 MHz: Cache hit: 1 cycle Cache miss: 1 cycle	A read operation takes seven cycles of FCLK in words or bytes (FCLK frequency is up to 60 MHz)
Value after erasure	FFh	Undefined
Programming/erasing methods	• Programming and erasing of code and data flash memory handled by FACI commands specified in the FACI command issuing area (407E 0000h) • Programming by dedicated flash-memory programmer transfer through a serial interface (serial programming) • Programming of flash memory by user program (self-programming)	
Security function	Protection against illicit tampering with or reading of data in flash memory	
Protection	Protection against erroneous overwriting of flash memory	
Background operations (BGOs)	• Code flash memory can be read during code flash memory programming*1 • Code flash memory can be read during data flash memory programming • Data flash memory can be read during code flash memory programming	
Units of programming and erasure	• 128-byte units for programming in user area • Block units for erasure in user area	• 4/8/16-byte units for programming in data area • 64/128/256-byte units for erasure in data area
Other functions	Interrupts can be accepted during self-programming	
	An expansion area of flash memory (option bytes) can be set in the initial MCU settings	
On-board programming (four types)	Programming in serial programming mode (SCI boot mode): • Asynchronous serial interface (SCI9) used • Transfer rate adjusted automatically Programming in serial programming mode (USB boot mode): • USBFS used • Dedicated hardware not required, so direct connection to PC is possible Programming in On-chip debug mode: • JTAG or SWD interface used • Dedicated hardware not required Programming by a routine for code and data flash memory programming within the user program: • Allows code and data flash memory programming without resetting the system	

图 20. RA6M3 MCU 上的代码闪存和数据闪存规范

注：代码闪存在擦除状态下为 FFh，但数据闪存在擦除状态下为未定义值。

8.4.1 后台运行

RA6 MCU 支持通过后台运行访问代码闪存和数据闪存。这意味着当执行编程或擦除操作时，用户可以继续从正在操作的存储区以外的其他存储区中执行程序和访问数据。例如，在对数据闪存进行擦除或编程操作时，CPU 可以从代码闪存执行应用程序代码。同样，在对代码闪存进行擦除或编程操作时，CPU 可以从 SRAM 执行应用程序代码。这项规则的唯一例外是，在对代码闪存进行编程或擦除期间无法访问数据闪存。

8.4.2 ID 代码保护

具有 Cortex-M4 内核的 RA6 MCU 在选项设置存储区中有一个用作 ID 代码的 128 位存储器。如果此 ID 代码留空 (0xFF)，则不实现任何保护功能。此时可以通过引导模式或使用片上调试器来访问 MCU。如果设置了 ID 代码，则用户可以控制对这些模式的访问。用户可以选择始终禁止连接，或者可以选择在输入匹配的 ID 代码时允许连接。有关更多信息，请参见“OCD/串行编程器 ID 设置寄存器 (OSIS)”和“ID 代码保护”以及《RA6 MCU 硬件用户手册》中的相关部分。

Renesas FSP 配置器提供了用于设置 ID 代码保护的选项。

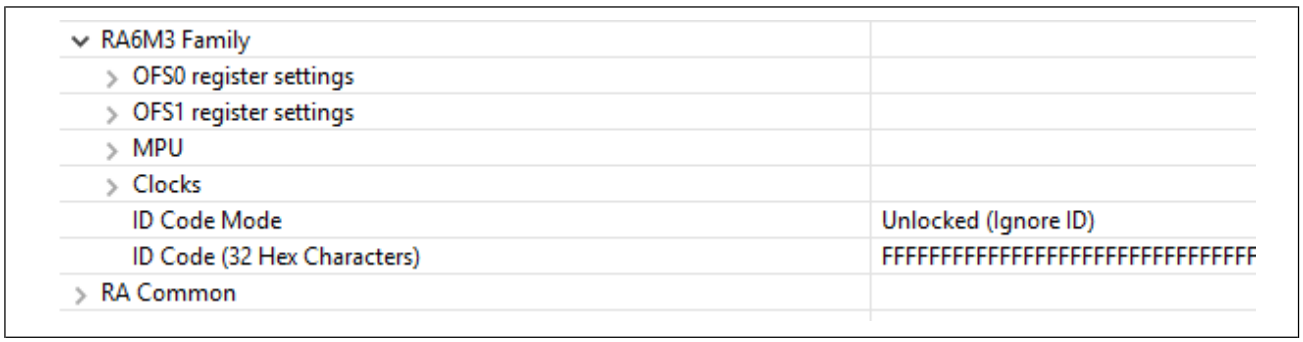

图 21. 使用 Renesas FSP 配置器设置 ID 代码

注：必须谨慎处理 ID 代码保护设置，以避免发生可能导致阻止访问 MCU 的错误。

8.4.3 器件生命周期管理

具有 Cortex-M33 内核的 RA6 MCU 配有器件生命周期管理 (DLM) 功能，能够对产品从开发伊始到生产再到使用寿命结束的整个过程进行管理。RA 系列 MCU 调试功能和串行编程功能由器件生命周期状态定义。

有关更多详细信息，请参见*器件生命周期管理密钥安装*应用笔记。

8.4.4 闪存块保护

具有 Cortex-M33 内核的 RA6 MCU 具有闪存块保护功能，可防止安全或非安全闪存区域被安全或非安全软件擦除或重新编程。值得注意的是，该保护功能同时适用于安全和非安全软件访问。

有关更多详细信息，请参见*《使用 Arm® TrustZone® 保护静态数据》*应用笔记中的“配置闪存块保护”部分。

8.4.5 内存保护单元

具有 Cortex-M4 内核的 RA6 MCU 具有内存保护单元 (MPU)。这些单元能够保护各种 MCU 区域免遭非法访问。具体选项包括允许读写、禁止写入和禁止读写。通过在特定的存储器地址上设置相应的常数值可选择这些选项之一。有关更多详细信息，请参见《MCU 硬件用户手册》中的“存储器保护单元”部分。

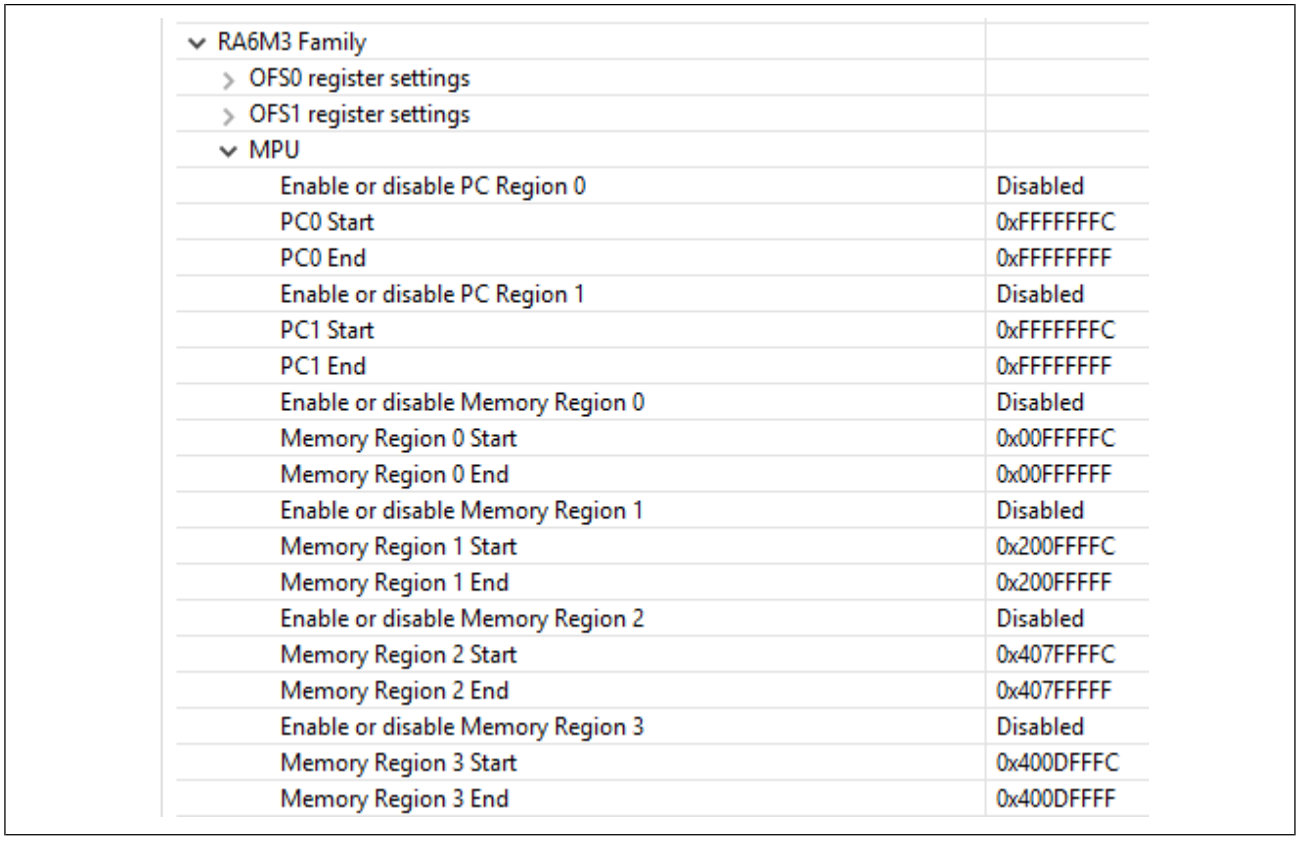

图 22. 使用 Renesas FSP 配置器设置 MPU

注：必须谨慎处理 MPU 设置，以避免发生可能导致阻止访问 MCU 区域的错误。

8.5 外部存储器

RA6 MCU 包含用于连接到外部存储器和器件的外部数据总线。某些产品还包括一个内置的 SDRAM 控制器，可通过该控制器使用最高达 128 MB 的外部 SDRAM。八个可编程片选提供了许多选项，可以在每个片选上设置这些选项，以允许连接到各种外部器件。存储器映射的外部片选区域地址从 0x60000000 开始。有关更多详细信息，请参见《硬件用户手册》。

8.5.1 使用外部 16 位存储器器件

连接具有字节选择线的外部 16 位存储器器件时，将 MCU 的 A1 连接到存储器的 A0，将 MCU 的 A0 连接到字节选择线。

8.5.2 SDRAM 初始化示例

Renesas FSP 提供了采用 CMSIS 数据结构的 C 语言头文件，此文件映射了所有外部总线控制寄存器。以下函数是在 Renesas FSP 中使用 CMSIS 寄存器结构初始化 SDRAM 存储器控制器的示例。

```
void bsp_sdram_init (void)
{
    /** Delay at least 100uS after SDCLK active */
    R_BSP_SoftwareDelay(100U, BSP_DELAY_UNITS_MICROSECONDS);

    /** Setting for SDRAM initialization sequence */
#if (BSP_PRV_SDRAM_TRP < 3)
    R_BUS->SDRAM.SDIR_b.PRC  = 3U;
#else
    R_BUS->SDRAM.SSDIR_b.PRC  = BSP_PRV_SDRAM_TRP - 3U;
#endif

    while(R_BUS->SDRAM.SDSR)
    {
        /* According to h/w maual, need to confirm that all the status bits in
SDSR are 0 before SDIR modification. */
    }

    R_BUS->SDRAM.SDIR_b.ARFC = BSP_PRV_SDRAM_SDIR_REF_TIMES;

    while(R_BUS->SDRAM.SDSR)
    {
        /* According to h/w maual, need to confirm that all the status bits in
SDSR are 0 before SDIR modification. */
    }

#if (BSP_PRV_SDRAM_TRFC < 3)
    R_BUS->SDRAM.SDIR_b.ARFI = 0U;
#else
    R_BUS->SDRAM.SDIR_b.ARFI = BSP_PRV_SDRAM_TRFC - 3U;
#endif

    while(R_BUS->SDRAM.SDSR)
    {
        /* According to h/w maual, need to confirm that all the status bits in
SDSR are 0 before SDICR modification. */
    }

    /** Start SDRAM initialization sequence.
     * Following operation is automatically done when set SDICR.INIRQ bit.
     * Perform a PRECHARGE ALL command and wait at least tRP time.
     * Issue an AUTO REFRESH command and wait at least tRFC time.
     * Issue an AUTO REFRESH command and wait at least tRFC time.
     */
    R_BUS->SDRAM.SDICR_b.INIRQ = 1U;
    while(R_BUS->SDRAM.SDSR_b.INIST)
    {
        /* Wait the end of initialization sequence. */
    }

    /** Setting for SDRAM controller */
    R_BUS->SDRAM.SDCCR_b.BSIZE  = BSP_PRV_SDRAM_BUS_WIDTH;              /* set
SDRAM bus width */
    R_BUS->SDRAM.SDAMOD_b.BE    = BSP_PRV_SDRAM_CONTINUOUS_ACCESSMODE;  /*
enable continuous access */
    R_BUS->SDRAM.SDCMOD_b.EMODE = BSP_PRV_SDRAM_ENDIAN_MODE;            /* set
endian mode for SDRAM address space */
```

```
    while(R_BUS->SDRAM.SDSR)
    {
        /* According to h/w maual, need to confirm that all the status bits in
SDSR are 0 before SDMOD modification. */
    }

    /** Using LMR command, program the mode register */
    R_BUS->SDRAM.SDMOD = ((((uint16_t)(BSP_PRV_SDRAM_MR_WB_SINGLE_LOC_ACC <<
9)
                    |(uint16_t)(BSP_PRV_SDRAM_MR_OP_MODE           << 7))
                    |(uint16_t)(BSP_PRV_SDRAM_CL                   << 4))
                    |(uint16_t)(BSP_PRV_SDRAM_MR_BT_SEQUENCTIAL    << 3))
                    |(uint16_t)(BSP_PRV_SDRAM_MR_BURST_LENGTH      << 0);

    /** wait at least tMRD time */
    while(R_BUS-
    >SDRAM.SDSR_b.MRSST)
    {
        /* Wait until Mode Register setting done. */
    }

    /** Set timing parameters for SDRAM */
    R_BUS->SDRAM.SDTR_b.RAS = BSP_PRV_SDRAM_TRAS - 1U;       /* set ACTIVE-to-
PRECHARGE command cycles*/
    R_BUS->SDRAM.SDTR_b.RCD = BSP_PRV_SDRAM_TRCD - 1U;       /* set ACTIVE to
READ/WRITE delay cycles */
    R_BUS->SDRAM.SDTR_b.RP  = BSP_PRV_SDRAM_TRP  - 1U;       /* set PRECHARGE
command period cycles */
    R_BUS->SDRAM.SDTR_b.WR  = BSP_PRV_SDRAM_TWR  - 1U;       /* set write
recovery cycles */
    R_BUS->SDRAM.SDTR_b.CL  = BSP_PRV_SDRAM_CL;             /* set SDRAM
column latency cycles */

    /** Set row address offset for target SDRAM */
    R_BUS->SDRAM.SDADR_b.MXC   = BSP_PRV_SDRAM_SDADR_ROW_ADDR_OFFSET - 8U;

    R_BUS->SDRAM.SDRFCR_b.REFW = (uint16_t)(BSP_PRV_SDRAM_TRFC - 1U);    /*
set Auto-Refresh issuing cycle */
    R_BUS->SDRAM.SDRFCR_b.RFC  = BSP_PRV_SDRAM_REF_CMD_INTERVAL - 1U;    /*
set Auto-Refresh period */

    /** Start Auto-refresh */
    R_BUS->SDRAM.SDRFEN_b.RFEN = 1U;

    /** Enable SDRAM access */
    R_BUS->SDRAM.SDCCR_b.EXENB = 1U;
}
```

8.6 数据对齐

没有对齐数据方面的限制。MCU 能够对奇数存储地址执行字节、字和长整型访问。虽然对齐数据访问仍然是最佳选择，但并不是必须的。

8.7 字节顺序限制

存储器空间必须采用小尾数法才能在 Cortex-M 内核上执行代码。

9. 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。使用保护寄存器 (PRCR) 设置要保护的寄存器。表 14 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 23. PRCR 寄存器

表 14. PRCR 保护位

PRCR 位	说明
PRC0	• 与时钟生成电路有关的寄存器： SCKDIVCR、SCKDIVCR2、SCKSCR、PLLCCR、PLLCR、BCKCR、MOSCCR、HOCOCR、MOCOCR、CKOCR、TRCKCR、OSTDCR、OSTDSR、EBCKOCR、SDCKOCR、MOCOUTCR、HOCOUTCR、MOSCWTCR、MOMCR、SOSCCR、SOMCR、LOCOCR、LOCOUTCR、HOCOWTCR、FLLCR1、FLLCR2
PRC1	• 与低功耗模式有关的寄存器： SBYCR、SNZCR、SNZEDCR、SNZREQCR、OPCCR、SOPCCR、DPSBYCR、DPSIER0-3、DPSIFR0-3、DPSIEGR0-2、SYOCDCR、STCONR • 与电池备用功能有关的寄存器： VBTBKRn（n = 0 到 511）、VBTICTLR
PRC3	• 与 LVD 有关的寄存器： LVD1CR1、LVD1SR、LVD2CR1、LVD2SR、LVCMPCR、LVDLVLR、LVD1CR0、LVD2CR0
PRKEY[7:0]	这些位控制对 PRCR 寄存器的写访问。要修改 PRCR 寄存器，将 A5h 作为高 8 位，将所需值作为低 8 位，并成一个 16 位的单元写入PRCR寄存器。

Renesas FSP 提供两个 API（R_BSP_RegisterProtectEnable 和 R_BSP_RegisterProtectDisable），分别用于使能和禁用寄存器写保护。

10. I/O 端口配置

《硬件用户手册》中的“I/O 端口”部分介绍了基于外设选择和其他寄存器设置的确切引脚配置。下面列出了一些一般信息。

务必注意的是，复位后，在应用配置之前，每个引脚都将处于其默认状态。其中某些引脚可能会短暂地处于非预期的状态。无论使用哪种配置方法，都会出现这种情况。用户应考虑这可能对各种应用产生的影响，包括可能对其他系统功能造成哪些影响。

10.1 多功能引脚选择设计策略

RA6 系列 MCU 上的大多数端口都具有多种外设功能。Renesas 提供了诸如 FSP 中的引脚配置器之类的工具，以帮助选择每个 RA6 器件的端口。当需要多个外设功能时，请使用以下设计策略来帮助选择端口功能。

- 首先，仅分配只有一个端口选项的外设功能。例如，调试功能中的每个跟踪数据信号只有一个端口选项。需要此功能时，先分配这些端口。
- 接下来，分配那些具有有限端口选项的外设功能。例如，支持 QSPI 外设的器件的每个 QSPI 信号通常只有两个选项。
- 最后，为有多个选项的外设功能分配端口。一个示例是串行通信接口 (SCI)，该接口通常具有许多可用的端口选项。
- 有些外设功能端口选项是可以互换的，而其他端口选项必须按逻辑组进行分配。例如，IIC 外设有一些端口的信号名称中带有后缀“_A”，而另一些端口的信号名称中带有后缀“_B”。为同一外设功能选择的端口应具有相同的后缀。其他外设功能不具有此类后缀，并且端口分配可以互换，例如 USB_VBUSEN 信号可用于 USBFS 外设功能。另请参见本文档的 15.3 节。

10.2 设置端口并将其用作 GPIO

有两种方法可以设置端口并将其用作 GPIO：一种是使用端口控制寄存器 (PCNTR1)，另一种是使用 PmnPFS 寄存器。

方法 1：端口控制寄存器 (PCNTR1)

- 通过向端口控制寄存器 1 (PCNTR1) 的端口方向位 (PDRn) 写入“1”，选择一个引脚作为输出。
- 端口方向位 (PDRn) 是可读写的。将该值设置为“1”将选择该引脚作为输出。I/O 端口的默认状态为“0”（输入）。可以在 RA6 MCU 上读取端口方向寄存器。
- 相应端口控制寄存器 (PCNTR1) 中的端口输出数据位 (PODRn) 是可读写的。读取 PODR 时，也会读取输出数据锁存器的状态（不是引脚电平）。
- 端口控制寄存器 2 (PCNTR2) 中的端口输入位 (PIDRn) 是只读的。读取 PCNTR2 寄存器中的 PIDRn 位以读取引脚状态。

方法 2：端口 mn 引脚功能选择 (PmnPFS) 寄存器

- 端口模式寄存器 (PMR) 是可读写的，用于指定各个引脚是用作 GPIO 还是用作外设引脚。复位后，所有 PMR 寄存器均置 0，这会将所有引脚都设置为 GPIO。如果 PMR 寄存器置 1，则该对应的引脚将用于实现外设功能。外设功能由该引脚的 MPC 设置定义。
- 将引脚设置为输出时，建议先将所需的端口输出值写入数据锁存器，然后将方向寄存器设置为输出。尽管在所有系统中此操作都不太重要，但这可以防止在设置端口时出现意外的输出毛刺。

通常，使用 PCNTR1 配置端口有助于提高访问速度，但可用的配置功能较少。使用 PmnPFS 寄存器将获得更多可用的配置功能，但访问速度较慢。

Renesas FSP 提供了引脚配置工具，以在复位后配置 GPIO 引脚，如下所示。配置 GPIO 后，可以在 FSP 中使用 HAL 层 API 对该 GPIO 进行控制。

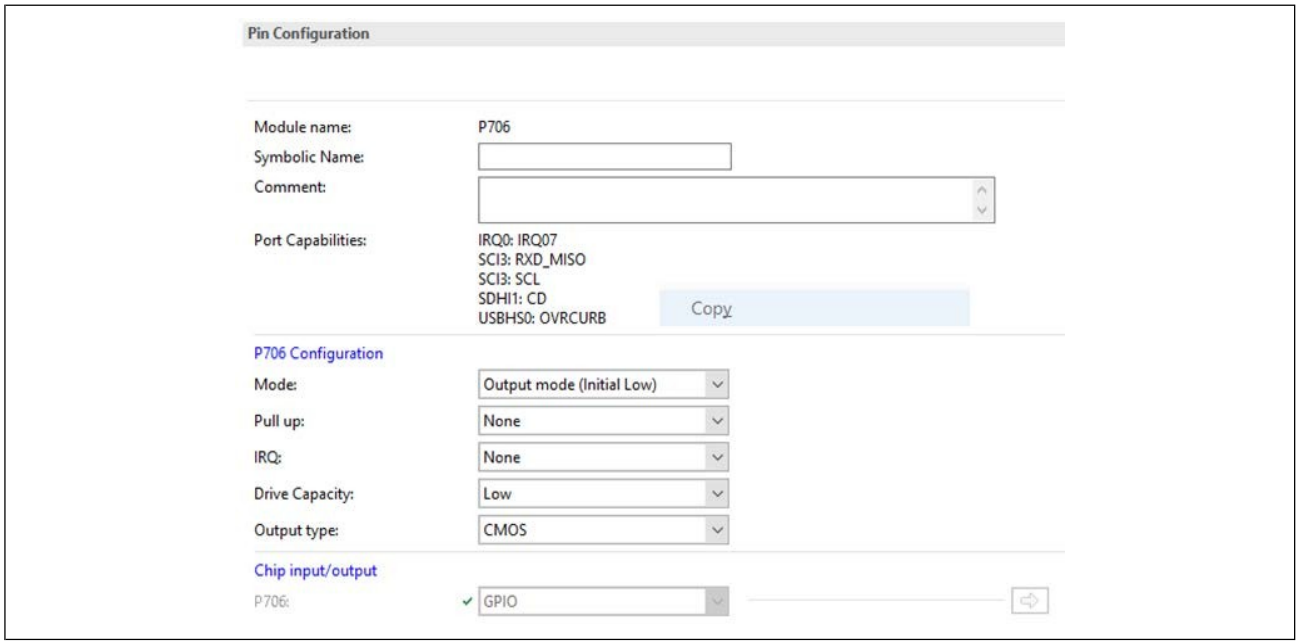

图 24. 使用 FSP 配置器将 P706 配置为输出和低电平

10.2.1 内部上拉

- 端口 0 到 9、A 和 B 上的大多数引脚都可以选择使能上拉电阻。上拉由每个端口 mn 引脚功能选择 (PmnPFS) 寄存器中的上拉 (PCR) 位控制。每个 PmnPFS 寄存器中的 PCR 位用于控制端口上的相应引脚。
- 首先，必须通过 PmnPFS 寄存器中的相关位将该引脚设置为输入。将 PCR 位置“1”以使能上拉，将其置“0”以禁用上拉。
- 复位后，会将所有 PCR 寄存器清零，因此将禁用所有上拉电阻。
- 每当将某个引脚指定为外部总线引脚、GPIO 输出或外设功能输出引脚时，上拉就会自动关闭。

10.2.2 漏极开路输出

- 配置为输出的引脚通常用作 CMOS 输出。
- 端口 0 到 9、A 和 B 上的大多数引脚都可以选择配置为 NMOS 漏极开路输出。
- 每个端口 mn 引脚功能选择 (PmnPFS) 寄存器中的 N 沟道漏极开路控制 (NCODR) 位控制哪些引脚以漏极开路模式工作。将每个寄存器中的适用位置“1”会使输出形式变为漏极开路。将每个寄存器中的适用位置“0”会将端口设置为 CMOS 输出。

10.2.3 驱动能力

- 端口 1 到 9、A 和 B（P200 和 P201 除外）上的每个引脚都可以选择使能低、中或高驱动能力输出。驱动能力的切换由每个端口 mn 引脚功能选择 (PmnPFS) 寄存器中的驱动能力控制寄存器 (DSCR) 位控制。
- 复位后，会将所有 DSCR 寄存器清零，因此会将所有引脚设置为低驱动能力输出。设置“00”以外的值会更改所选引脚的输出的驱动能力。
- 所有引脚的最大总输出为 80 mA。
- 驱动能力的差异如下所示：

典型输出引脚	DSCR[1:0]	驱动能力	平均电流 (mA)	最大电流 (mA)
每个引脚允许的输出电流	0 0	低驱动能力	2.0	4.0
每个引脚允许的输出电流	0 1	中驱动能力	4.0	8.0
无效设置；不要用	1 0	-		
每个引脚允许的输出电流	1 1	高驱动能力	16	32

高驱动能力输出引脚	DSCR[1:0]	驱动能力	平均电流 (mA)	最大电流 (mA)
每个引脚允许的输出电流	0 0	低驱动能力	2.0	4.0
每个引脚允许的输出电流	0 1	中驱动能力	4.0	8.0
无效设置；不要用	1 0	-		
每个引脚允许的输出电流	1 1	高驱动能力	20	40

输出驱动能力可能会对电路板设计的整体性能造成重大影响。为每个输出选择驱动能力时，应考虑以下几点。

- 建议首先将所有引脚设置为低驱动能力（默认）并评估性能。
- 根据电路板布局的不同，驱动能力设置为中或高的引脚可能会产生较高的 EMI 辐射。
- 较长的走线可能需要更高的驱动能力，才能使信号正确传播到接收器。

10.3 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 位用于选择为每个端口选择的外设功能。

- 由于大多数引脚具有多种功能，因此 RA6 MCU 提供了引脚功能控制寄存器 (PmnPFS)，可用于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 寄存器都允许将引脚用于外设功能（PSEL 位）、用作 IRQ 输入引脚（ISEL 位）或用作模拟输入引脚（ASEL 位）。如果 ASEL 位置“1”（将引脚用作模拟输入引脚），则应将该引脚的 PMR 位置 1 以用于 GPIO，并将该引脚的 PDR 位置 1 以用于输入。
- 请参见《硬件用户手册》中“I/O 端口”一章的“每种产品的外设选择设置”部分。
- 为了确保外设引脚上没有非预期的边沿输入或输出，确保在修改引脚的 PmnPFS 寄存器之前将目标引脚的端口模式控制 (PMR) 位清零。
- 复位后，所有 PmnPFS 寄存器均受到写保护。为了对这些寄存器执行写入操作，必须首先使用写保护寄存器 (PWPR) 来使能写入。
- 设置 PmnPFS 寄存器时应格外小心，切勿将一个功能分配给多个引脚。用户不应这样做，但 MCU 本身不禁止这样的设定。如果发生这种情况，则引脚上的功能将处于未定义状态。
- 如果打算使用外部总线、以太网控制器或 USB，则在使用这些外设之前，必须先配置 MPC 中的其他寄存器。
- 下图显示了使用 FSP 引脚配置器使能 QSPI 引脚的示例。

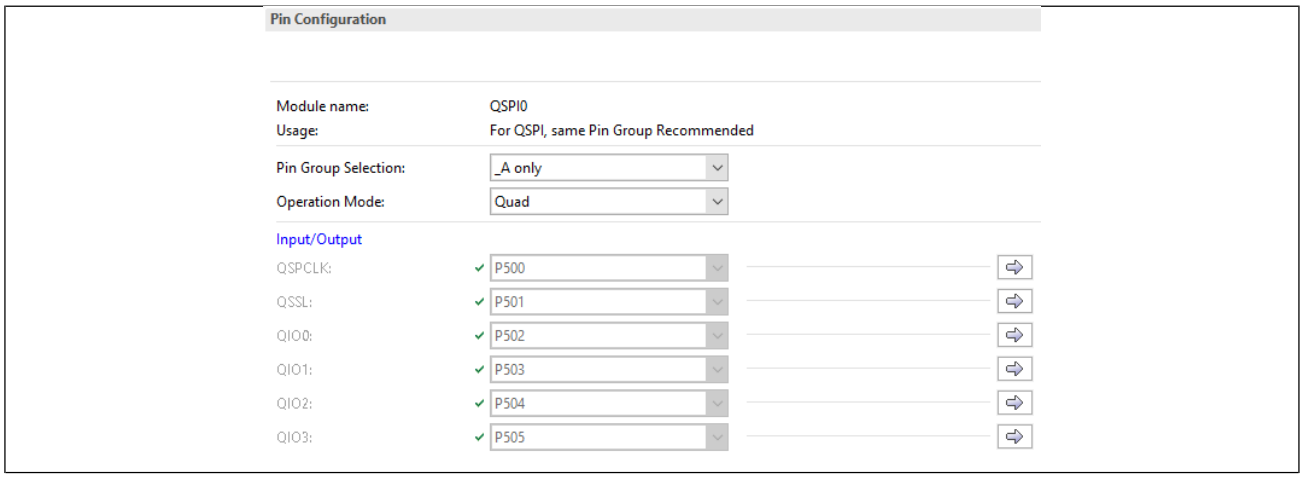

图 25. 使用 Renesas FSP 中的引脚配置器使能 QSPI 引脚

10.4 设置和使用 IRQ 引脚

- 某些端口引脚可以用作硬件中断输入 (IRQ)。有关哪些引脚可用于您的 MCU 的信息，请参见《硬件用户手册》中“I/O 端口”一章的“每种产品的外设选择设置”部分。
- 某些 IRQ 引脚带有“-DS”后缀（例如 IRQ1-DS）。“-DS”表示该引脚可用于将 MCU 从深度软件待机模式中唤醒。
- 注：不能同时使用 IRQn 和 IRQn-DS。带 -DS 后缀和不带 -DS 后缀的相同编号的中断在内部连接到同一中断，即使它们使用不同的外部引脚连接。
- 要将端口引脚设置为 IRQ 引脚，必须将其 PFS 寄存器中的中断输入功能选择位 (ISEL) 置“1”。
- 引脚可同时用于 IRQ 和外设功能。要使能此功能，用户应将引脚的 PFS 寄存器中的 ISEL 和 PSEL 位置 1。
- 具有相同编号的 IRQ 功能必须只在一个引脚上使能。
- IRQ 引脚可以在检测到以下情况时触发中断：
  — 低电平
  — 下降沿
  — 上升沿
  — 上升沿和下降沿
  使用 IRQ 控制寄存器 (IRQCRi) 选择要使用的触发形式。

- 数字滤波可用于 IRQ 引脚。滤波器基于以四个可选时钟速率（PCLK、PCLK/8、PCLK/32、PCLK/64）之一进行的重复信号采样。它们可以滤除短脉冲（在特定滤波速率下，任何少于 3 个样本的高脉冲或低脉冲）。滤波器可用于滤除这些线路中的振铃和噪声，但是由于速率过快而无法滤除诸如机械开关弹跳之类的长时间事件。使能滤波会给硬件 IRQ 输入增加一小段响应延迟时间（滤波时间）。
- 可以针对每个 IRQ 引脚单独使能数字滤波。此过程通过将每个 IRQ 的 IRQCRi 寄存器中的 IRQ 引脚数字滤波器使能 (FLTEN) 位置 1 来实现。
- 可为每个 IRQ 引脚单独配置数字滤波的时钟速率。此过程通过将每个 IRQ 的 IRQCRi 寄存器中的 IRQ 引脚数字滤波器设置 (FCLKSEL[1:0]) 位置 1 来实现。
- 图 26 和图 27 给出了使用 Renesas FSP 使能和配置 IRQ 引脚的示例。

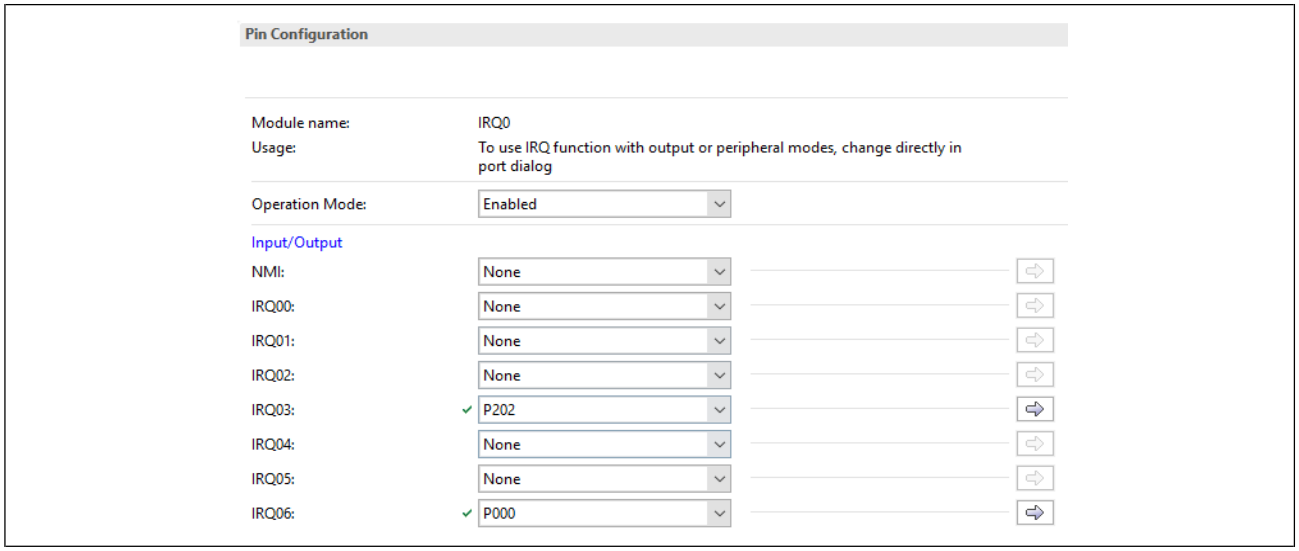

图 26. 使用 Renesas FSP 中的引脚配置器分别使能 P202、P000 即 IRQ03、IRQ06 输入

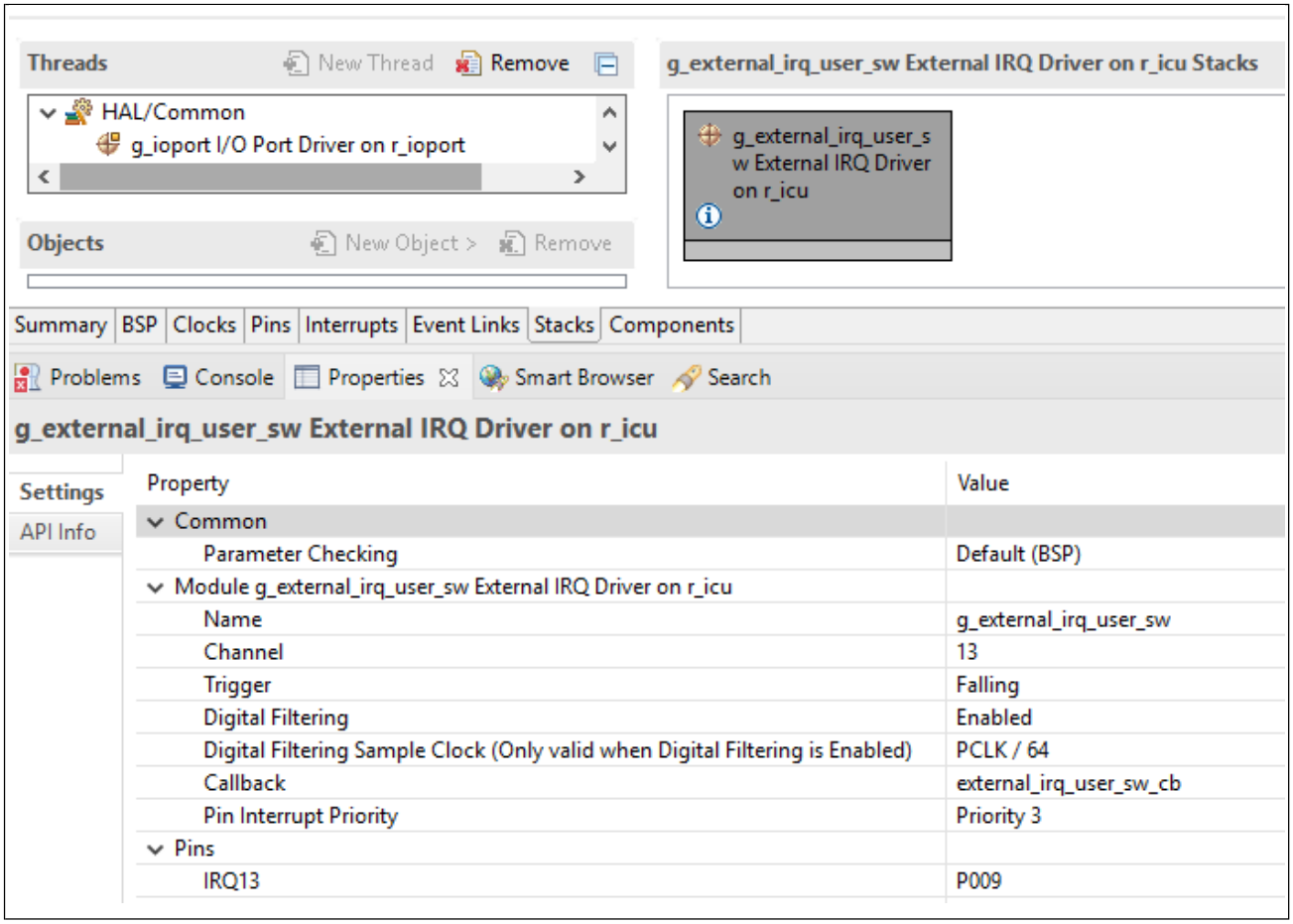

图 27. 使用 Renesas FSP 配置器配置 IRQ13

10.5 未使用的引脚

注：某些引脚需要特定的端接：有关具体建议，请参见《硬件用户手册》中的“未使用引脚的处理”部分。

悬空的未使用引脚会产生额外的功耗，并使系统更容易受到噪声问题的影响。使用下面详述的方法之一对未使用的引脚进行端接：

1. 第一种选择是将引脚设置为输入（复位后的默认状态），然后使用电阻将其连接至 Vcc 或 Vss。不同的连接方法对 MCU 来说没有区别；但是，从系统噪声的角度来看，有的方法可能更具优势。Vss 可能是最典型的选择。避免将引脚直接连接到 Vcc 或 Vss，因为对端口的方向寄存器（用于将端口设置为输出）执行意外的写入操作可能会导致输出短路。
2. 第二种方法是将引脚设置为输出。无论是将引脚电平设置为高还是低都无关紧要；但是，将引脚设置为输出并将输出设置为低电平会在内部将引脚连接至接地层。这可能有助于解决整个系统的噪声问题。将未使用的引脚设置为输出有一个缺点，即必须通过软件控制来完成端口配置。在将方向寄存器设置为输出前 MCU 一直保持在复位状态时，该引脚将为悬空输入，可能会消耗额外的电流。如果可以接受在此期间消耗额外的电流，则该方法可消除第一种方法所需的外部电阻。
3. 将引脚保留为输入并使用外部电阻将它们端接的一种变化方式是使用 MCU 多个端口上的内部上拉电阻。这与将引脚设置为输出有相同的限制（需要程序来设置端口），但是由于器件不会驱动引脚，因此确实可以减少因引脚意外接地、短接到相邻引脚或 Vcc 而产生的影响。

10.6 不存在的引脚

每个 RA6 MCU 系列都有多种封装尺寸，总引脚数也各有不同。对于小于该 MCU 系列最大封装尺寸（通常为 176 引脚）的任何封装，需将 PDR 寄存器中不存在的端口的对应位置“1”（输出），并将 PODR 寄存器中不存在的端口的对应位置“0”。通过查看《硬件用户手册》中“I/O 端口”部分的“I/O 端口规范”表，用户可以查看每个 MCU 封装上可用的端口。例如，端口 1 上的引脚 0 和 1 仅在 176 引脚封装中可用。请注意，不需要对不存在的引脚进行额外处理。

10.7 电气特性

常规 GPIO 端口通常需要 CMOS 电平输入（高电平 ≥ 0.8 * Vcc，低电平 ≤ 0.2 * Vcc）。某些 GPIO 端口具有施密特触发输入，在输入要求方面略有不同。有关更多信息，请参见《硬件用户手册》中的“电气特性”部分。

11. 模块停止功能

为了尽可能提高电源效率，RA6 系列 MCU 允许通过对模块停止控制寄存器（MSTPCRi，i = A、B、C、D、E）执行写入操作来分别停止片上外设。模块停止后，将无法访问模块寄存器。

复位后，除 DMAC、DTC 和 SRAM 外，大多数模块都处于模块停止状态。有关详细信息，请参见《硬件用户手册》。

在访问外设的任何寄存器之前，必须通过向 MSTPCRi 寄存器中的相应位写入“0”以使其退出停止模式来使能寄存器。

可以通过向 MSTPCRi 寄存器中的相应位写入“1”来停止外设。

Renesas FSP 中的 HAL 驱动程序会自动处理模块的启动/停止功能。

12. 中断控制单元

中断控制器单元 (ICU) 控制将哪些事件信号链接到 NVIC、DTC 和 DMAC 模块。此外，ICU 还控制不可屏蔽中断。图 28 给出了 ICU 规范的示例，图 29 给出了从 I/O 引脚引发 IRQi 事件的功能示例。有关每个 RA6 MCU 系列的详细信息，请参见《硬件用户手册》。

Parameter		Specifications
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 315 (select factors within event list numbers 64 to 511)
	External pin interrupts	• Interrupt detection on low level, falling edge, rising edge, rising and falling edges One of these detection methods can be set for each source. • Digital filter function supported • 16 sources, with interrupts from IRQ0 to IRQ15 pins.
	DTC and DMAC control	The DTC and DMAC can be activated using interrupt sources*1
	Interrupt sources for NVIC	96 sources
Non-maskable interrupts*2	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported.
	Oscillation stop detection interrupt*3	Interrupt on detecting that the main oscillator has stopped
	WDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Voltage monitor 1 interrupt*3	Voltage monitor interrupt of Low Voltage Detection detector 1 (LVD1)
	Voltage monitor 2 interrupt*3	Voltage monitor interrupt of Low Voltage Detection detector 2 (LVD2)
	RPEST	Interrupt on SRAM parity error
	RECCST	Interrupt on SRAM ECC error
	BUSSST	Interrupt on MPU bus slave error
	BUSMST	Interrupt on MPU bus master error
	SPEST	Interrupt on CPU stack pointer monitor
Return from low power mode*4		• Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source • Software Standby mode: Return is initiated by non-maskable interrupts Interrupts can be selected in the WUPEN register. • Snooze mode: Return is initiated by non-maskable interrupts Interrupts can be selected in the SELSR0 and WUPEN registers.

图 28. RA6M3 ICU 规范示例

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to 15)	Input	External interrupt request pins

图 29. RA6M3 ICU I/O 引脚示例

下图是使用 Renesas FSP 配置器使能和配置 Renesas FSP 中断的示例。通过 FSP 将 ICU 和中断配置为 HAL 驱动程序配置的一部分。

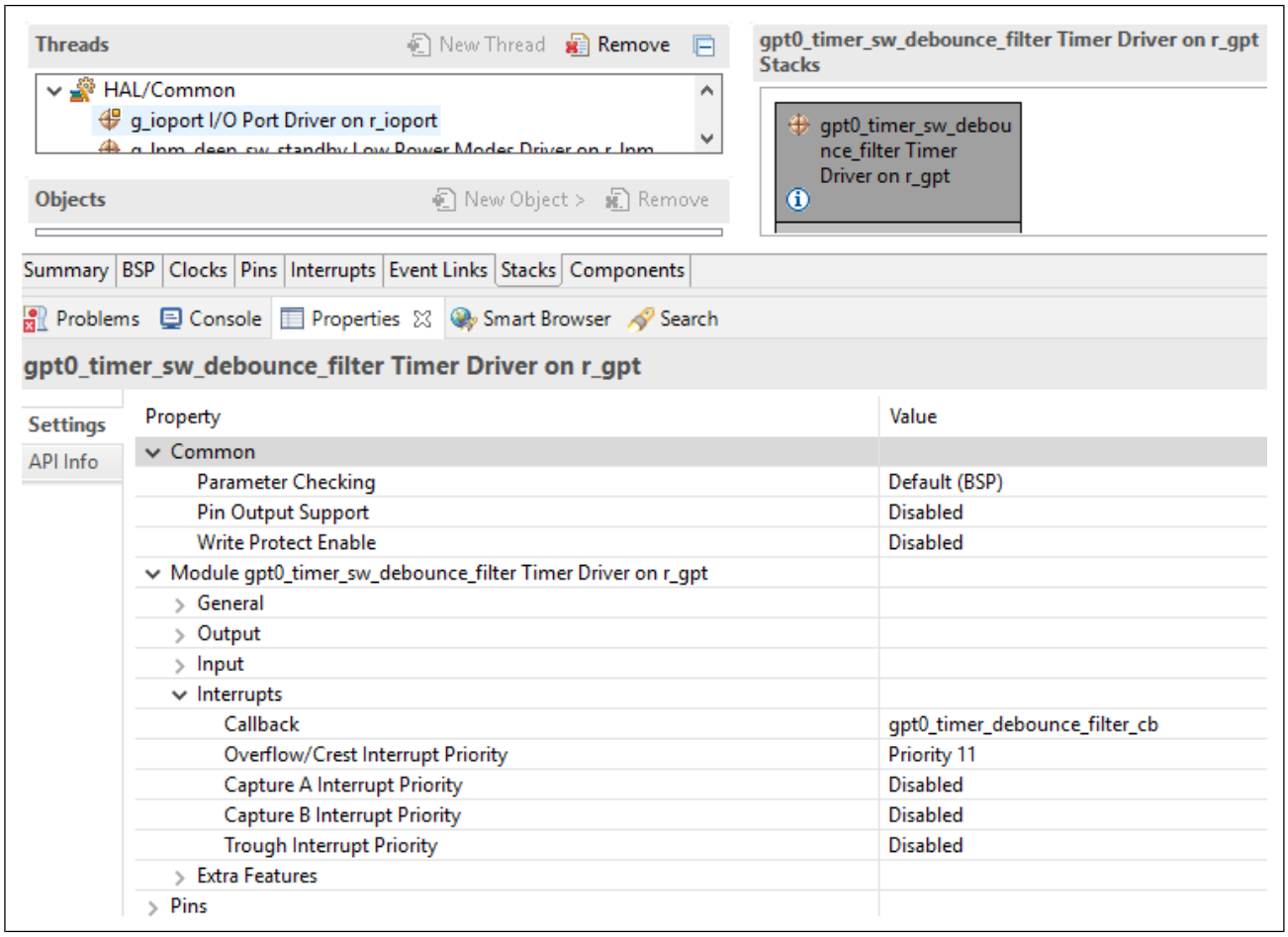

图 30. 使能 GTP0 溢出中断并设置将由中断服务程序调用的用户回调函数

13. 低功耗

RA6 器件具有多种用于降低功耗的功能。这包括设置时钟分频器、EBCLK 输出控制、停止模块、在正常模式下选择电源控制模式以及转换为低功耗模式。有关更多详细信息，请参见《硬件用户手册》中的“低功耗模式”一章。

RA6 MCU 支持四种不同类型的 LPM，具体取决于 MCU 系列。这些类型包括：

- 休眠模式
- 软件待机模式
- 贪睡模式
- 深度软件待机模式。

下表概述了可用于降低功耗的功能。

表 15. 低功耗模式功能规范

项目	规范
通过修改时钟信号降低功耗	可以为系统时钟 (ICLK)、外设模块时钟（PCLKH、PCLKL、PCLKADC、PCLKGPT）、外部总线时钟 (BCLK) 和闪存接口时钟 (FCLK) 单独选择分频比*1
EBCLK 输出控制	可以选择 BCLK 输出或高电平输出。
模块停止	可以单独停止每个外设模块的功能。
低功耗模式	• 休眠模式 • 软件待机模式 • 贪睡模式 • 深度软件待机模式
电源控制模式	三种工作电源控制模式： • 高速模式 • 低速模式 • 子时钟振荡器速度模式
TrustZone 过滤器*2	可以为每个寄存器设置安全属性

注：1. 有关详细信息，请参见《硬件用户手册》中的“时钟生成电路”一章。
2. 适用于支持 TrustZone 安全功能的器件。

下表列出了转换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 16. 低功耗模式

运行状态*1	休眠模式	全模块时钟停止模式	软件待机模式	深度软件待机模式
转换条件	SBYCR.SSBY=0 时的 WFI 指令	SBYCR.SSBY=1 和 DPSBYCR.DPSBY=0 时的 WFI 指令	在软件待机模式下，贪睡请求触发。SNZCR.SNZE=1	SBYCR.SSBY=1 和 DPSBYCR.DPSBY=1 时的 WFI 指令
退出方法	所有中断。该模式下可用的任何复位。	为此模式定义的中断。该模式下可用的任何复位。	为此模式定义的中断。该模式下可用的任何复位。	为此模式定义的中断。该模式下可用的任何复位。
通过中断退出后的状态	程序执行状态（中断处理中）	程序执行状态（中断处理中）	程序执行状态（中断处理中）	复位状态
通过复位退出后的状态	复位状态	复位状态	复位状态	复位状态

注：1. 有关更多详细信息，请参见《硬件用户手册》中的“每种低功耗模式的工作条件”表。

RA6 器件包括一些寄存器设置，允许 MCU 在正常模式和休眠模式下以较低功耗工作。这些模式称为工作电源控制模式，由 OPCCR 寄存器控制。

下表总结了各种工作功耗控制模式，以及每种模式下允许使用的最大时钟和电压值。

表 17. 每种工作功耗控制模式下可用的振荡器

模式	振荡器						
	PLL 和 PLL2	**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**	**子时钟振荡器**	**IWDT 专用片上振荡器**
高速	可用	可用	可用	可用	可用	可用	可用
低速	不适用	可用	可用	可用	可用	可用	可用
子时钟速度	不适用	不适用	不适用	可用	不适用	可用	可用

注：虽然可以将 OPCCR 寄存器中的值设置为任何低功耗工作模式，但也必须设置相应的时钟和电压值来满足所需模式的要求。否则，OPCCR 寄存器中的设置不会对降低功耗产生任何影响。

为了获得最低的功耗值，应在时钟生成电路中使用可能的最大分频器。

可通过各种中断源退出低功耗模式，例如 RES 引脚复位、上电复位、电压监视器复位和外设中断。有关不同 LPM 的中断源列表，请参见《硬件用户手册》中的“低功耗模式”部分。

从软件待机模式进入贪睡模式的贪睡请求仅触发贪睡模式。通过在待机控制寄存器 (SBYCR) 中执行具有适当设置的 WFI 指令，可以完成向其他 LPM 的转换。

Renesas FSP 提供了低功耗模式 (LPM) 驱动程序和驱动程序配置器，可用于设置低功耗模式、唤醒源/退出源等。

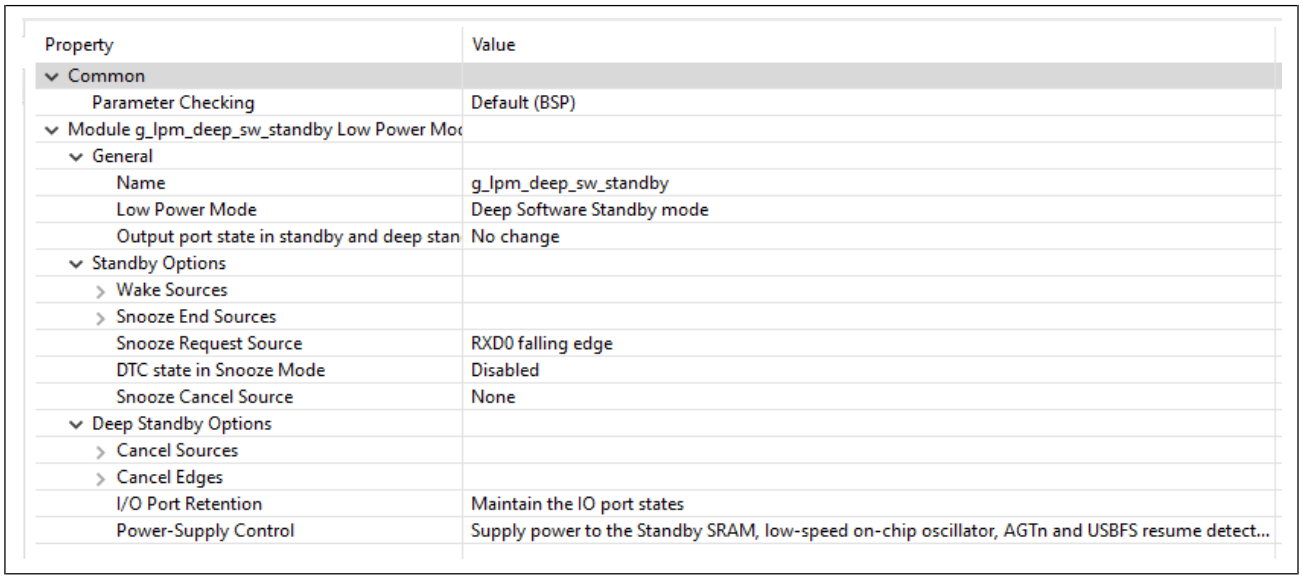

图 31. 使用 Renesas FSP 配置器设置低功耗模式

在通过 FSP 配置器设置了特定的 LPM 模式后，可以使用 LPM 驱动程序的 API 初始化 LPM 驱动程序并使 MCU 进入已配置的 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);
  }
```

14. 外部总线

RA6 器件包括一个外部总线控制器。某些 RA6 器件具有内置的 SDRAM 控制器。

14.1 总线宽度和多路复用

外部存储区的访问宽度可以设置为 8 位或 16 位。通过将 CSnCR 寄存器或 SDC 控制寄存器 (SDCCR) 中的 BSIZE 位置 1，可以选择按每个片选进行宽度设置。可以通过将 CSnCR 寄存器中的 MPXEN 位置 1 来复用片选区域的地址和数据线。

14.2 总线信号的驱动能力

使用外部存储区时，应在高速设置中将用于控制总线信号的引脚设置为高驱动能力输出。有关设置引脚驱动能力的更多信息，请参见“I/O 端口”一章中的“端口 mn 引脚功能选择寄存器”一节和《MCU 硬件用户手册》中的“电气特性”一章。

14.3 总线错误

每条总线上都可能发生以下类型的错误：

- 非法地址访问
- 总线主 MPU 错误
- TrustZone 过滤器错误（仅限支持 Trustzone 的 MCU）。
- 每个从 IP 传来的总线错误

发生总线错误时，无法保证正常运行，并且会将错误返回到发出请求的主 IP。每个主器件发生的总线错误均存储在 BUSnERRADD 和 BUSnERRSTAT 寄存器中。只能通过复位来清空这些寄存器。有关更多信息，请参见《硬件用户手册》的“总线错误地址寄存器 (BUSnERRADD)”和“总线错误状态寄存器 (BUSnERRSTAT)”部分。

注：DMAC 和 DTC 不会收到总线错误，因此它们的运行不受总线错误的影响。

15. 一般布线实践

15.1 数字域与模拟域

Renesas RA6 微控制器器件主要有三种类型的引脚功能：电源、数字和模拟。

通常，电源引脚专门用于电压和参考输入，没有多种功能。电源引脚通常专门用于 MCU 内的特定部分或域。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。可以将数字域定义为数字电路、数字 I/O 引脚以及相关的电源引脚。指定用于模拟功能的电源引脚（例如 AVCC0 和相关的 AVSS0）在 MCU 内部提供特定的模拟电路，这类模拟电路与数字域电路分开。可以将模拟域定义为模拟电路、模拟 I/O 引脚以及相关的电源引脚。

数字信号通常是与周期性时钟相关联的重复切换模式。数字信号上的跳变往往是相对尖锐的边沿，同时跳变之间保持稳定的高电平或低电平。在指定的时间范围内，每个信号必须在可接受的电压大小下处于稳定状态，称为逻辑状态。通常使用时钟的边沿跳变以预定的时钟间隔对信号状态进行采样，以评估相关的数据信号。只要电平保持在指定范围内，数字信号电压值的小幅度变化通常是可以接受的。但是，如果数字信号受到可能会对其造成严重影响的较大外部作用，则可能会在对数据进行采样时引起错误的逻辑状态。

模拟信号通常截然不同。模拟信号可能是周期性的，但模拟信号的评估通常是在一定范围内而不是逻辑状态下测量电压。根据特定的触发事件对模拟信号的电压大小进行采样，然后使用 MCU 中的模拟电路处理得到的测量结果。模拟测量的精度与采样电压值的精度直接相关。任何可能会略微改变模拟输入信号电压值的非预期外部作用，都可能影响测量的准确性。

由于 Renesas RA6 MCU 器件的 I/O 引脚的高度复用特性，许多 I/O 引脚都可用于实现模拟或数字功能。这可能会导致数字和模拟功能发生重叠，并造成数据错误。

为了尽可能地减少数字信号域和模拟信号域之间的潜在问题，请考虑以下准则。

- 在分配 I/O 引脚功能时，选择的引脚功能应尽量使模拟引脚和数字引脚在物理上分开。
- 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。避免在同一区域内连接其他任何信号走线，无论是模拟信号还是数字信号。
- 确保模拟电源电压和模拟参考电压包含适当的交流滤波器。可以采用如下形式：在 MCU 电压引脚附近放置建议使用的电容，或使用适当的感应滤波器。此举旨在提供很少甚至没有电压纹波的电源电压和参考电压。
- 在 PCB 设计中使用专用电源层时，避免在模拟电压区域内连接数字信号走线，并避免在数字电压区域内连接模拟信号走线。

对于灵敏度较高的应用，强烈建议使用仿真工具评估特定的设计，以了解电路设计对性能的影响。例如，这可能包括诸如精密传感器设计或超高速数字总线接口之类的应用。有关每种外设功能的特定要求，请参见《硬件用户手册》中的“电气特性”一章。

15.2 高速信号设计注意事项

随着数字信号时钟速度的增加，外部刺激对这些信号的影响会变得更加明显。某些外设功能可以归类为“高速”数字信号。对于高速数字信号，还应考虑其他设计注意事项。

在发生串扰时，一个信号上的跳变会对附近的另一个信号产生感应影响。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上发生错误。为了减少串扰的影响，请使用以下一般 PCB 布线准则。

- 为同一布线层上的已连线信号之间提供足够的空间。通常，在同一数字组的信号之间至少保留一倍走线宽度的空间，而在不同数字组的信号之间至少保留 3-5 倍走线宽度的空间。
- 为同一布线层上的时钟信号和数据信号之间提供额外的空间。通常，在时钟和任何其他数字信号之间至少要保留 3-5 倍走线宽度的空间。
- 避免在任何相邻的布线层上平行连接数字信号走线。如果必须在相邻的信号层上走线，请尽可能尝试仅使用正交交叉走线。

如有可能，请在信号层之间使用电源层或接地层来分隔 PCB 信号层。电源层或接地层的铜皮可以用作数字信号的“屏蔽层”。

每个标准化接口都有特定的要求。为确保 PCB 设计不会出现信号串扰问题，强烈建议设计每个接口时都参考相关标准。

15.3 信号组选择

某些引脚名称带有附加的 _A、_B 或 _C 后缀来表示信号组。当分配某些外设功能（例如 IIC、SPI、SSIE、ETHERC 和 SDHI）时，请选择具有相同后缀的功能引脚。在某些情况下，将针对每个信号组测量《硬件用户手册》的“电气特性”一章中显示的交流时序特性。如果混用信号组，则无法保证外设正常工作，并且所述的交流时序特性可能不适用。

如果外设功能的引脚名称不带信号组后缀，则可以为每个功能信号选择最方便的引脚分配。

请参见《硬件用户手册》中“I/O 端口”一章的“每种产品的外设选择设置”和“PmnPFS 寄存器设置的注意事项”部分。

16. 参考资料

在编写本《快速设计指南》时，使用了以下文档：

参考资料	文档编号	说明
1	R01UH0884EJ0100	Renesas RA6M1 系列用户手册：硬件
2	R01UH0885EJ0100	Renesas RA6M2 系列用户手册：硬件
3	R01UH0886EJ0100	Renesas RA6M3 系列用户手册：硬件
4	R01UH0890EJ0110	Renesas RA6M4 系列用户手册：硬件
5	R01UH0891EJ0110	Renesas RA6M5 系列用户手册：硬件
6	R20AN0577EG0101	RA Arm® TrustZone® 工具入门
7	R11AN0467EU0100	Arm® TrustZone® 的安全设计 - IP 保护
8	R11AN0468EU0100	使用 Arm® TrustZone® 保护静态数据
9	R11AN0475EU0100	使用 SCE9 和 Arm® TrustZone® 建立和保护器件标识
10	R11AN0469EU0110	器件生命周期管理密钥安装

网站和支持

如需了解 RA 系列的关键信息、下载软件和相关文档以及获得支持，请访问以下短 URL。

RA 产品信息	www.renesas.com/ra
RA 产品支持论坛	www.renesas.com/ra/forum
RA 灵活配置软件包	www.renesas.com/FSP
Renesas 支持	www.renesas.com/support

版本历史记录

版本	日期	说明	
		页码	摘要
1.00	2021 年 2 月 18 日	—	初始版本
1.01	2021 年 4 月 6 日	—	添加了用于区分Cortex-M33 MCU的说明 修正了5.1，RA6 MCU以MOCO启动

注意

1. 本文档中电路、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、软件和信息的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，对于因使用本文档中所述的 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. 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。
5. 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。
6. 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、工作电源电压范围、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条件下使用 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。
7. 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任何其他适当的措施。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。
8. 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），并依照所有适用的法律和法规使用 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。
9. 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。
10. 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，负责将本文档中阐明的内容和条件提前通知前述第三方。
11. 未经 Renesas Electronics 事先书面同意，不得以任何形式全部或部分重印、再现或复制本文档。
12. 如果对本文档中包含的信息或 Renesas Electronics 产品有任何疑问，请联系 Renesas Electronics 销售办事处。

（注 1）本文档中的“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 4.0-1，2017 年 11 月）

公司总部

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

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/

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。

© 2021 Renesas Electronics Corporation。保留所有权利。