Renesas RA Family

RA6 Quick Design Guide

Introduction

This document answers common questions and points out subtleties of the RA6 MCU that might be missed unless the hardware manual was extensively reviewed. The document is not intended to be a replacement for the hardware manual; it is intended to supplement the manual by highlighting some key items most engineers will need to start their own design. It also discusses some design decisions from an application point of view.

Target Device

RA6 MCU Series

Contents

1. Power Supplies 3
1.1 References 4
2. Emulator Support 5
2.1 SWD Interface 6
2.2 JTAG Interface 7
2.3 Serial Programming Interface using SCI 8
2.4 Serial Programming Interface using SCI: Devices with TrustZone® Support 9
2.5 Multiple Emulator Interface 10
2.6 Software Setups for Emulator Connections 11
2.6.1 SWD and JTAG Interfaces 11
2.6.2 Trace Port 11
3. MCU Operating Modes 12
4. Option Setting Memory 13
4.1 Option Setting Memory Registers 13
5. Clock Circuits 14
5.1 Reset Conditions 15
5.2 Clock Frequency Requirements 15
5.2.1 Requirements for USB Communications 16
5.2.2 Requirements for Ethernet Controller 16
5.2.3 Requirements for Programming and Erasing ROM or Data Flash 16
5.2.4 Requirements for SDRAM Controller 16
5.3 Lowering Clock Generation Circuit (CGC) Power Consumption 16
5.4 Writing the System Clock Control Registers 16
5.5 Clock Setup Example 17
5.6 HOCO Accuracy 17
5.7 Flash Interface Clock 17

5.8 Board Design 17
5.9 External Crystal Resonator selection 17

6. Reset Requirements and the Reset Circuit 18
6.1 Pin Reset 19
6.2 Power-On Reset 19
6.3 Independent Watchdog Timer Reset 19
6.4 Watchdog Timer Reset 20
6.5 Voltage-Monitoring Resets 20
6.6 Deep Software Standby Reset 20
6.7 Software Reset 20
6.8 Other Resets 20
6.9 Determination of Cold/Warm Start 20
6.10 Determining the Reset Source 20

7. TrustZone® Enable 21
7.1 Implementation of Arm® TrustZone® Technology 21
7.2 Emulator Support for TrustZone 22

8. Memory 24
8.1 SRAM 25
8.2 Standby SRAM 25
8.3 Peripheral I/O Registers 26
8.4 On-Chip Flash Memory 26
8.4.1 Background Operation 27
8.4.2 ID Code Protection 28
8.4.3 Device Lifecycle Management 28
8.4.4 Flash Block Protection 28
8.4.5 Memory Protection Unit 28
8.5 External Memory 29
8.5.1 Using External 16-bit Memory Devices 29
8.5.2 Example of SDRAM Initialization 29
8.6 Data Alignment 31
8.7 Restriction on Endian 31

9. Register Write Protection 31

10. I/O Port Configuration 32
10.1 Multifunction Pin Selection Design Strategies 32
10.2 Setting Up and Using a Port as GPIO 33
10.2.1 Internal Pull-Ups 34
10.2.2 Open-Drain Output 34
10.2.3 Drive Capacity 34

10.3 Setting Up and Using Port Peripheral Functions 35
10.4 Setting Up and Using IRQ Pins 36
10.5 Unused Pins 37
10.6 Nonexistent Pins 38
10.7 Electrical Characteristics 38
11. Module Stop Function 38
12. Interrupt Control Unit 38
13. Low Power Consumption 39
14. External Buses 42
14.1 Bus Width and Multiplexing 42
14.2 Drive Strength for Bus Signals 42
14.3 Bus Errors 42
15. General Layout Practices 43
15.1 Digital Domain vs. Analog Domain 43
15.2 High Speed Signal Design Considerations 44
15.3 Signal Group Selections 44
16. References 44
Revision History 46

1. Power Supplies

The RA family has digital power supplies and analog power supplies. The power supplies use the following pins.

Table 1. Digital Power Supplies

Symbol	Name	Description
VCC	Power supply	Power supply pin. Connect to the system power supply. Connect this pin to VSS via a 0.1 µF capacitor placed close to the VCC pin.
VSS	Ground	Ground
VCL	Power supply	Connect this pin to VSS via a 0.1 µF capacitor close to the VCL pin.
VCL0	Power supply	Connect this pin to VSS via a 0.1 µF capacitor close to the VCL0 pin.
VBATT	Backup power	Backup power pin. Supplies power to RTC and sub-clock oscillator in the absence of VCC. When VBATT pin is not used, connect to VCC or VSS.
VCC_USB	USB FS power supply	USB Full-speed power supply pin. Connect this pin to VCC. Connect this pin to VSS_USB via a 0.1 µF capacitor placed close to the VCC_USB pin.
VSS_USB	USB FS ground	USB Full-speed ground pin. Connect this pin to VSS.
VCC_USBHS[1]	USB HS power supply	USB High-speed power supply pin. Connect this pin to VCC. Connect this pin to VSS1_USBHS or VSS2_USBHS via a 0.1 µF capacitor placed close to the VCC_USBHS pin.

Symbol	Name	Description
VSS1_USBHS[1]	USB HS ground	USB High-speed ground pin. Connect this pin to VSS.
VSS2_USBHS[1]	USB HS ground	USB High-speed ground pin. Connect this pin to VSS.

Note: 1. Only for devices with USB High Speed peripheral.

Table 2. Analog Power Supplies

Symbol	Name	Description
AVCC0	Analog power supply	Analog voltage supply pin for the respective modules. Connect this pin to the same voltage as the VCC pin.
AVSS0	Analog ground	Analog ground for the respective modules. Connect this pin to the same voltage as the VSS pin.
VREFH0	12-bit ADC high reference voltage	Reference voltage input pin for the 12-bit A/D converter (unit 0) and sample-and-hold circuit for AN000 to AN002. Connect this pin to AVCC0 if these features are not used.
VREFL0	12-bit ADC low reference voltage	Analog reference ground pin for the 12-bit A/D converter (unit 0) and sample-and-hold circuit for AN000 to AN002. Connect this pin to VSS if these features are not used.
VREFH	12-bit ADC and DAC analog supply	Reference voltage input pin for the 12-bit A/D converter (unit 1), sample-and-hold circuit for AN100 to AN102, and D/A converter. This is used as the analog power supply for these modules. Connect this pin to AVCC0 if these features are not used.
VREFL	12-bit ADC and DAC analog ground	Reference ground pin for the 12-bit A/D converter (unit 1), sample-and-hold circuit for AN100 to AN102, and D/A converter. This is used as the analog ground for the respective modules. Connect this pin to VSS if these features are not used.
AVCC_USBHS[1]	USB HS analog power supply	USB High-speed analog power supply.
AVSS_USBHS[1]	USB HS analog ground	USB High-speed analog ground pin. Must be shorted to the PVSS_USBHS pin.
PVSS_USBHS[1]	USB HS PLL circuit ground	Ground pin for the USB High-speed PLL circuit. Must be shorted to the AVSS_USBHS pin.
USBHS_RREF[1]	USB HS current reference	USB High-speed reference current source pin. Connect this pin to the AVSS_USBHS pin through a 2.2 kΩ resistor (±1%)

Note: 1. Only for devices with USB High Speed peripheral.

1.1 References

Further information regarding the power supply for the RA MCU Group can be found in the following documents:

- R01UH0884EJ0100 RA6M1 Group, *RA6M1 Group User's Manual: Hardware*
- R01UH0885EJ0100 RA6M2 Group, *RA6M2 Group User's Manual: Hardware*
- R01UH0886EJ0100 RA6M3 Group, *RA6M3 Group User's Manual: Hardware*
- R01UH0890EJ0110 RA6M4 Group, *RA6M4 Group User's Manual: Hardware*
- R01UH0891EJ0110 RA6M5 Group, *RA6M5 Group User's Manual: Hardware*

Chapter numbers may vary between Arm® Cortex®-M4 and Arm® Cortex®-M33 devices.

The **Overview** chapter lists power pins in each package with recommended bypass capacitors.

The **Resets** chapter discusses the Power-on reset and how to differentiate this from other reset sources.

The **Low Voltage Detection** chapter provides details on the Low-Voltage Detection Circuit that can be used to monitor the power supply. The **Option-Setting Memory** chapter additionally describes how to enable Low-Voltage Detection 0 Circuit automatically at startup.

The **Battery Backup Function** chapter shows how to provide battery backup to the RTC and sub-clock oscillator.

If you plan to use the on-chip Analog to Digital Converters (ADC) or the Digital to Analog Converter (DAC), see *12-Bit A/D Converter (ADC12)* and *12-Bit D/A Converter (DAC12)* for chapters in the respective Hardware User's Manuals for details.

Table 3. RA6 MCU Groups, User's Manual: Hardware

Chapter Name	Description
Overview	Lists power pins in each package with notes on termination and bypassing.
Resets	Discusses the power-on reset and how to differentiate this from other reset sources.
Voltage Detection Circuit	Provides details on the low-voltage detection circuit that can be used to monitor the power supply.
Low Power Modes	Using low power modes may allow you to reduce the voltage of the power supply. See this chapter for details on how operating modes affect power supply requirements.
Battery Backup Function	Shows how to provide battery backup to the RTC and sub-clock oscillator.
12-Bit A/D Converter, 12-bit D/A Converter	If you plan to use the on-chip A/D or D/A converters, these chapters give details on how to provide filtered power supplies for these peripherals.
Clock Generation Circuit	Provides detailed descriptions on how to configure and use the available clocks, including PCB design recommendations.

2. Emulator Support

RA MCU devices have an emulator interface that supports both debugging using SWD or JTAG communication, and serial programming using SCI communication. This emulator makes it easy to switch between debugging and serial programming.

The SWD or JTAG emulator interface should be connected to an Arm® standard 10-pin or 20-pin socket. MD, TXD and RXD pins are added for serial programming using SCI communication.

To comply with the Arm specification, pull-up resistors are required on the JTAG, SWD and SCI signals. Without the correct pull-up resistors, the interface may not function correctly. However, RA6 MCU devices have internal pull up resistors that are enabled by default for these signals. When the internal pull up resistors are enabled, no external resistors are required on these signals.

The serial programming interface must be used to program the Arm® TrustZone® IDAU boundary register settings. For devices that support TrustZone, it is recommended to connect P300/SWCLK/TCK and P201/MD pins using a wired-OR circuit on the board to use both debugging and serial programming.

Emulator support is useful for product development and prototyping but may not be needed once a design moves to production. If emulator support is no longer needed for a design, make sure to configure the ports according to the *Handling of Unused Pins* section of the related MCU Hardware User's Manual. See also section 10.5 in this document.

2.1 SWD Interface

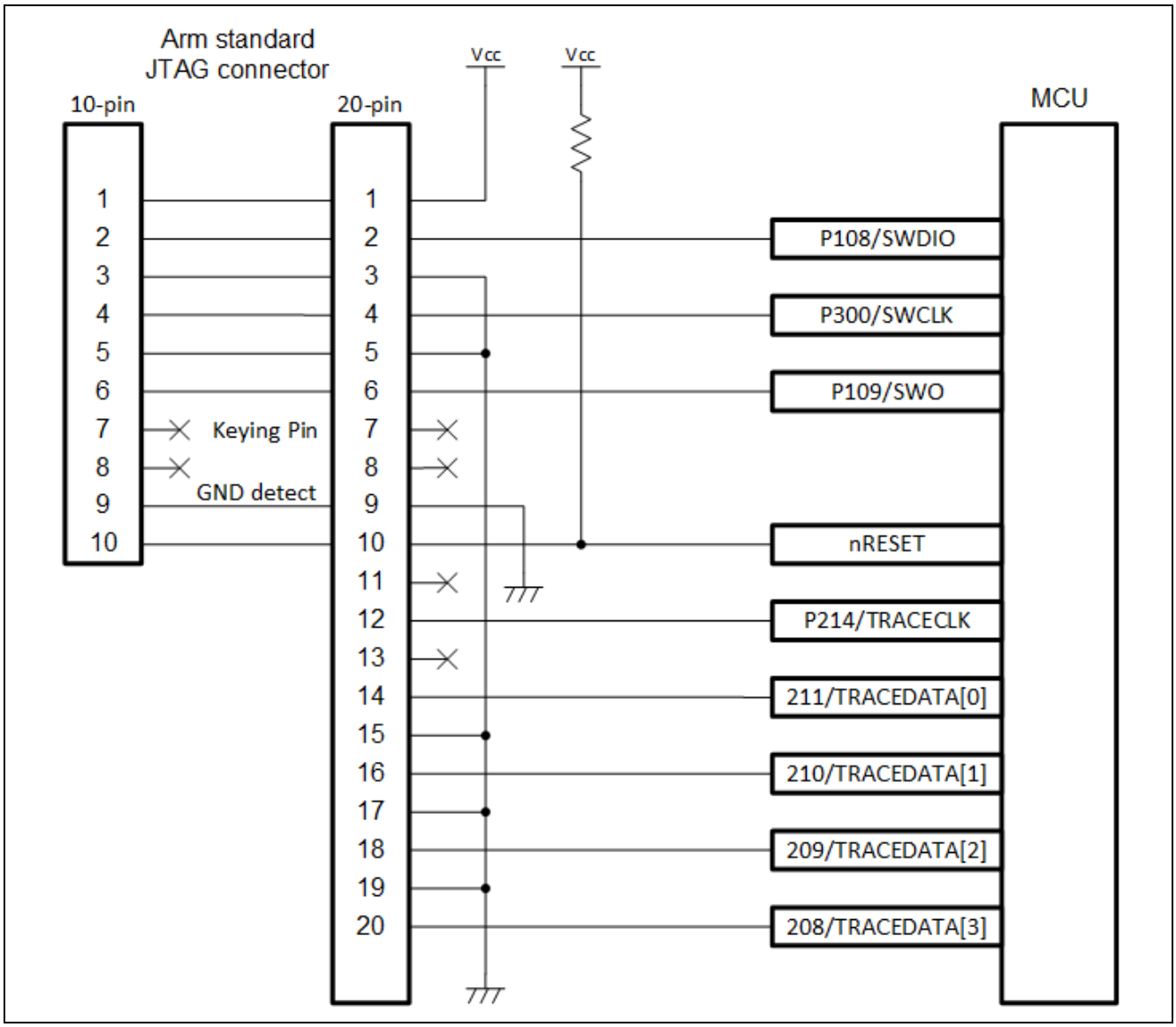

Figure 1. SWD Interface Connections

Note: The output of the reset circuit of the user system must be open collector.

2.2 JTAG Interface

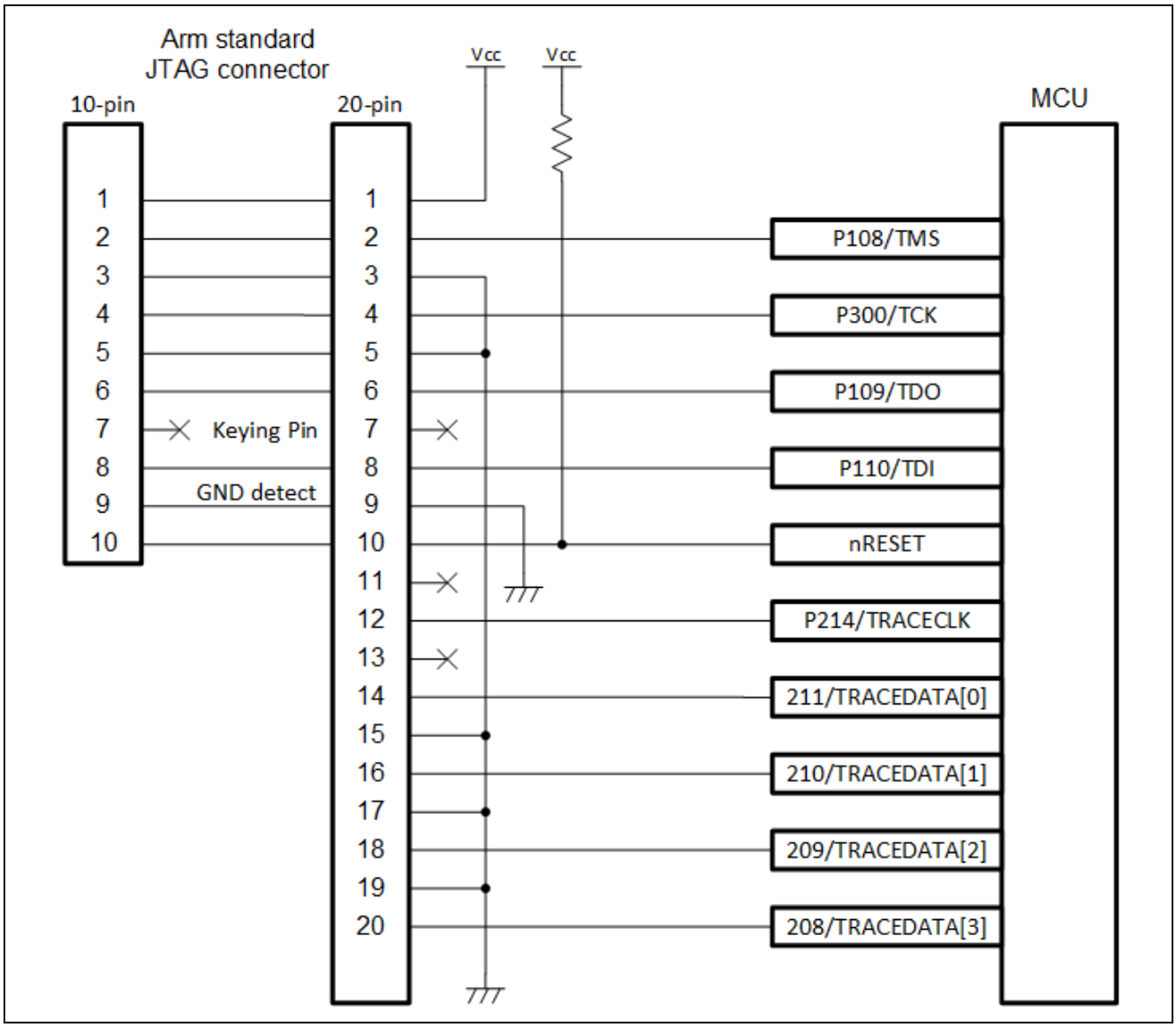

Figure 2. JTAG Interface Connections

Note: The output of the reset circuit of the user system must be open collector.

2.3 Serial Programming Interface using SCI

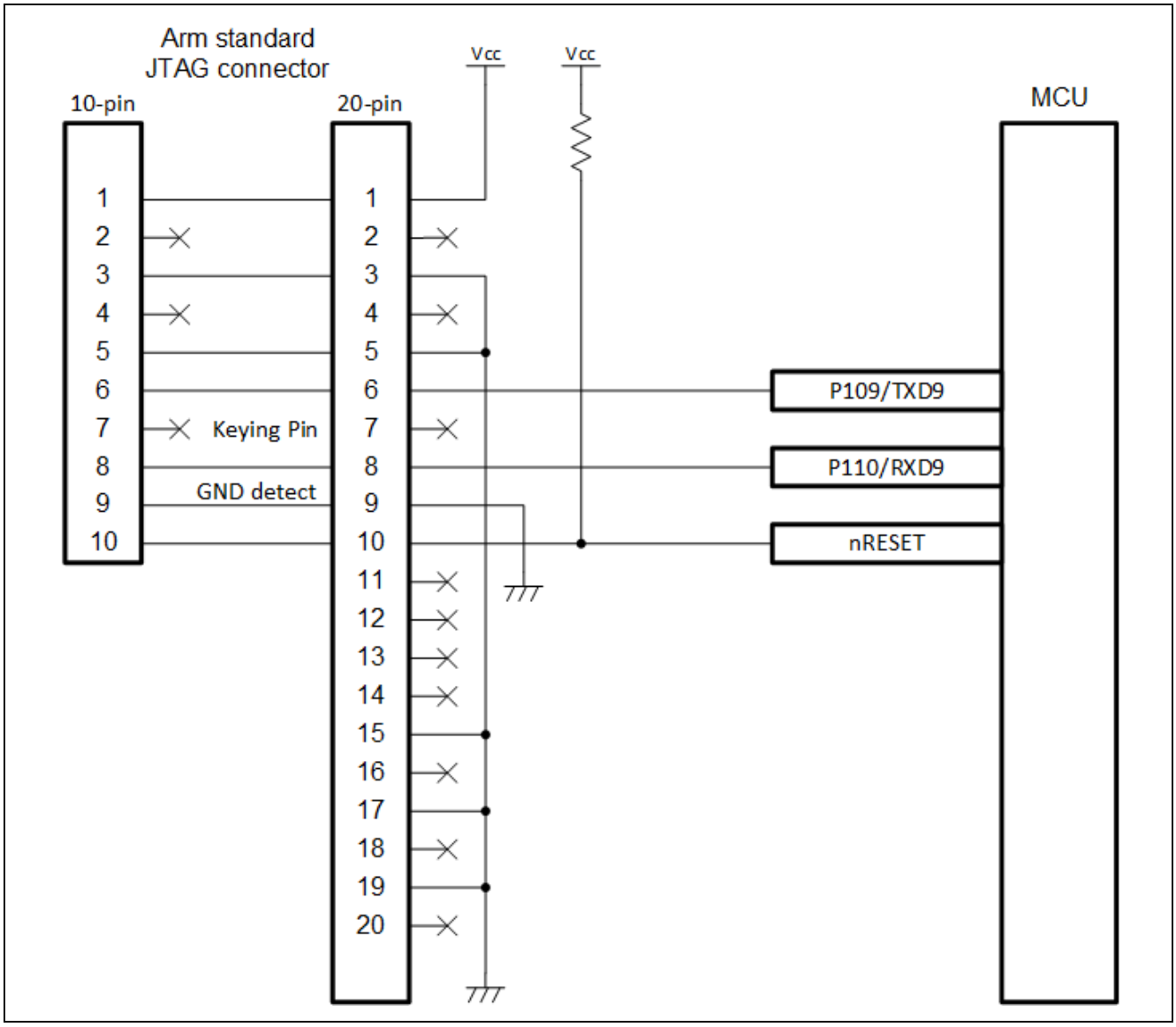

Figure 3. Serial Programming Interface using SCI Connections

Note: The output of the reset circuit of the user system must be open collector.

2.4 Serial Programming Interface using SCI: Devices with TrustZone® Support

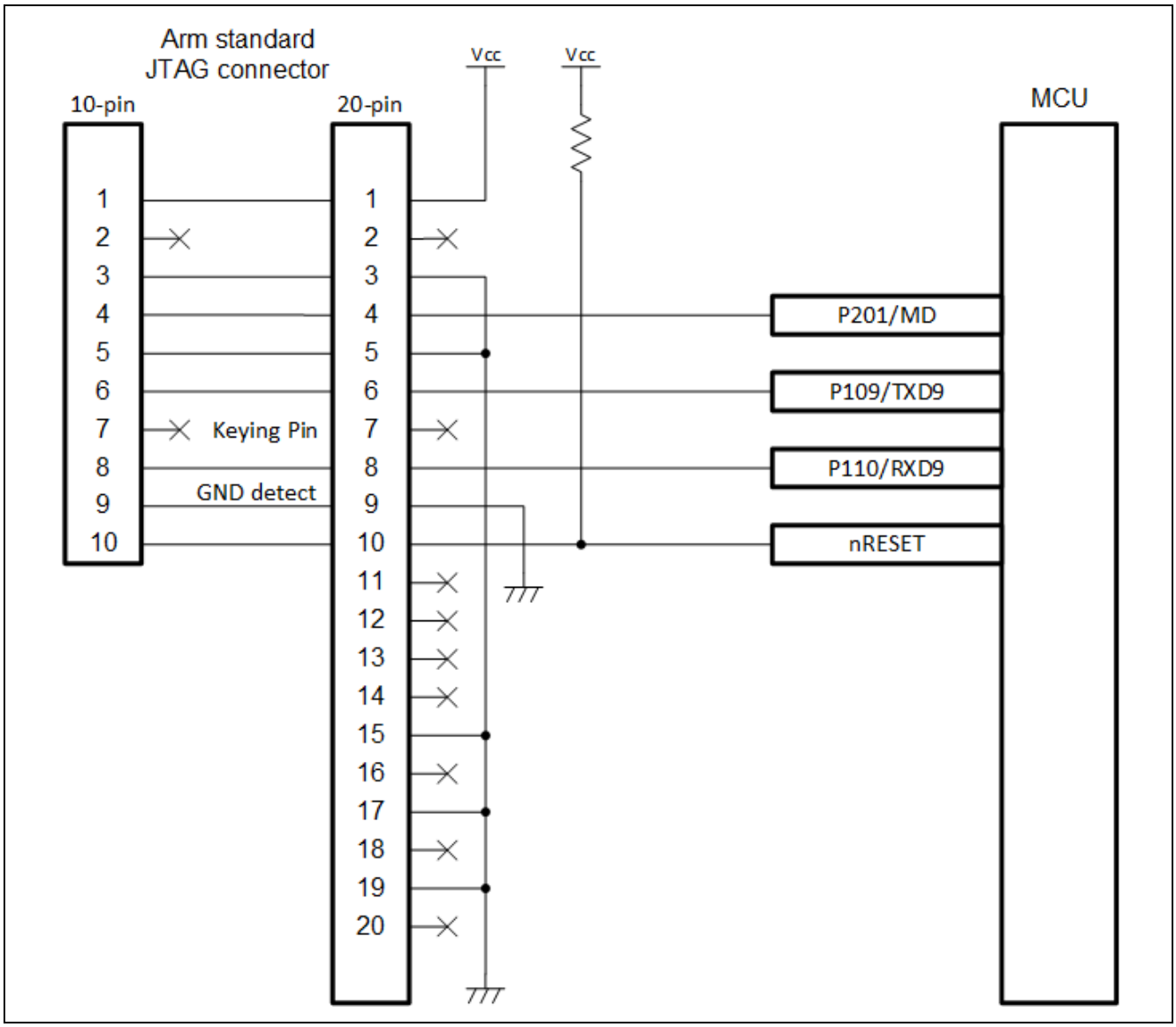

Figure 4. Serial Programming Interface using SCI: Devices with TrustZone® Support Connections

Notes: 1. The output of the reset circuit of the user system must be open collector.
2. P201/MD must be connected to the Arm standard JTAG connector pin 4 to support programing the Arm® TrustZone IDAU boundary register settings.

2.5 Multiple Emulator Interface

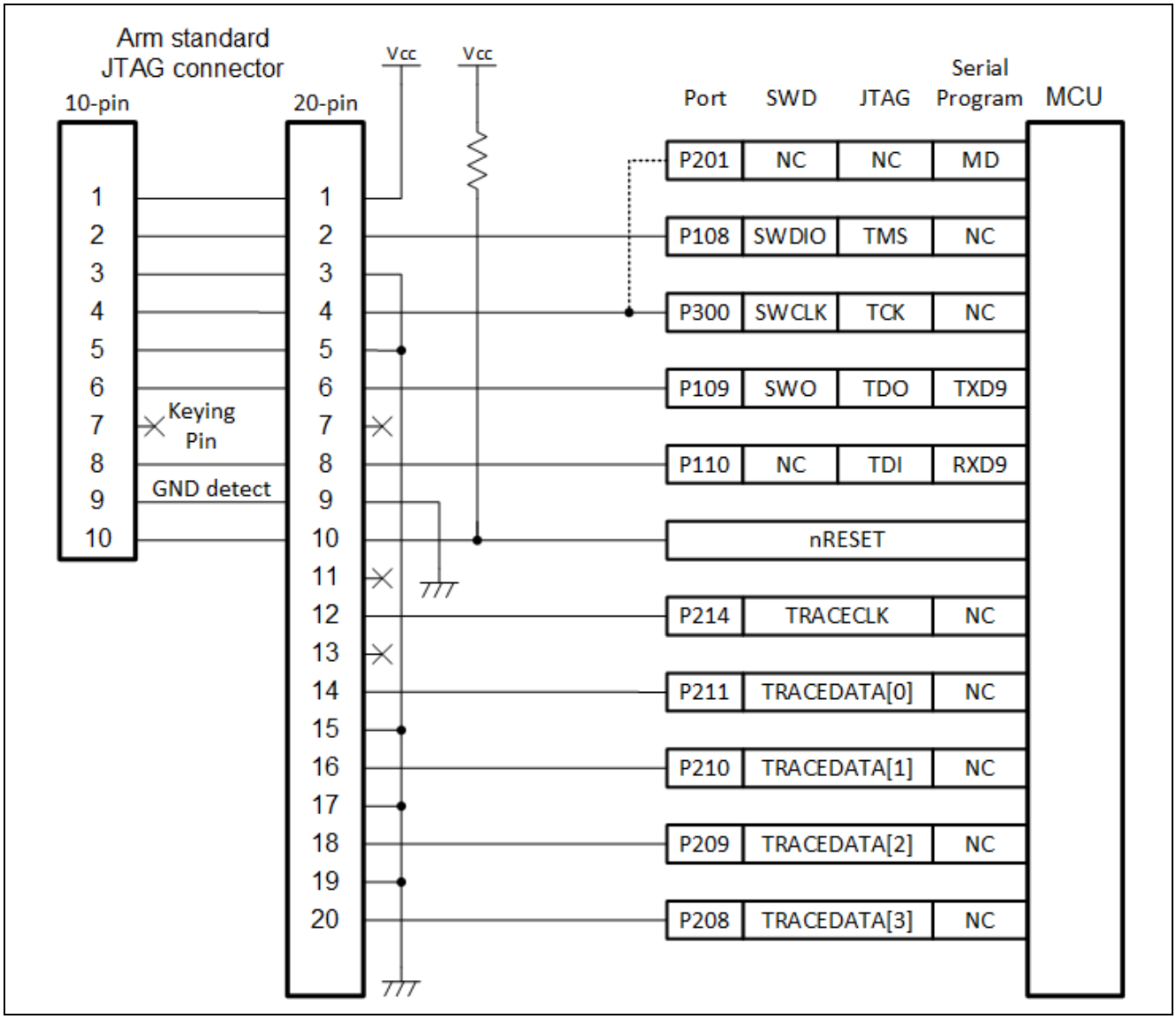

Figure 5. Multiple Emulator Interface Connections

Notes: 1. Reset circuitry on the target must be open-collector. Pull up the nRESET signal. Do not put a capacitor on this signal as it will affect the operation of the power-on reset circuit.
2. To use both debugging and serial programming on devices with TrustZone® support, connect P201/MD to P300/SWCLK/TCK using a wired OR circuit.

P201/MD can be connected to P300/SWCLK to allow the debugger to control the MCU Operating Mode. See section 3 for more details.

2.6 Software Setups for Emulator Connections

2.6.1 SWD and JTAG Interfaces

SWD and JTAG pins are in default state after reset.

Table 4. SWD/JTAG Pins

Pin	P108	P109	P110	P300
Function	TMS/SWDIO	TDO/SWO	TDI	TCK/SWCLK

2.6.2 Trace Port

A 4-bit Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO) provide trace output in RA6 devices.

Trace ports and clock need to be enabled before they can be used by the debugger script. When using the Trace Port functionality, avoid using the trace pins for other functions.

Table 5 lists the Trace Port pins and their associated functions.

Table 5. Trace Ports

Pin	P208	P209	P210	P211	P214
Function	TDATA3	TDATA2	TDATA1	TDATA0	TCLK

For an example of using the Trace Port on RA6M3 with SEGGER J-Trace Pro, refer the following link:

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

Trace ports can also be enabled at runtime by using Pin Configurator in Renesas FSP but some trace data may be lost in this case.

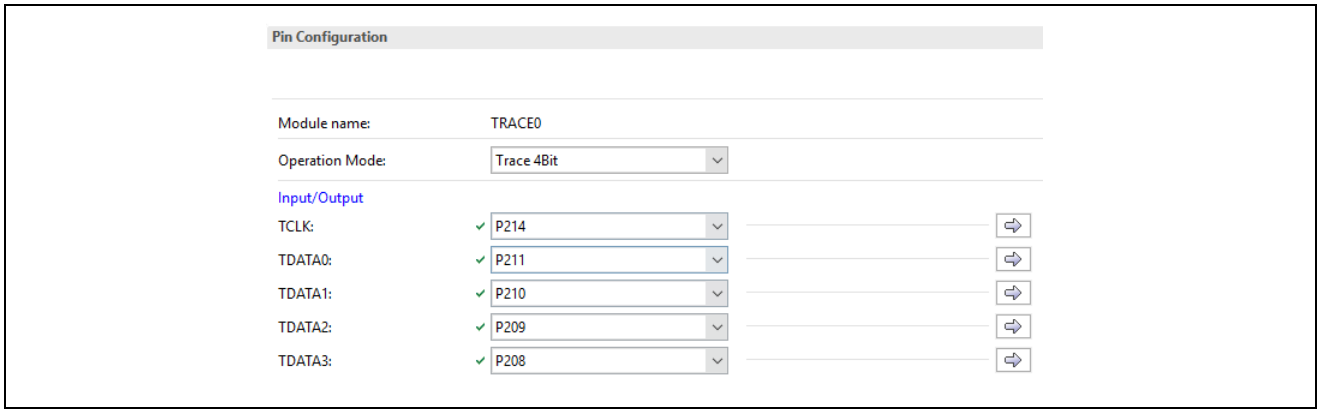

Figure 6. Enabling Trace Ports in Runtime Using FSP Configurator

3. MCU Operating Modes

The RA6 MCU series can enter one of two modes after reset: Single-chip mode or SCI/USB boot mode. The boot mode is selected by the MD pin:

Table 6. Operating Modes Available at Reset

Operating Mode	MD	On-Chip Flash Memory	External Bus
Single-chip mode	1	Enable	Disable
SCI/USB boot mode	0	Enable	Disable

Figure 7 shows operating mode transitions as determined by the Mode-Setting (MD) pin.

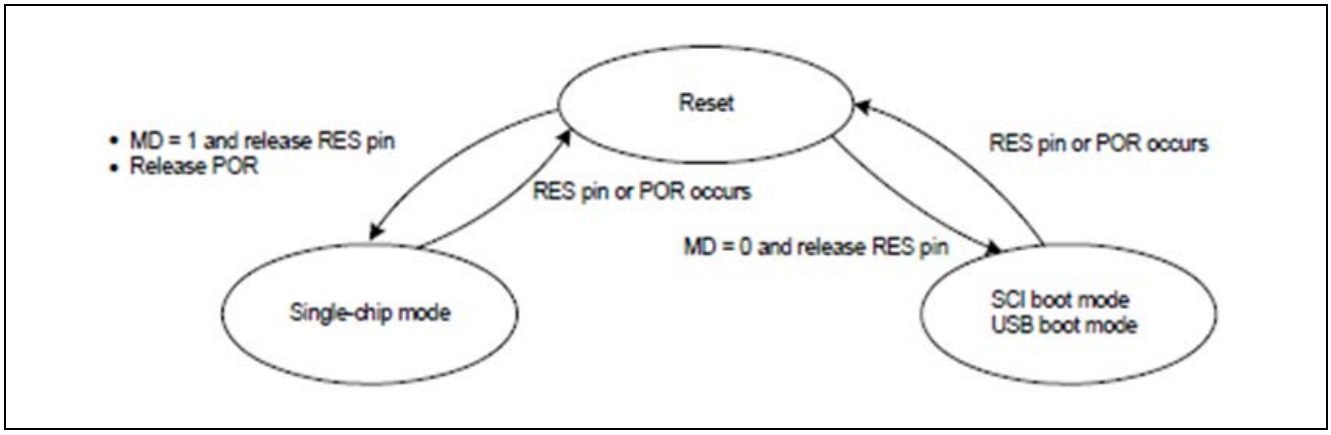

Figure 7. Mode Setting Pin level and Operating Mode

A typical MCU boot mode circuit includes a jumper and a couple of resistors to allow selections to connect the MD pin to VCC or Ground. RA6 MCU devices include an internal pull up resistor on P201/MD, which is enabled by default. The internal pull up resistor may replace the external pull up resistor in Figure 8.

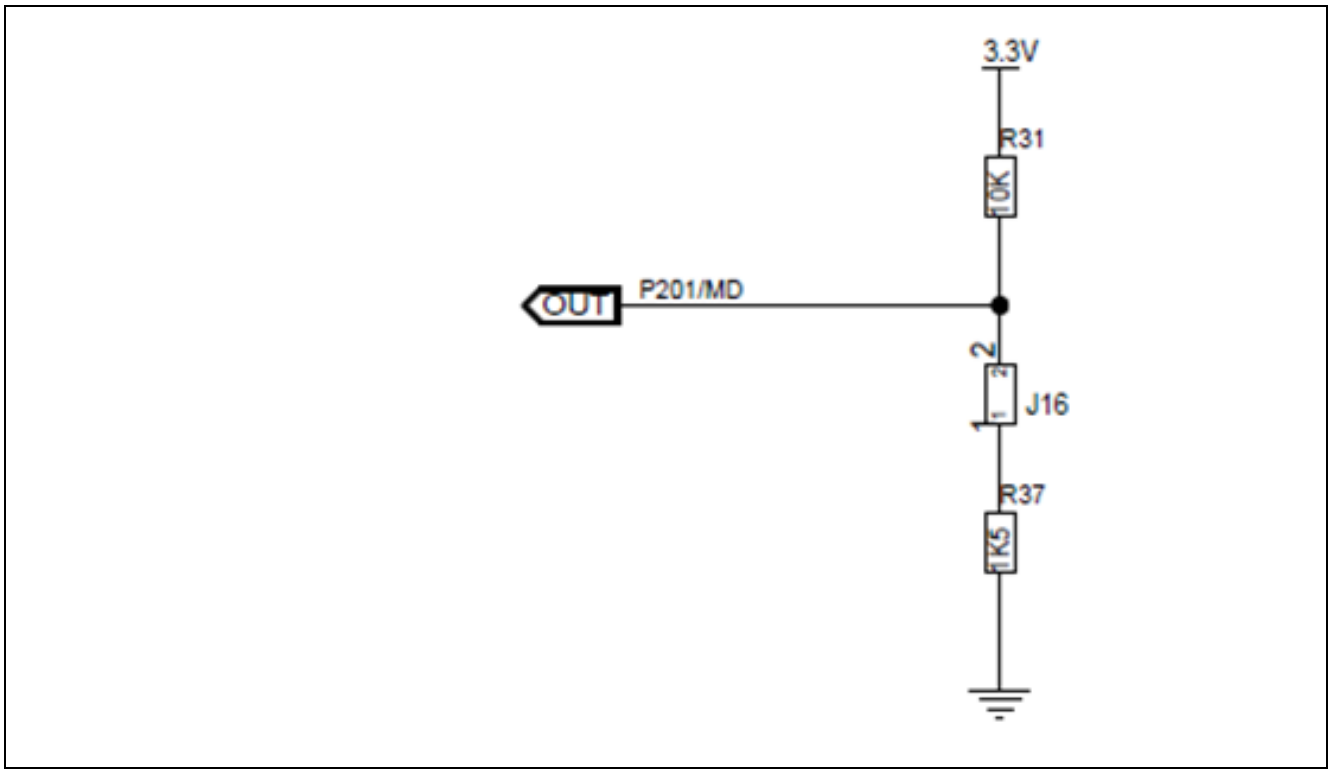

Figure 8. Typical Circuit for MCU Boot Mode Selection

4. Option Setting Memory

The option-setting memory determines the state of the MCU after a reset. It is allocated to the configuration setting area and the program flash area of the flash memory. The available methods of setting are different for the two areas. Option setting memory may be different in size and layout for Arm® Cortex®-M33 based devices.

The registers are detailed in the *Option Setting Memory* chapter in the Hardware User's Manual.

The flash option registers occupy space in the code flash memory map. Although the registers are located in a portion of the flash memory that was reserved on the RA MCUs, **it is possible that some customers may store data in these locations inadvertently**. The user must check to ensure that no unwanted data is written to these locations or else unexpected behavior of the chip may result. For instance, settings in the flash option registers can enable the Independent Watchdog Timer (IWDT) immediately after reset. If data stored in program ROM inadvertently overlaps the option setting memory register, it is possible to turn on the IWDT on without realizing it. This will cause the debugger to have communications problems with the board.

The following figure shows the option setting memory which consists of the option function select registers on RA6M3, which is an Arm Cortex-M4 device.

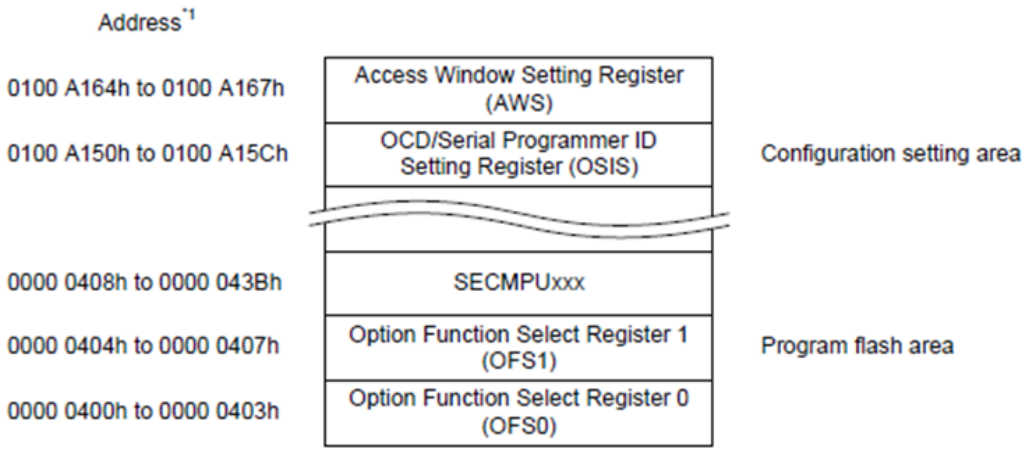

Figure 9. Option Function Select Registers for RA6M3

4.1 Option Setting Memory Registers

Following is a summary of the Option Setting Memory registers. Make sure that they are configured properly before startup.

- OFS0 register
  - Independent Watchdog Timer (IWDT) auto start
  - IWDT timeout, frequency, windowing, interrupt type, and low power mode behavior
  - Watchdog Timer (WDT) auto start
  - WDT timeout, frequency, windowing, and interrupt type
- OFS1 register
  - LVD0 enable after reset
  - HOCO startup after reset

Renesas FSP Configurator supports setting of option memory in BSP settings, as shown in the following figure for RA6M3 MCU.

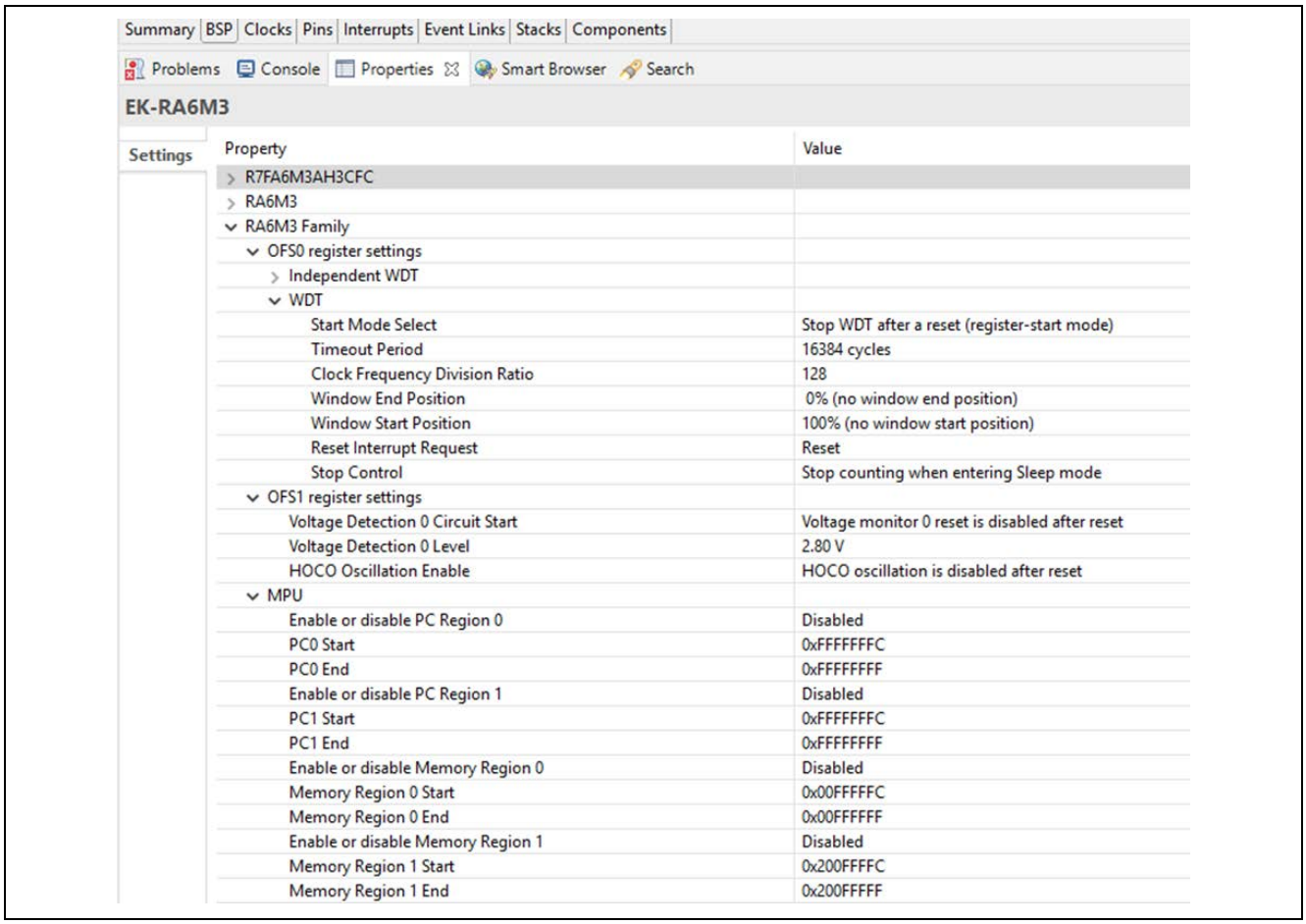

Figure 10. Option Memory Settings in FSP Configuration for RA6M3 MCU

5. Clock Circuits

The RA6 MCUs have six primary oscillators. Five of these may be used as the source for the main system clock. The remaining oscillator is dedicated to the Independent Watchdog Timer. In a typical system, the main clock is driven with an external crystal or clock. This input is directed to the PLL where it is multiplied up to the PLL clock, then post-divided down into the main system clock (ICLK), flash clock, peripheral module clocks, external bus clock, trace clock, and USB clock. The ICLK is further used for the CPU clock, DMAC clock, internal flash clock and SDRAM clock. Refer to the *Clock Generation Circuit* chapter in Hardware User's Manual for the block diagram of the clock generation circuit.

Each clock has specific tolerances and timing values. Refer to the *AC Characteristics* section in the *Electrical Characteristics* chapter in the Hardware User's Manual for the Frequency and Clock Timing specifications. Refer to the *Clock Generation Circuit* chapter in the Hardware User's Manual for the relationship between the various clock frequencies.

Table 7. RA6 Oscillators

Oscillator	Input Source	Frequency	Primary Uses
Main clock	External crystal/resonator or External clock	8 MHz to 24 MHz Up to 24 MHz	PLL input, main system clock, CLKOUT, CAN clock, CAC clock
Sub-clock (SOSC)	External crystal/resonator	32.768 kHz	Real-time clock, system clock in low power modes, CLKOUT, AGT clock, CAC clock
High-speed on-chip (HOCO)	On-chip oscillator	16/18/20 MHz	PLL input, main system clock, CLKOUT, CAC clock

Oscillator	Input Source	Frequency	Primary Uses
Middle-speed on-chip (MOCO)	On-chip oscillator	8 MHz	System clock at startup, CLKOUT, CAC clock
Low-speed on-chip (LOCO)	On-chip oscillator	32.768 kHz	Main system clock in low power modes, and during main oscillator stop detection, AGT clock, CAC clock, Real-time clock
Independent Watchdog (IWDT)	On-chip oscillator	15 kHz	Independent watchdog timer clock

5.1 Reset Conditions

After reset, RA6 MCUs begin running with the middle-speed on-chip oscillator (MOCO) as the main clock source. At reset, the main oscillator and the PLL are off by default. The HOCO and IWDT may be on or off depending on the settings in the Option Setting Memory (see section 4).

5.2 Clock Frequency Requirements

The ICLK must always be greater than or equal to the BCLK. Minimum and maximum frequencies are shown in the following tables. Details can be found in the *Overview* section of the *Clock Generation Circuit* chapter in the MCU Hardware User's Manual, including external and internal clock source specifications. Additional details can be found in the *AC Characteristics* section of the *Electrical Characteristics* chapter in the MCU Hardware User's Manual.

Table 8. Frequency Range for Arm Cortex-M4 MCU Internal Clocks

	ICLK [1]	PCLKA [1]	PCLKB	PCLKC	PCLKD
Maximum Frequency [MHz]	120	120	60	60	120
Minimum Frequency [MHz]	—	—	—	—	—

Note 1: The ICLK and PCLKA frequencies must be the same and at least 12.5 MHz if the Ethernet controller is in use

	FCLK [1]	BCLK	UCLK	CANCLK	SDCLK
Maximum Frequency [MHz]	60	120	48	24	120
Minimum Frequency [MHz]	4	—	48	8	—

Note 1: The FCLK must run at a frequency of at least 4 MHz when writing or erasing ROM or data flash.

Table 9. Frequency Range for Arm® Cortex®-M33 MCU Internal Clocks

	ICLK [1]	PCLKA [1]	PCLKB	PCLKC	PCLKD
Maximum Frequency [MHz]	200	100	50	50	100
Minimum Frequency [MHz]	—	—	—	—	—

Note 1: The ICLK and PCLKA frequencies must be the same and at least 12.5 MHz if the Ethernet controller is in use

	FCLK [1]	BCLK	USBCLK	CANCLK	OCTACLK
Maximum Frequency [MHz]	50	100	48	24	200
Minimum Frequency [MHz]	4	—	48	8	—

Note 1: The FCLK must run at a frequency of at least 4 MHz when writing or erasing ROM or data flash.

5.2.1 Requirements for USB Communications

The USB 2.0 Full-Speed Module (USBFS) and USB 2.0 High-Speed Module (USBHS) available on some members of the RA family require a 48 MHz USB clock signal (UCLK or USBCLK). When USB is used and the HOCO is selected as the clock source for the PLL, the Frequency Locked Loop (FLL) function must be enabled.

When using the USB, the main clock oscillator frequency is limited to the following choices: 8 MHz, 10 MHz, 12 MHz, 15 MHz, 16 MHz, 20 MHz, or 24 MHz. This is due to the specific division ratios available in the clock generation circuit and the 48 MHz clock required by the USB modules.

For Arm® Cortex®-M4 devices, the divider used depends on the setting of the UCK bits in the SCKDIVCR2 register.

For Arm Cortex-M33 devices, the divider used depends on the setting of the USBCKDIV bits in the USBCKDIVCR register.

5.2.2 Requirements for Ethernet Controller

For Arm Cortex-M4 devices, when the Ethernet controller (EtherC) and Ethernet DMA Controller (EDMAC) are used, PCLKA (Ethernet) must be the same as ICLK, and both must be in the range 12.5 MHz to 120 MHz.

For Arm Cortex-M33 devices, when the Ethernet controller (EtherC) and Ethernet DMA Controller (EDMAC) are used, PCLKH (Ethernet) must be in the range 12.5 MHz to 100 MHz.

5.2.3 Requirements for Programming and Erasing ROM or Data Flash

The FCLK must be at least 4 MHz to perform programming and erasing on internal ROM and data flash.

5.2.4 Requirements for SDRAM Controller

The SDCLK is sourced from the BCLK. Do not set SDCLK to a frequency higher than the system clock (ICLK).

5.3 Lowering Clock Generation Circuit (CGC) Power Consumption

To aid in saving power, set the dividers for any unused clocks (for example, BCLK) to the highest possible value whenever possible. Also, if not using a clock, then make sure that it has been stopped by setting the appropriate register(s). The registers for controlling each clock source are shown in the following table.

Table 10. Clock Source Configuration Registers

Oscillator	Register	Description
Main clock	MOSCCR	Starts/stops main clock oscillator
Sub-clock	SOSCCR	Starts/stops sub-clock oscillator
High-speed on-chip (HOCO)	HOCOCR	Starts/stops HOCO
Middle-speed on-chip (MOCO)	MOCOCR	Starts/stops MOCO
Low-speed on-chip (LOCO)	LOCOCR	Starts/stops LOCO

5.4 Writing the System Clock Control Registers

Care should be taken when writing to the individual bit fields in the System Clock Division Control Register (SCKDIVCR), System Clock Division Control Register 2 (SCKDIVCR2), and System Clock Source Control Register (SCKSCR).

When changing any value in SCKDIVCR or SCKDIVCR2 from a lower division ration to a higher division ratio, wait at least 750 ns before changing the value. When changing any value from a higher division ratio to a lower division ratio, wait at least 250 ns after changing the value before starting subsequent processing.

When changing the value of SCKSCR from the PLL to a different clock source, wait at least 750 ns before changing the value. When changing the value from a non-PLL clock source to the PLL, wait at least 250 ns after changing the value before starting subsequent processing.

The recommended method to measure the wait time is to do so in software by counting instruction cycles. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses.

5.5 Clock Setup Example

Renesas FSP provides a simple, visual clock configuration tool for RA6 MCUs shown as follows.

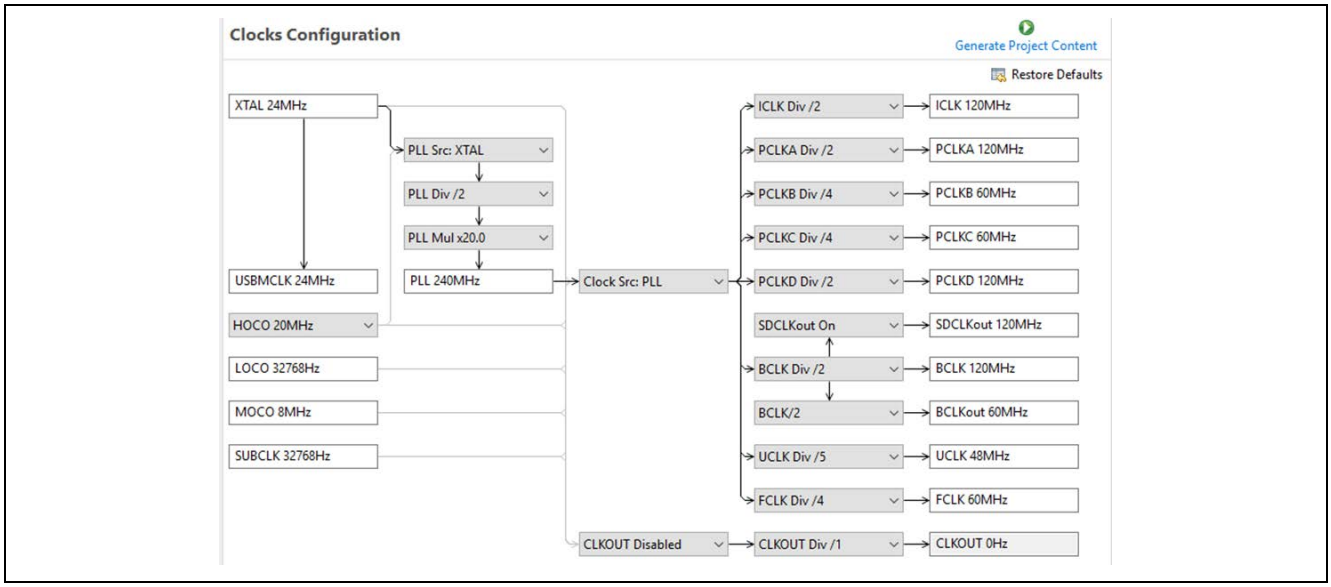

Figure 11. Clock Settings Using Renesas FSP Configurator

5.6 HOCO Accuracy

The internal high-speed on-chip oscillator (HOCO) runs at 16 MHz, 18 MHz, or 20 MHz, with an accuracy of +/-2% or better. The accuracy of the HOCO may be improved by enabling the Frequency Locked Loop (FLL) function, which results in a clock accuracy of +/-0.3% or better. Refer to the *Electrical Specifications* in the hardware manual for details.

The HOCO may be used as an input to the PLL circuit. When the HOCO is used this way, no external oscillator is required. This may be an advantage when space constraints or other limitations require a reduced component count in a PCB design. However, there are performance tradeoffs and limitations due to the clock accuracy, which should be evaluated for your application.

5.7 Flash Interface Clock

The Flash interface Clock (FCLK) is used as the operating clock when programming and erasing internal flash (ROM and Data Flash) and for reading from the data flash. Therefore, the frequency setting of the FCLK will have a direct impact on the amount of time it takes to read from the data flash. If the user's program is reading from the data flash, or performing programming or erasures on internal flash, then using the maximum FCLK frequency is recommended.

Please note that the FCLK frequency does not have any impact upon reading from ROM or reading and writing to RAM.

5.8 Board Design

Refer to the *Usage Notes* section of the *Clock Generation Circuit (CGC)* chapter in the Hardware User's Manual for more information on using the CGC and for board design recommendations.

When a crystal resonator is used, place the crystal resonator and its load capacitors as close to the MCU clock pins (XTAL/EXTAL, XCIN/XCOUT) as possible. Avoid routing any other signals between the crystal resonator and the MCU. Minimize the number of connecting vias used on each trace.

5.9 External Crystal Resonator Selection

An external crystal resonator may be used as the main clock source. The external crystal resonator is connected across the EXTAL and XTAL pins of the MCU. The frequency of the external crystal resonator must be in the frequency range of the main clock oscillator.

Selection of a crystal resonator will be largely dependent on each unique board design. Due to the large selection of crystal resonators available that may be suitable for use with RA6 MCU devices, carefully evaluate the electrical characteristics of the selected crystal resonator to determine the specific implementation requirements.

The following graphic shows a typical example of a crystal resonator connection.

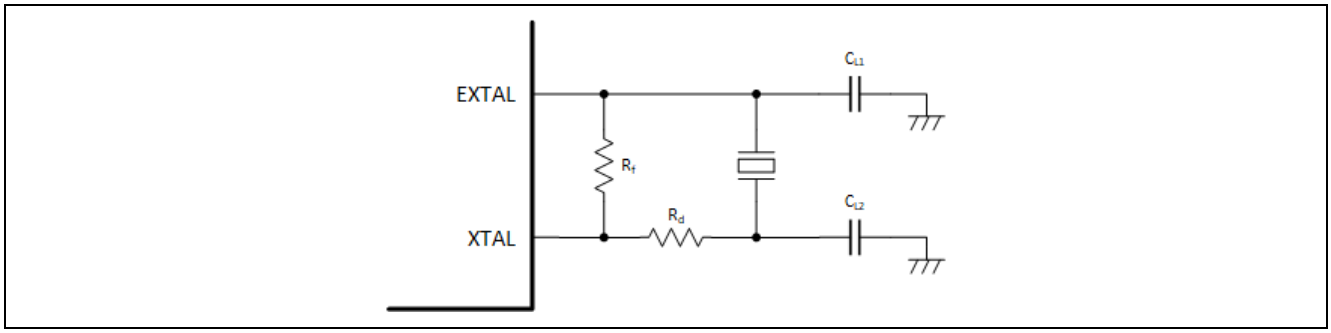

Figure 12. Example of Crystal Resonator Connection

Careful evaluation must be used when selecting the crystal resonator and the associated capacitors. The external feedback resistor (R_f) and damping resistor (R_d) may be added if recommended by the crystal resonator manufacturer.

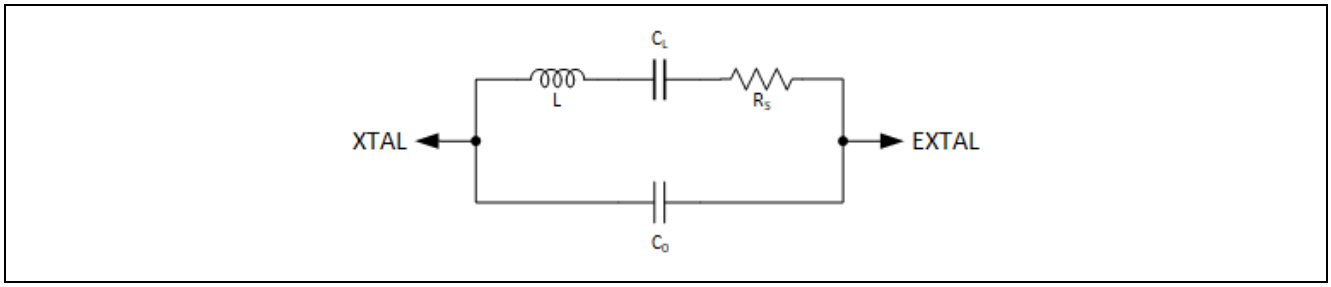

Figure 13. Equivalent Circuit of the Crystal Resonator

Selection of the capacitor values for CL1 and CL2 will affect the accuracy of the internal clock. To understand the impact of the values for CL1 and CL2, the circuit should be simulated using the equivalent circuit of the crystal resonator in the figure above. For more accurate results, also take in to account the stray capacitance associated with the routing between the crystal resonator components.

6. Reset Requirements and the Reset Circuit

There are fourteen types of resets. These resets differ slightly between Arm® Cortex®-M4 devices and Arm Cortex-M33 devices.

Table 11. Arm Cortex-M4 Device Resets

Reset Name	Source
Pin reset	RES# is driven low
Power-on reset	VCC rises (voltage detection: VPOR)
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh occurs
Watchdog timer reset	The watchdog timer underflows, or a refresh occurs
Voltage monitor 0 reset	VCC falls (voltage detection Vdet0)
Voltage monitor 1 reset	VCC falls (voltage detection Vdet1)
Voltage monitor 2 reset	VCC falls (voltage detection Vdet2)
SRAM parity error reset	SRAM parity error detection
SRAM ECC error reset	SRAM ECC error detection
Bus master MPU error reset	Bus master MPU error detection
Bus slave MPU error reset	Bus slave MPU error detection
Stack pointer error reset	Stack pointer error detection
Deep software standby reset	Deep Software Standby mode is canceled by an interrupt
Software reset	Register setting

Table 12. Arm Cortex-M33 Device Resets

Reset Name	Source
Pin reset	RES# is driven low
Power-on reset	VCC rises (voltage detection: VPOR)
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh occurs
Watchdog timer reset	The watchdog timer underflows, or a refresh occurs
Voltage monitor 0 reset	VCC falls (voltage detection Vdet0)
Voltage monitor 1 reset	VCC falls (voltage detection Vdet1)
Voltage monitor 2 reset	VCC falls (voltage detection Vdet2)
SRAM parity error reset	SRAM parity error detection
SRAM ECC error reset	SRAM ECC error detection
Bus master MPU error reset	Bus Master MPU error detection
TrustZone error reset	TrustZone error detection
Cache Parity error reset	Cache Parity error detection
Deep software standby reset	Deep Software Standby mode is canceled by an interrupt
Software reset	Register setting

6.1 Pin Reset

When the RES# pin is driven low, all processing is aborted and the MCU enters a reset state. To reset the MCU while it is running, RES# should be held low for the specified reset pulse width. Refer to the *Reset Timing* section of the *Electrical Characteristics* chapter of the Hardware User's Manual for more detailed timing requirements. Also refer to section 2 of this document, *Emulator Support* for details on reset circuitry in relation to debug support.

There is no need to use an external capacitor on the RES# line because the POR circuit holds it low internally for a good reset and a minimum reset pulse is required to initiate this process.

6.2 Power-On Reset

There are two conditions that will generate a power-on reset (POR):

1. If the RES# pin is in a high-level state when power is supplied.
2. If the RES# pin is in a high-level state when VCC is below V_{POR}.

After VCC has exceeded the power on reset voltage (V_{POR}) and the power-on reset time (t_{POR}) has elapsed, the chip is released from the power-on reset state. The power-on reset time is a period that allows for stabilization of the external power supply and the MCU. Refer to the *POR and LVD Characteristics* section of the *Electrical Characteristics* chapter of the Hardware User's Manual for voltage level and timing details.

Because the POR circuit relies on having RES# high concurrently with VCC, don't place a capacitor on the reset pin. This will slow the rise time of RES# in relation to VCC, preventing the POR circuit from properly recognizing the power-on condition.

If the RES# pin is high when the power supply (VCC) falls to or below V_{POR} , a power-on reset is generated. The chip is released from the power-on state after VCC has risen above V_{POR} and the t_{POR} has elapsed.

After a power on reset, the PORF bit in RSTSR0 is set to 1. Following a pin reset PORF is cleared to 0.

6.3 Independent Watchdog Timer Reset

This is an internal reset generated by the Independent Watchdog Timer (IWDT).

When the IWDT underflows, an independent watchdog timer reset is optionally generated (NMI can be generated instead) and the IWDTRF bit in RSTSR1 is set to a 1. After a short delay (typically 320 µs) the IWDT reset is canceled.

6.4 Watchdog Timer Reset

This is an internal reset generated by the Watchdog Timer (WDT).

When the WDT overflows, a watchdog timer reset is optionally generated (NMI can be generated instead), and the WDTRF bit in RSTSR1 is set to a 1. After a short delay (typically 320 µs) the WDT reset is canceled.

6.5 Voltage-Monitoring Resets

The RA6 group includes circuitry that allows the MCU to protect against unsafe operation during brownouts. On-board comparators check the supply voltage against three reference voltages, V_{det0}, V_{det1} and V_{det2}. As the supply dips below each reference voltage, an interrupt or a reset can be generated. The detection voltages V_{det0}, V_{det1} and V_{det2} are each selectable from 3 different levels.

When VCC subsequently rises above V_{det0}, V_{det1}, or V_{det2}, release from the voltage-monitoring reset proceeds after a stabilization time has elapsed.

Low Voltage Detection is disabled after a power on reset. Voltage monitoring can be enabled by using the Option Function register OFS1. For more details, see the chapter *Low Voltage Detection (LVD)* in the Hardware User's Manual.

After an LVD Reset, the LVDnRF (n = 0, 1, 2) bit in RSTSR0 is set to 1.

6.6 Deep Software Standby Reset

This is an internal reset generated when Deep Software Standby mode is canceled by an interrupt.

When Deep Software Standby mode is canceled, a Deep Software Standby reset is generated, and clock oscillation starts. On receiving the interrupt, after the Deep Standby Cancellation Wait Time (tDSBYWT 34-35 clock cycles) has elapsed, reset is canceled, and normal processing starts. For details of the deep software standby mode refer to the *Low Power Modes* chapter in the Hardware User's Manual.

After a Deep Software Standby Reset, the DPSRSTF bit in RSTSR0 is set to 1.

6.7 Software Reset

This is an internal reset generated by writing 0xA501 to the SWRR register. The internal reset time when using software reset is a maximum of 960 µs. When using software reset, make sure that the watchdogs are serviced first before issuing the software reset command.

When a software reset is generated, the SWRF pin in RSTSR1 is set to a 1. After a short delay (typically 320 µs) the software reset is canceled.

6.8 Other Resets

Most peripheral functions within the MCU can generate a reset under specific fault conditions. No hardware configuration is required to enable these resets. Refer to the relevant chapters in the Hardware User's Manual for details of the conditions that will generate a reset for each peripheral function.

6.9 Determination of Cold/Warm Start

The RA6 MCUs allow the user to determine the cause of the reset processing. The CWSF flag in RSTSR2 indicates whether a power on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start.)

The flag is set to 0 when a power-on reset occurs. Otherwise, the flag is not set to 0. The flag is set to 1 when 1 is written to it through software. It is not set to 0 even on writing 0 to it.

6.10 Determining the Reset Source

The RA6 MCUs allow the user to determine the reset signal generation source. Read RSTSR0, RSTSR1, and RSTSR0 to determine which reset was the source of the reset. Refer to the Hardware User's Manual section *Determination of Reset Generation Source* for the flow diagram.

The following code sample shows how to determine if a reset is caused by software reset, Deep Software Standby or power-on reset using CMSIS based register structure in Renesas FSP.

```
/* 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. TrustZone® Enable

7.1 Implementation of Arm® TrustZone Technology

Some RA6 MCUs, such as RA6M4 and RA6M5, include Arm TrustZone (TZ) security features. For full details of TZ implementation, please refer to the Arm documentation (https://developer.arm.com/ip-products/security-ip/trustzone) or the appropriate Hardware User's Manual.

Arm TZ technology divides the MCU and therefore the application into Secure and Non-Secure partitions. Secure applications can access both Secure and Non-Secure memory and resources. Non-Secure code can access Non-Secure memory and resources as well as Secure resources through a set of so-called veneers located in the Non-Secure Callable (NSC) region. This ensures a single access point for Secure code when called from the Non-Secure partition. The MCU starts up in the Secure partition by default. The security state of the CPU can be either Secure or Non-Secure. The MCU code flash, data flash, and SRAM are divided into Secure (S) and Non-Secure (NS) regions. Code flash and SRAM include a further region known as Non-Secure Callable (NSC). These memory security attributes are set into the non-volatile memory via SCI or USB boot mode commands when the device lifecycle is Secure Software Debug (SSD) state. The memory security attributes are loaded into the Implementation Defined Attribution Unit (IDAU) peripheral and the memory controller before application execution and cannot be updated by application code.

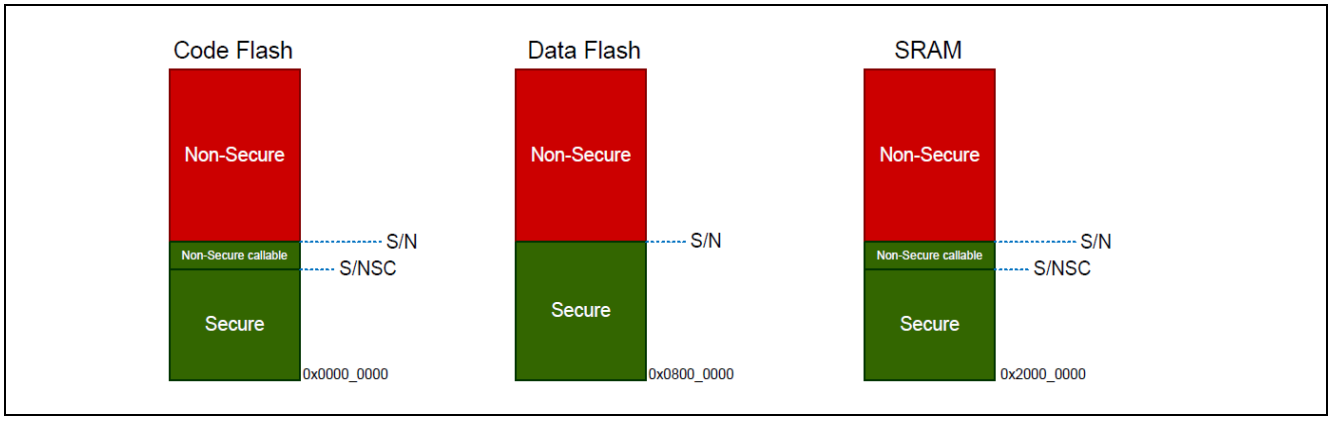

Figure 14. Secure and Non-Secure Regions

Note: All external memory accesses are considered to be Non-Secure.

Code Flash and SRAM can be divided into Secure, Non-Secure, and Non-Secure Callable. All secure memory accesses from the Non-Secure region MUST go through the Non-Secure Callable gateway and target a specific Secure Gateway (SG) assembler instruction. This forces access to Secure APIs at a fixed location and prevents calls to sub-functions and so on. Failing to target an SG instruction will generate a TZ exception. TZ enabled compilers will manage generation of the NSC veneer automatically using CMSE extensions

7.2 Emulator Support for TrustZone

Renesas provides an emulator which supports both debugging using SWD or JTAG communication and serial programming using SCI communication. This emulator makes it easy to switch between debugging and serial programming. The following table shows the pinout of 10-pin or 20-pin socket when using this emulator. The pinout of SWD and JTAG is Arm standard, and MD, TXD, RXD pins are added for the serial programming using SCI communication.

The serial programming interface must be used to program the TrustZone IDAU boundary register settings. It is recommended to connect P300/SWCLK/TCK and P201/MD pins using wired OR circuit on the board to use both debugging and serial programming. If the serial programming interface is not used, designs using the Ethernet or EDMAC peripherals may not work correctly.

Table 13. Pin Assignments for Emulator

Pin No.	SWD	JTAG	Serial Programming Using SCI
1	VCC	VCC	VCC
2	P108/SWDIO	P108/SWDIO	NC
4	P300/SWCLK Wired OR with P201/MD	P300/TCK Wired OR with 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

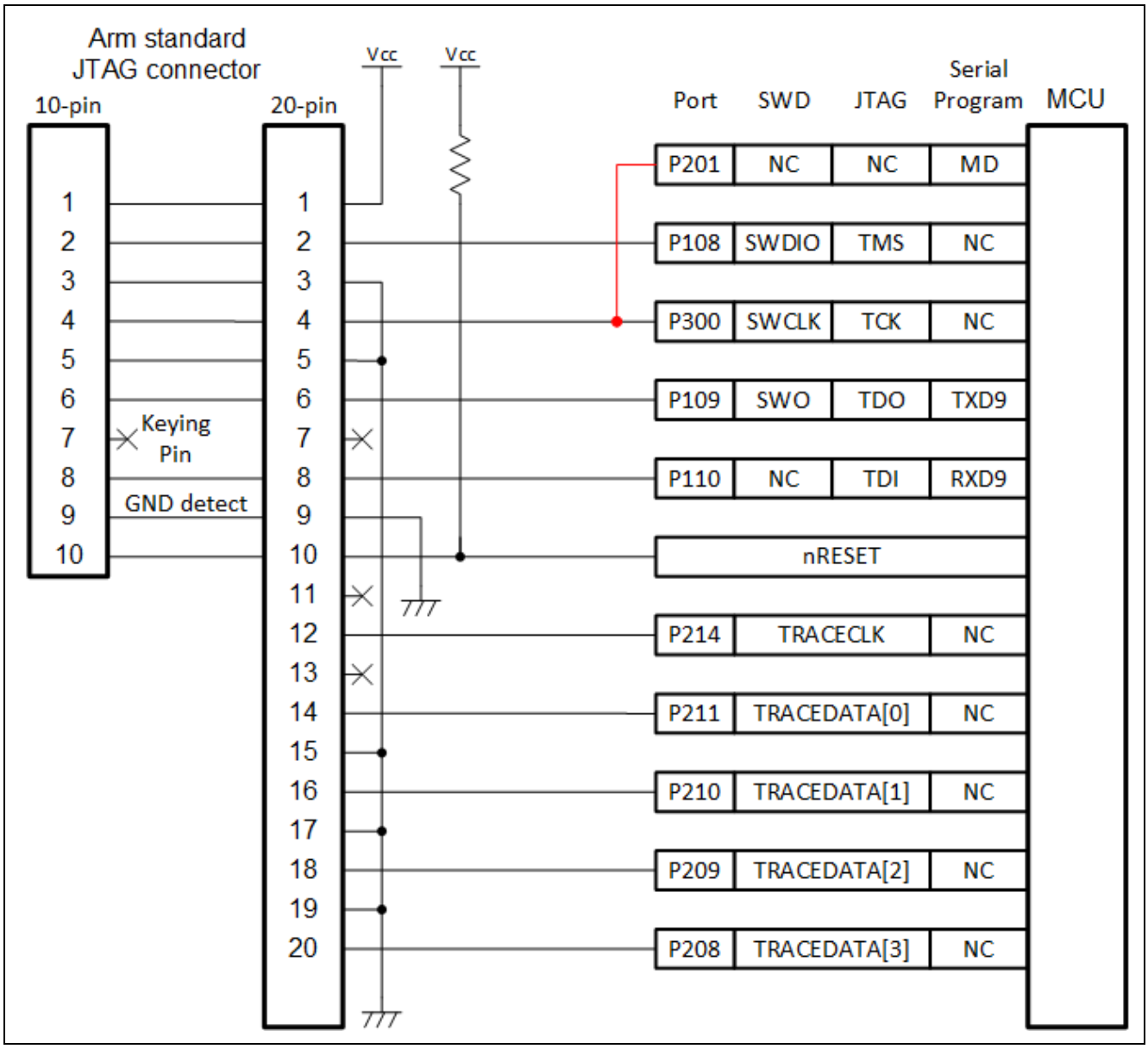

Figure 15. Emulator Connections for MCUs that Support Arm® Trustzone®

8. Memory

The RA6 MCUs support a 4-GB linear address space ranging from 0000 0000h to FFFF FFFFh that can contain program, data, and external memory bus. Some members of the family include an SDRAM controller that allows access to an SDRAM device connected to external memory bus. Program and data memory share the address space; separate buses are used to access each, increasing performance and allowing same-cycle access of program and data. Contained within the memory map are regions for on-chip RAM, peripheral I/O registers, program ROM, data flash, and external memory.

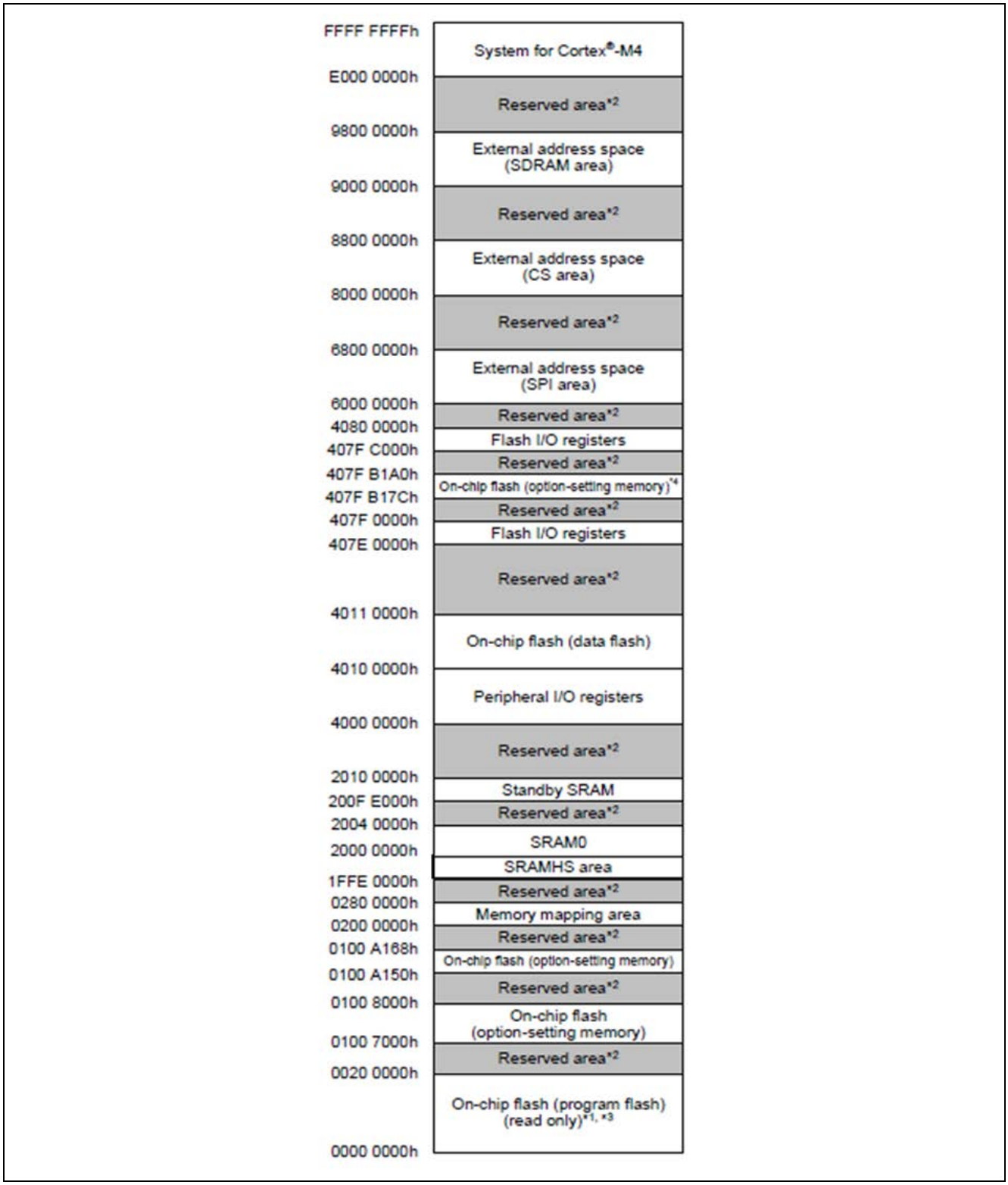

Figure 16. RA6M3 Memory Map

8.1 SRAM

The RA6 MCUs provide on-chip high-speed SRAM modules with either parity-bit checking or ECC (Error Correction Code). The area of the first 32 KB of SRAM0 is subject to ECC. Parity check is performed on other areas. The following table lists the SRAM specifications. The number of SRAM modules and capacity vary by device. Consult the Hardware User's Manual for specifics.

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*.

Figure 17. RA6M3 SRAM Specification

8.2 Standby SRAM

The RA6 MCUs provide an on-chip SRAM to retain data in Deep Software Standby mode. The following table lists the Standby SRAM specifications.

The power supply to the Standby SRAM in Deep Software Standby mode is enabled by the DPSBYCR.DEEPCUT[1:0] bits. If the DPSBYCR.DEEPCUT[1:0] bits are set to 00b, data in the Standby SRAM is retained in Deep Software Standby mode. See section 11, Low Power Modes, for details on the DPSBYCR.DEEPCUT[1:0] bits.

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

Figure 18. RA6M3 Standby SRAM Specification

The LPM (Low Power Mode) driver in Renesas FSP provides an option to cut or keep power to Standby SRAM as shown in the following figure. The LPM driver's APIs still needs to be invoked to write the selected settings to the MCU registers.

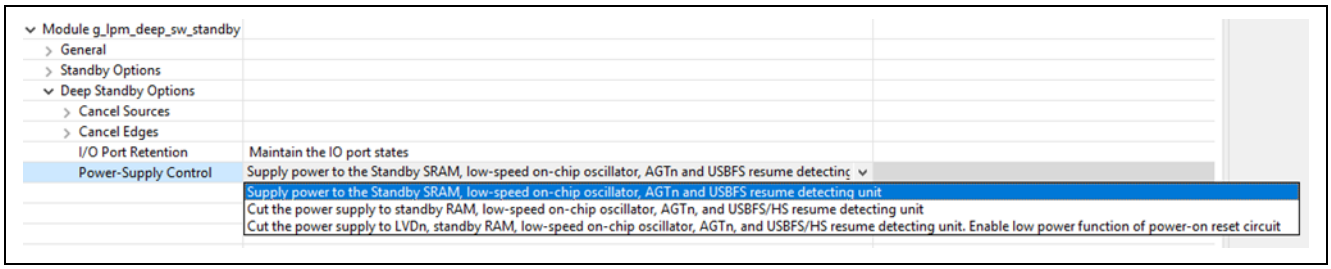

Figure 19. Enable/Disable Power Supply to Standby SRAM Using Renesas FSP Configurator

8.3 Peripheral I/O Registers

Blocks of peripheral I/O registers appear at various locations in the memory map depending on the device and the current operating mode. The majority of peripheral I/O registers occupy a region from address 4000 0000h to 400F FFFFh. However, this may vary in location and size by device. See the Hardware User's Manual for specifics. Details can be found in the *I/O Registers* appendix, and also in the register descriptions for each peripheral function. This region contains registers that are available at all times in all modes of operation. Flash I/O registers to control access flash memory occupy two regions, 407E 0000h to 407E FFFFh and 407F C000h to 407F FFFFh.

The Renesas FSP contains C header files in CMSIS data structure that map all of the peripheral I/O registers for a specific device to easily accessible I/O data structures.

8.4 On-Chip Flash Memory

The RA6 MCUs feature two flash memory sections: code flash and data flash, which vary in size and programmable cycle capacity by device. The Flash Control Unit (FCU) controls programming and erasure of the flash memory. The Flash Application Command Interface (FACI) controls the FCU in accordance with the specified FACI commands.

The code flash is designed to store user application code and constant data. The data flash is designed to store information that may be updated from time to time such as configuration parameters, user settings, or logged data. The units of programming and erasure in the data flash area are much smaller than that of the code flash (2 bytes for data flash versus 128 bytes for code flash).

Both the data flash and code flash areas can be programmed or erased by application code. This enables field firmware updates without having to connect an external programming tool.

Renesas FSP provides HAL layer drivers for both code flash memory and data flash memory.

The following figure shows example specifications of code flash memory and data flash memory.

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	

Figure 20. Specifications of Code Flash Memory and Data Flash Memory on RA6M3 MCU

Note: Erase state of code flash is FFh but erase state of data flash is undefined.

8.4.1 Background Operation

RA6 MCUs support background operations for code flash and data flash. This means that when a program or erase is started, the user can keep executing and accessing memory from memory areas other than the one being operated on. For example, the CPU can execute application code from code flash while the data flash memory is being erased or programmed. Also, the CPU can execute application code from SRAM while the code flash memory is being erased or programmed. The only exception to this rule is that the data flash cannot be accessed during code flash programming or erasing.

8.4.2 ID Code Protection

RA6 MCUs with Cortex-M4 core have a 128-bit memory in option setting memory area that is used as an ID code. If this ID code is left blank (0xFF's) then no protection is enabled and access to the MCU is allowed through boot mode or using the on-chip debugger. If the ID code is set then the user can control access to these modes. The user can choose to always disallow connections or can choose to allow connections when a matching ID code is input. Refer to the *OCD/Serial Programmer ID Setting Register (OSIS)* and *ID Code Protection* and sections of RA6 MCU Hardware User's Manual for more information.

Renesas FSP configurator provides options to set up ID code protection.

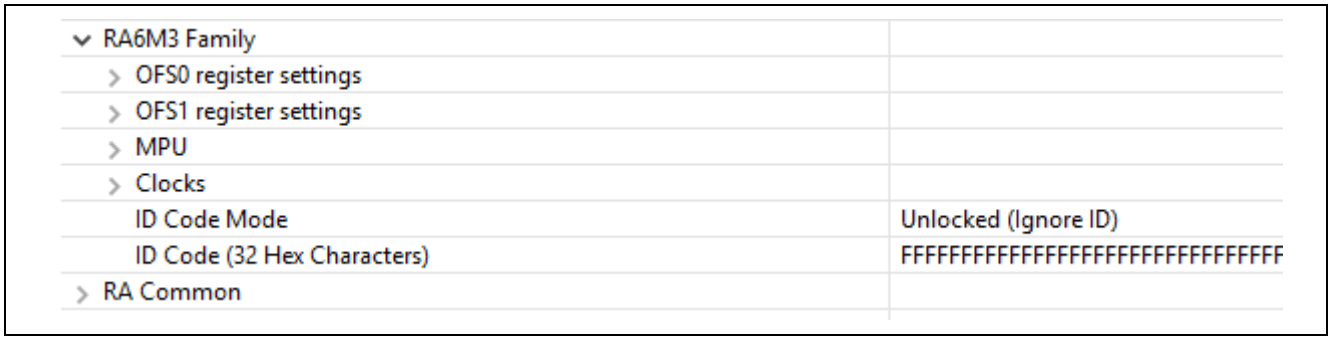

Figure 21. ID Code Setup Using Renesas FSP Configurator

Note: ID code protection settings must be handled carefully to prevent mistakes that may result in blocking accesses to the MCU.

8.4.3 Device Lifecycle Management

RA6 MCUs with Cortex-M33 core are equipped with Device Lifecycle Management (DLM), which is the management of the process by which a product goes from inception to development to production and then eventually end-of-life. The RA Family MCU debug capability and serial programming capability are defined by the device lifecycle states.

Refer to the *Device Lifecycle Management Key Installation* application note for more details.

8.4.4 Flash Block Protection

RA6 MCUs with Cortex-M33 core have a Flash Block Protection feature that protects secure or non-secure flash region from being erased or reprogrammed by secure or non-secure software. It is worth noting that the protection is for both Secure and Non-secure software accesses.

Refer to section *Configuring the Flash Block Protection* in *Securing Data at Rest Using the Arm® TrustZone®* application note for more details.

8.4.5 Memory Protection Unit

RA6 MCUs with Cortex-M4 core have Memory Protection Units (MPUs). They have the ability to protect various MCU regions from illegal access. The choices include allowing both reading and writing, prohibiting writing, and prohibiting writing and reading. Select one of these options by setting the corresponding constant value at the specific memory address. See *Memory Protection Unit* in the MCU Hardware User's Manual for more details.

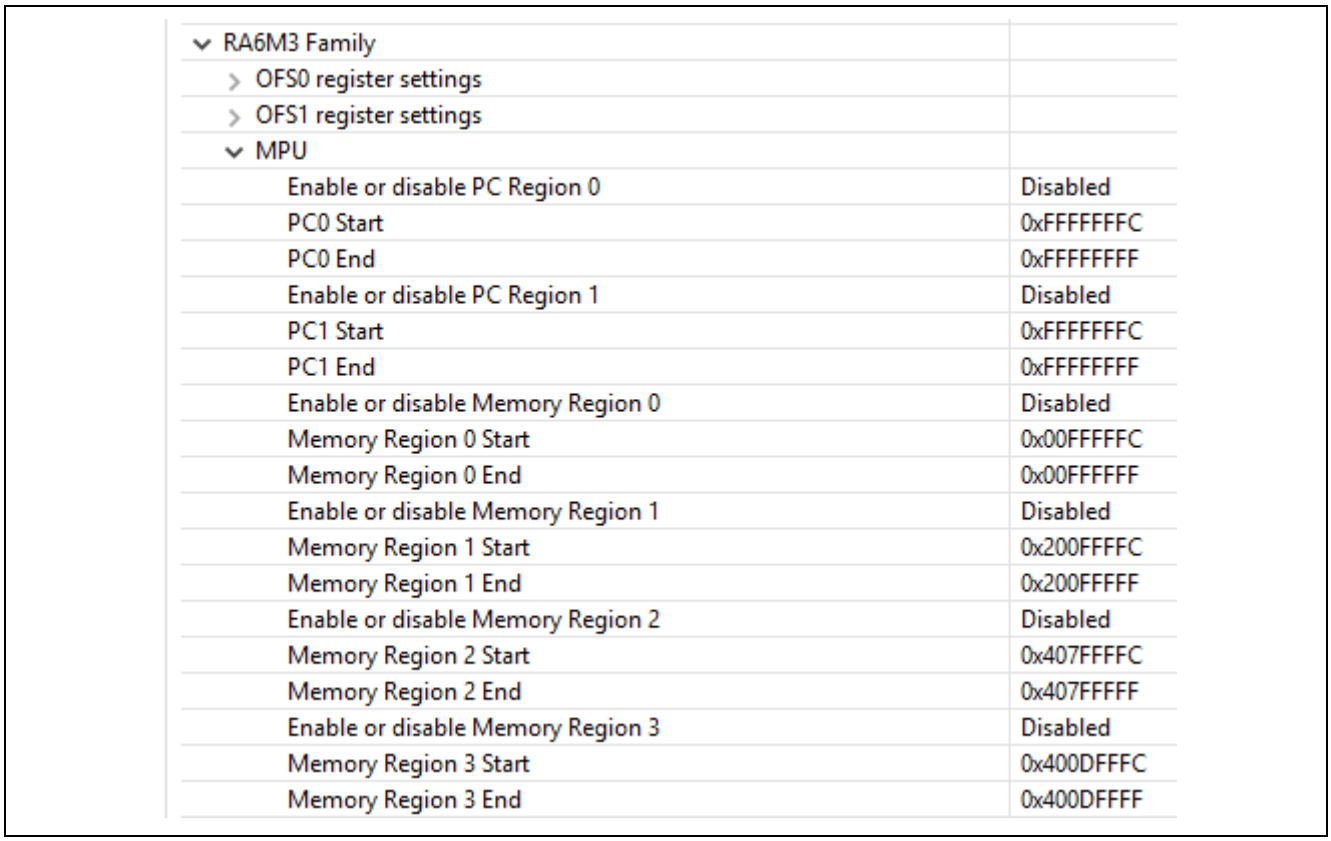

Figure 22. MPU Setup Using Renesas FSP Configurator

Note: MPU settings must be handled carefully to prevent mistakes that may result in blocking accesses to an MCU region.

8.5 External Memory

The RA6 MCUs include an external data bus for connection to external memory and devices. Some members also include a built-in SDRAM controller that allows the use of up to 128 Mbytes of external SDRAM. Eight programmable chip selects provide a number of options that are settable on a per-chip select basis to allow connection to a wide range of external devices. The external chip select area of the memory map begins at address 0x60000000. Consult the Hardware User's Manual for more details.

8.5.1 Using External 16-bit Memory Devices

When connecting an external 16-bit memory device that has a byte select line, connect A1 of the MCU to A0 of the memory and A0 of the MCU to the byte select line.

8.5.2 Example of SDRAM Initialization

The Renesas FSP provides C header files in the CMSIS data structure that map all of the external bus control registers. The below function is an example to initialize SDRAM memory controller using CMSIS register structures in Renesas FSP.

```
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 Data Alignment

There are no limits for aligning data. The MCU is capable of performing byte, word, and long accesses on odd memory locations. While it is still optimal to align data accesses, it is not required.

8.7 Restriction on Endian

Memory space must be little-endian to execute code on the Arm® Cortex®-M core.

9. Register Write Protection

The register write protection function protects important registers from being overwritten because of software errors. The registers to be protected are set with the Protect Register (PRCR). Table 14 lists the association between the PRCR bits and the registers to be protected.

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

Figure 23. PRCR Register

Table 14. PRCR Protection Bits

PRCR bit	Description
PRC0	• Registers related to the Clock Generation Circuit: 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	• Registers related to the low power modes: SBYCR, SNZCR, SNZEDCR, SNZREQCR, OPCCR, SOPCCR, DPSBYCR, DPSIER0-3, DPSIFR0-3, DPSIEGR0-2, SYOCDCR, STCONR • Registers related to the battery backup function: VBTBKRn (n = 0 to 511), VBTICTLR
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVCMPCR, LVDLVLR, LVD1CR0, LVD2CR0
PRKEY[7:0]	These bits control write access to the PRCR register. To modify the PRCR register, write A5h to the eight higher-order bits and the wanted value to the eight lower-order bits as a 16-bit unit.

Renesas FSP supplies two APIs (R_BSP_RegisterProtectEnable and R_BSP_RegisterProtectDisable) to enable and disable Register Write Protection respectively.

10. I/O Port Configuration

The *I/O Ports* section of the Hardware User's Manual describes exact pin configurations based on peripheral selection and other register settings. Some general information is listed as follows.

It is important to note that after a reset, each pin will be in the default state for that pin until the configuration is applied. There may be a small period where some pins may be in an undesirable state. This will be true regardless of what configuration approach is used. The user should consider the impact this may have for each application, including how this may affect other system features.

10.1 Multifunction Pin Selection Design Strategies

Most ports on the RA6 Series of MCUs are capable of multiple peripheral functions. Tools, such as the pin configurator in FSP, are available from Renesas to assist with port selection for each RA6 device. When several peripheral functions are needed, use the following design strategies to help with port function selection.

- Assign peripheral functions with only one port option first. For example, there is only one port option for each Trace Data signal in the debug function. When this function is needed, assign these ports first.
- Assign peripheral functions with limited port options next. For example, devices that support the QSPI peripheral typically only have two options for each QSPI signal.
- Assign peripheral functions with multiple port options last. One example would be the Serial Communications Interface (SCI) which typically has many available port options.
- Some peripheral function port options are interchangeable, while others must be assigned in logical groups. For example, the IIC peripheral has some ports with the suffix "_A" while others have the suffix "_B" in the signal name. Ports should be selected to have the same suffix for the peripheral function. Other peripheral functions do not have this type of suffix, and ports may be assigned interchangeably, such as the USB_VBUSEN signal for the USBFS peripheral function. Also see section 15.3 in this document.

10.2 Setting Up and Using a Port as GPIO

There are two methods for setting up and using a port as GPIO, either using the Port Control Register (PCNTR1), or the PmnPFS registers.

Method 1: Port Control Register (PCNTR1)

- Select a pin as an output by writing a “1” to the Port Direction bit (PDRn) in Port Control Register 1 (PCNTR1).
- The Port Direction bits (PDRn) are read/write. Setting the value to a “1” selects the pin as an output. Default state for I/O Ports is “0” (input). The port direction registers can be read on the RA6 MCUs.
- The Port Output Data bits (PODRn) in the corresponding Port Control Register (PCNTR1) are read/write. When the PODR is read the state of the output data latch (not the pin level) is read.
- The Port Input bits (PIDRn) in Port Control Register 2 (PCNTR2) are read only. Read the PIDRn bit in the PCNTR2 register to read the pin state.

Method 2: Port mn Pin Function Select (PmnPFS) registers

- The Port Mode Register (PMR) is read/write and is used to specify whether individual pins function as GPIO or as peripheral pins. Out of reset all PMR registers are set to 0 which sets all pins to work as GPIO. If a PMR register is set to 1 then that corresponding pin will be used for peripheral functions. The peripheral function is defined by that pin’s MPC setting.
- When setting a pin as an output it is recommended that the desired output value of the port be written to the data latch first, then the direction register is set to an output. Though not important in all systems, this prevents an unintended output glitch on the port being setup.

In general, using PCNTR1 to configure a port will provide faster access, but will have fewer configuration features available. Using the PmnPFS registers will have more configuration features available but will have slower access.

Renesas FSP provides Pin Configuration to configure GPIO pin after reset as shown below. After the GPIO is configured, it can be controlled using HAL layer APIs in FSP.

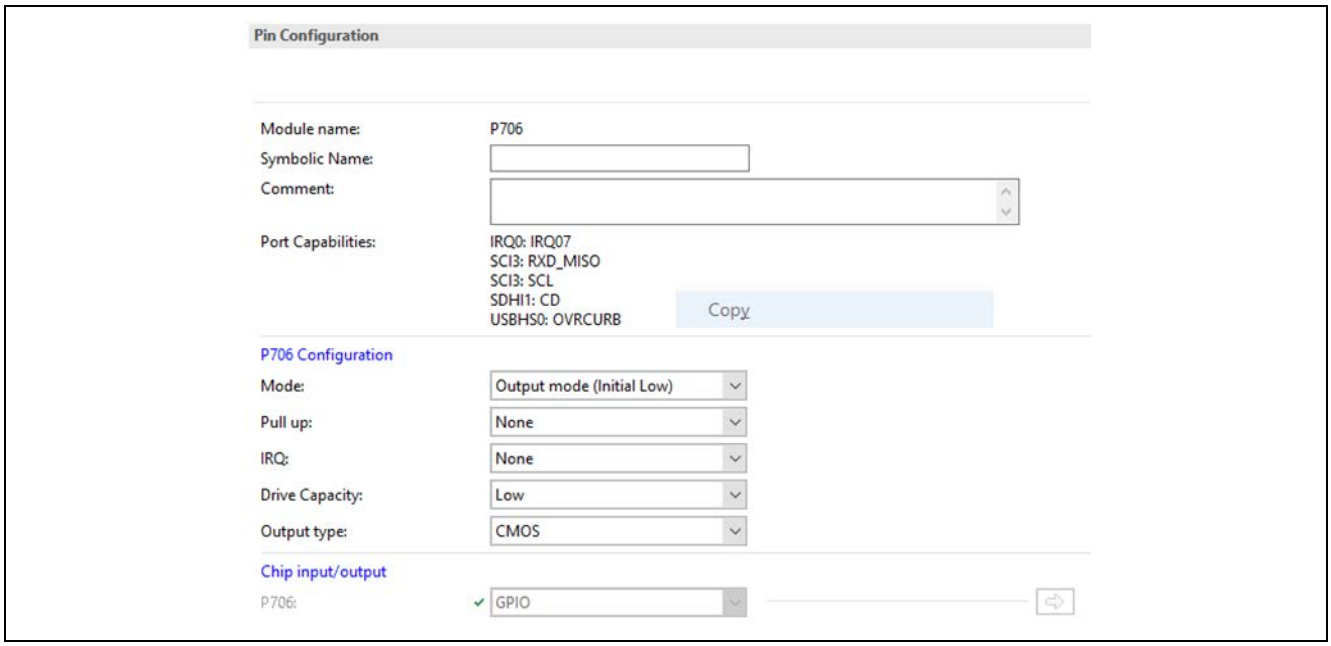

Figure 24. Configuring P706 as Output and Low using FSP Configurator

10.2.1 Internal Pull-Ups

- Most pins on ports 0 through 9, A and B have the option of enabling a pull-up resistor. The pull-up is controlled by the Pull-Up bit (PCR) bit in each Port mn Pin Function Select (PmnPFS) Register. The PCR bit in each PmnPFS register controls the corresponding pin on the port.
- The pin must first be set as an input with the associated bit in the PmnPFS register. Set the PCR bit to “1” to enable the pull-up and to “0” to disable it.
- Out of reset all PCR registers are cleared to 0, therefore all pull-up resistors are disabled.
- The pull-up is automatically turned off whenever a pin is designated as an external bus pin, a GPIO output, or a peripheral function output pin.

10.2.2 Open-Drain Output

- Pins configured as outputs normally operate as CMOS outputs.
- Most pins on ports 0 through 9, A and B have the option of being configured as an NMOS open-drain output.
- The N-channel open-drain control (NCODR) bit in each Port mn Pin Function Select (PmnPFS) Register controls which pins operate in open-drain mode. Setting the applicable bit in each register to a “1” makes the output open-drain. Setting the applicable bit in each register to a “0” sets the port to CMOS output.

10.2.3 Drive Capacity

- Each pin on ports 1 through 9, A and B (except P200 and P201) has the option of enabling low-, middle-, or high-drive output. The drive capacity switching is controlled by the Drive Capacity Control Register (DSCR) bits in each Port mn Pin Function Select (PmnPFS) register.
- Out of reset all DSCR registers are cleared to 0 therefore all pins are set to low drive output. Setting a value other than “00” will change the drive capacity of the output for the selected pin.
- The maximum total output of all pins summed together is 80 mA.
- The differences the drive levels are shown below:

Typical output pins	**DSCR[1:0]**	**Drive Capacity**	**Average (mA)**	**Max (mA)**
Permissible output current per pin	0 0	Low Drive	2.0	4.0
Permissible output current per pin	0 1	Middle Drive	4.0	8.0
Invalid setting; do not use	1 0	-		
Permissible output current per pin	1 1	High Drive	16	32

High drive output pins	**DSCR[1:0]**	**Drive Capacity**	**Average (mA)**	**Max (mA)**
Permissible output current per pin	0 0	Low Drive	2.0	4.0
Permissible output current per pin	0 1	Middle Drive	4.0	8.0
Invalid setting; do not use	1 0	-		
Permissible output current per pin	1 1	High Drive	20	40

Output drive capacity can have a significant impact on overall performance of a board design. The following points should be considered when selecting the drive capacity for each output.

- It is recommended to start with all pins set to low-drive capacity (default) and evaluate the performance.
- Depending on the board layout, pins set to middle- or high-drive capacity may result in higher EMI radiation.
- Long traces may require higher drive capacity for signals to propagate correctly to the receiver.

10.3 Setting Up and Using Port Peripheral Functions

The Port mn Pin Function Select Registers (PmnPFS) are used to configure the characteristics of each port. The PSEL bits select the peripheral function selected for each port.

- Since most pins have multiple functions the RA6 MCUs have Pin Function Control Registers (PmnPFS) that allow you to change the function assigned to a pin.
- Each pin has its own PmnPFS register.
- Each PmnPFS register allows a pin to be used for peripheral function (PSEL bits), as an IRQ input pin (ISEL bit), or as an analog input pin (ASEL bit). If the ASEL bit is set to “1” (use pin as analog input pin) then the pin’s PMR bit should be set for GPIO use and the pin’s PDR bit should be set for input.
- Refer to the “Peripheral Select Settings for each Product” section in the “I/O Ports” chapter of the Hardware User’s Manual.
- In order to ensure that no unexpected edges are input or output on peripheral pins, make sure to clear the Port Mode Control (PMR) bit for the targeted pin before modifying the pin’s PmnPFS register.
- All PmnPFS registers are write protected out of reset. In order to write to these registers, the Write-Protect Register (PWPR) must first be used to enable writing.
- Care should be taken when setting PmnPFS registers such that a single function is not assigned to multiple pins. The user should not do this but the MCU will allow it. If this occurs the function on the pins will be undefined.
- If you are using the external bus, the Ethernet controller, or USB, there are additional registers in the MPC that must be configured before using these peripherals.
- The following figure shows an example of enabling QSPI pins using FSP Pin configuration.

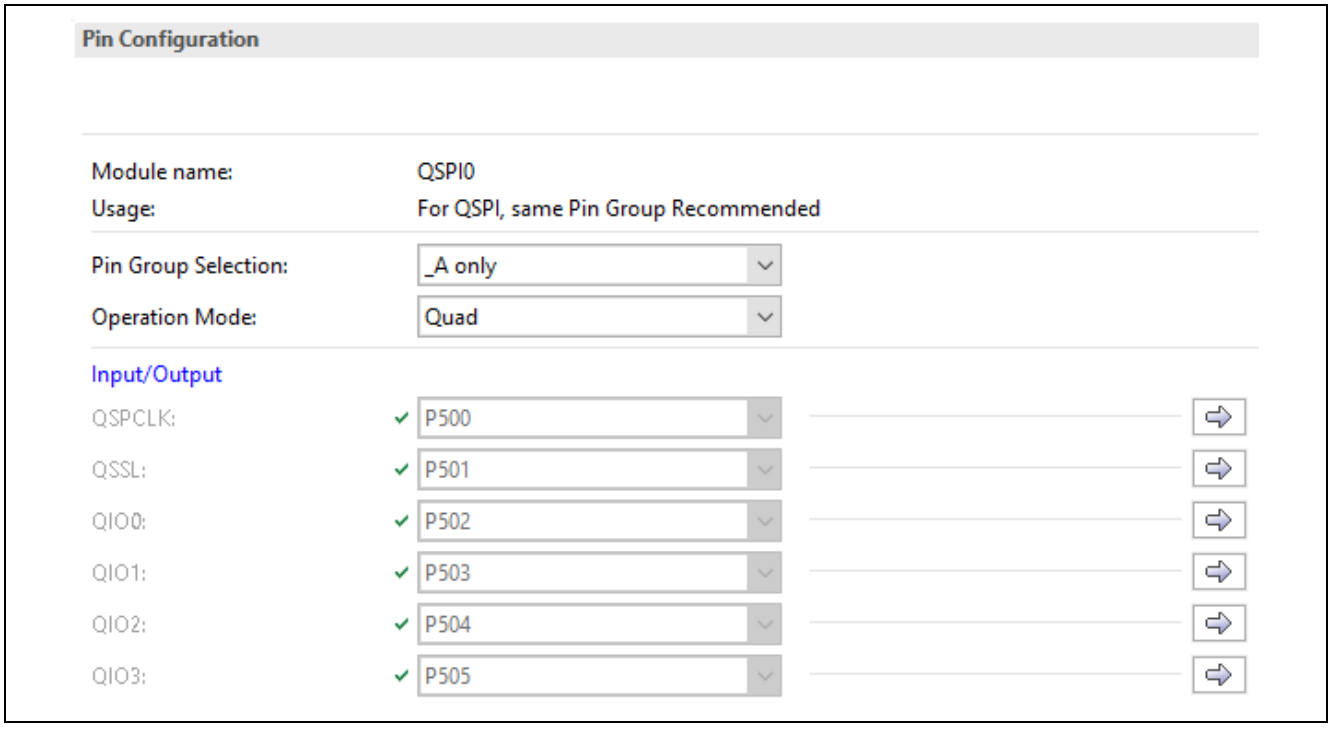

Figure 25. Enabling QSPI pins using Pin Configurator in Renesas FSP

10.4 Setting Up and Using IRQ Pins

- Certain port pins can be used as hardware interrupt lines (IRQ). See the *Peripheral Select Settings for each Product* section in the *I/O Ports* chapter of the Hardware User's Manual for information on which pins are available for your MCU.
- Some IRQ pins have a "-DS" suffix (for example, IRQ1-DS). The "-DS" designates that this pin can be used to wake the MCU out of deep software standby mode.
- It is not possible to use IRQn and IRQn-DS at the same time. Same number interrupts with the -DS and without the -DS suffix connect to the same interrupt internally, even though they use different external pin connections.
- To set a port pin to be used as an IRQ pin, the Interrupt Input Function Select bit (ISEL) in the pin's PFS register must be set to "1".
- Pins can be used for both IRQ and peripheral functions simultaneously. To enable this the user should set both the ISEL and PSEL bits in the pin's PFS register.
- IRQ functions of the same number must only be enabled on one pin.
- IRQ pins can trigger interrupts on detection of:
  - Low level
  - Falling edge
  - Rising edge
  - Rising and falling edges

  Which trigger is selected is chosen using the IRQ Control Registers (IRQCRi).
- Digital filtering is available for IRQ pins. The filters are based on repetitive sampling of the signal at one of four selectable clock rates (PCLK, PCLK/8, PCLK/32, PCLK/64). They filter out short pulses: any high or low pulse less than 3 samples at the filter rate. The filters are useful for filtering out ringing and noise in these lines but are much too quick for filtering out long events like mechanical switch bounce. Enabling filtering adds a short bit of latency (the filter time) to the hardware IRQ lines.
- Digital filtering can be enabled for each IRQ pin independently. This is done by setting the IRQ Pin Digital Filter Enable (FLTEN) bit in the IRQCRi register for each IRQ.
- The clock rate for digital filtering is configurable for each IRQ pin independently. This is done by setting the IRQ Pin Digital Filter Setting (FCLKSEL[1:0]) bits in the IRQCRi register for each IRQ.
- Figure 26 and Figure 27 show examples of enabling and configuring IRQ pins using Renesas FSP.

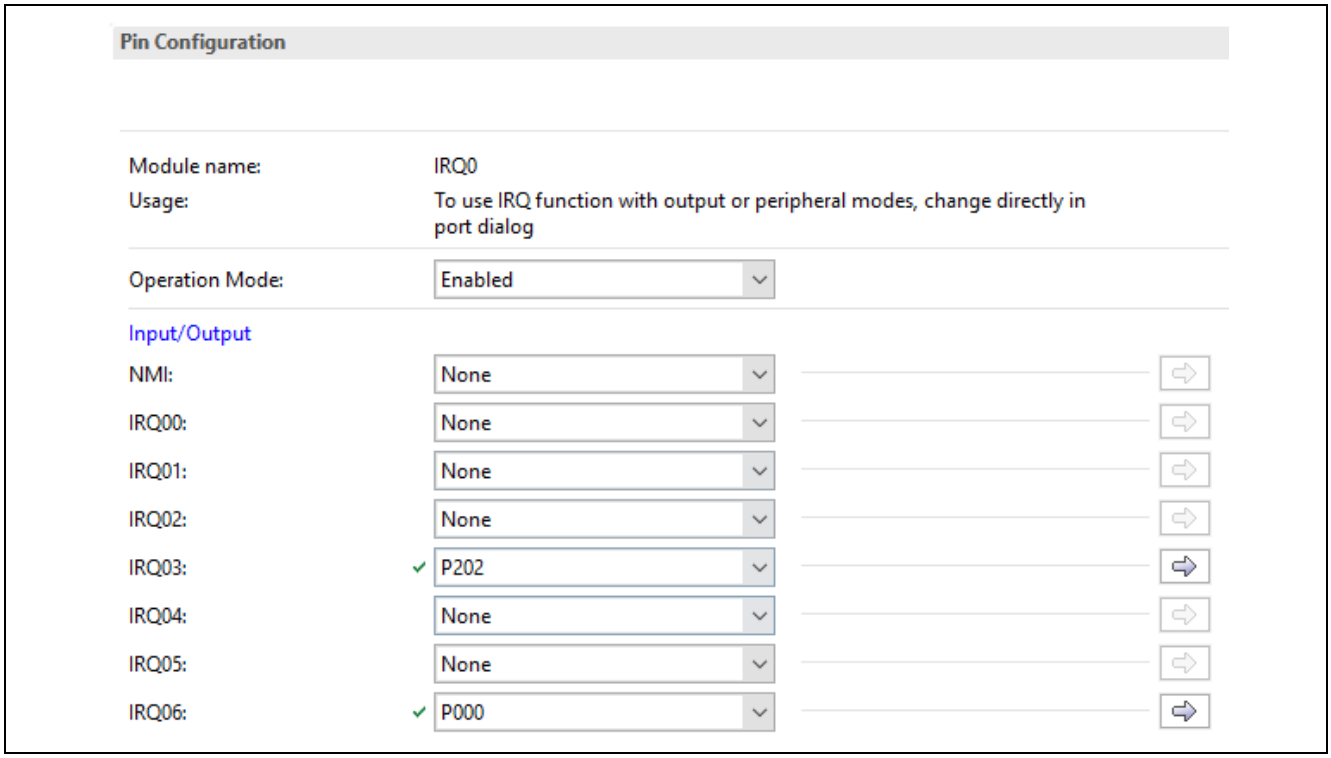

Figure 26. Enable P202, P000 as IRQ03, IRQ06 inputs Respectively using Pin Configurator in Renesas FSP

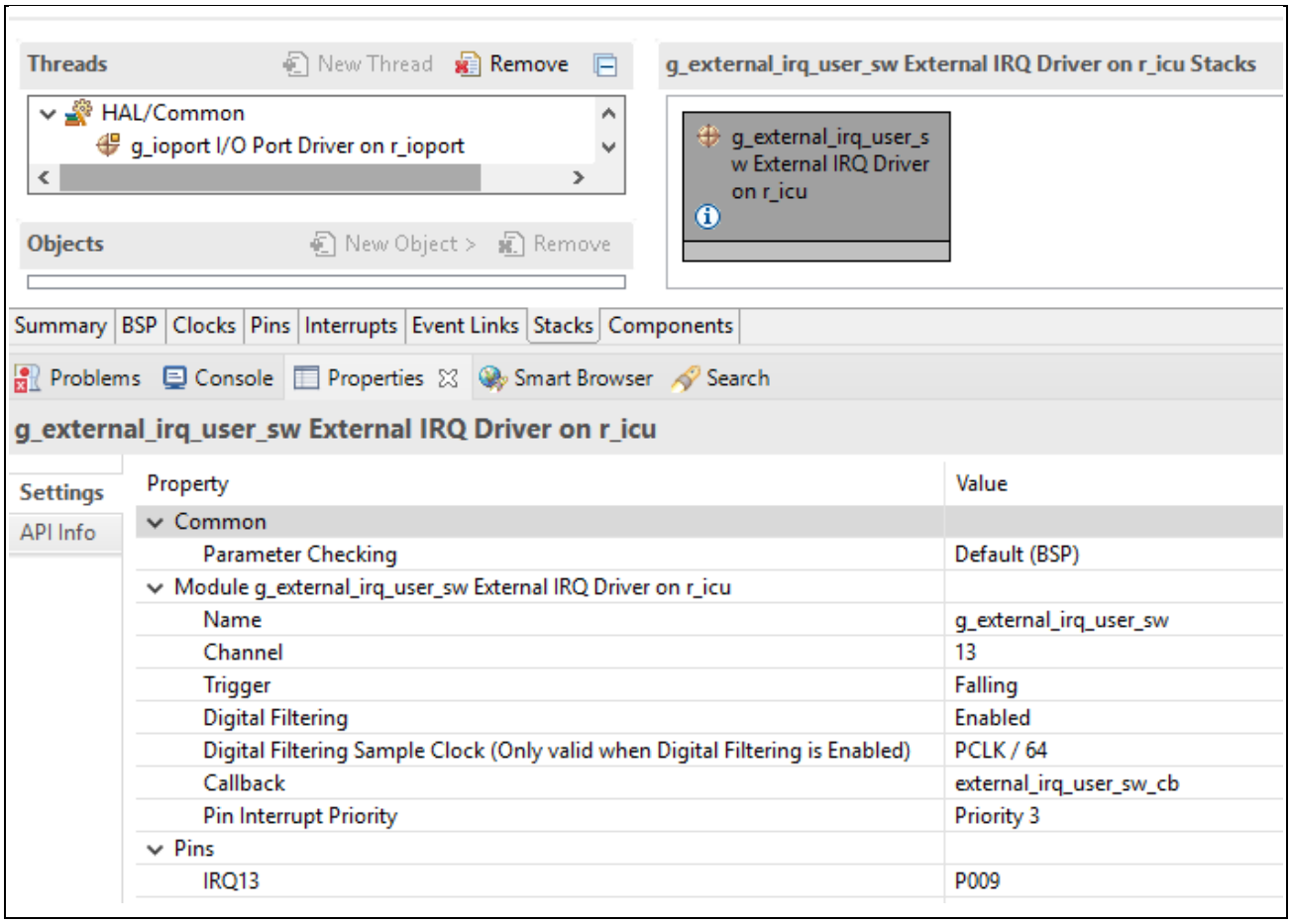

Figure 27. Configure IRQ13 using Renesas FSP Configurator

10.5 Unused Pins

Note: Some pins require specific termination: See the *Handling of Unused Pins* section of the Hardware User's Manual for specific recommendations.

Unused pins that are left floating can consume extra power and leave the system more susceptible to noise problems. Terminate unused pins with one of the methods detailed here:

1. The first option is to set the pin to an input (the default state after reset) and connect the pin to V_{cc} or V_{ss} using a resistor. There is no difference to the MCU between one connection or another; however, there may be an advantage from a system noise perspective. V_{ss} is probably the most typical choice. Avoid connecting a pin directly to V_{cc} or V_{ss} since an accidental write to the port's direction register that sets the pin to an output could create a shorted output.
2. A second method is to set the pin to an output. It does not matter whether the pin level is set high or low; however, setting the pin as an output and making the output low connects the pin internally to the ground plane. This may help with overall system noise concerns. A disadvantage of setting unused pins to outputs is that the configuration of the port must be done via software control. While the MCU is held in Reset and until the direction register is set for output, the pin will be a floating input and may draw extra current. If the extra current can be tolerated during this time, this method eliminates the external resistors required in the first method.
3. A variation on leaving the pins as inputs and terminating them with external resistors uses the internal pull-ups available on many ports of the MCU. This has the same limitation as setting the pins to outputs (requires the program to set up the port) but it does limit the effect of accidental pin shorts to ground, adjacent pins or V_{cc} since the device will not be driving the pin.

10.6 Nonexistent Pins

Each RA6 MCU group is available in multiple package sizes, with different total pin counts. For any package smaller than the largest package for that MCU group (typically 176 pins) set the corresponding bits of nonexistent ports in the PDR register to “1” (output) and in the PODR register to “0”. The user can see which ports are available on each MCU package by reviewing the *Specifications of I/O Ports* table in the *I/O Ports section* of the Hardware User’s Manual. For example, pins 0 and 1 on port 1 are only available on 176 pin packages. Note that no additional handling of nonexistent pins is required.

10.7 Electrical Characteristics

Normal GPIO ports typically require CMOS level inputs (High ≥ 0.8 * V_{cc} , Low ≤ 0.2 * V_{cc}). Some GPIO ports have Schmitt Trigger inputs, which have slightly different input requirements. See the Hardware User’s Manual section *Electrical Characteristics* for more information.

11. Module Stop Function

To maximize power efficiency, the RA6 Series of MCUs allow on-chip peripherals to be stopped individually by writing to the Module Stop Control Registers (MSTPCRi, i=A, B, C, D, E). Once a module stops, access to the module registers is not possible.

After a reset, most of the modules are placed in module-stop state, except for DMAC, DTC, and SRAM. See Hardware User’s Manual for details.

Before accessing any of the registers for a peripheral, it must be enabled by taking it out of stop mode by writing a ‘0’ to the corresponding bit in the MSTPCRi register.

Peripherals may be stopped by writing a ‘1’ to the proper bit in the MSTPCRi register.

HAL drivers in Renesas FSP handle module start/stop function automatically.

12. Interrupt Control Unit

The Interrupt Controller Unit (ICU) controls which event signals are linked to the NVIC, DTC, and DMAC modules. The ICU also controls non-maskable interrupts. Figure 28 shows an example of the ICU specifications, and Figure 29 shows an example of the ability to raise the IRQi event from the I/O pins. Refer to the Hardware User’s Manual for details for each RA6 MCU Group.

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.

Figure 28. RA6M3 ICU Specification

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

Figure 29. RA6M3 ICU I/O Pins

The following figure is an example of using Renesas FSP configurator to enable and configure an interrupt using Renesas FSP. The ICU and interrupts are configured as part of the HAL driver configuration through FSP.

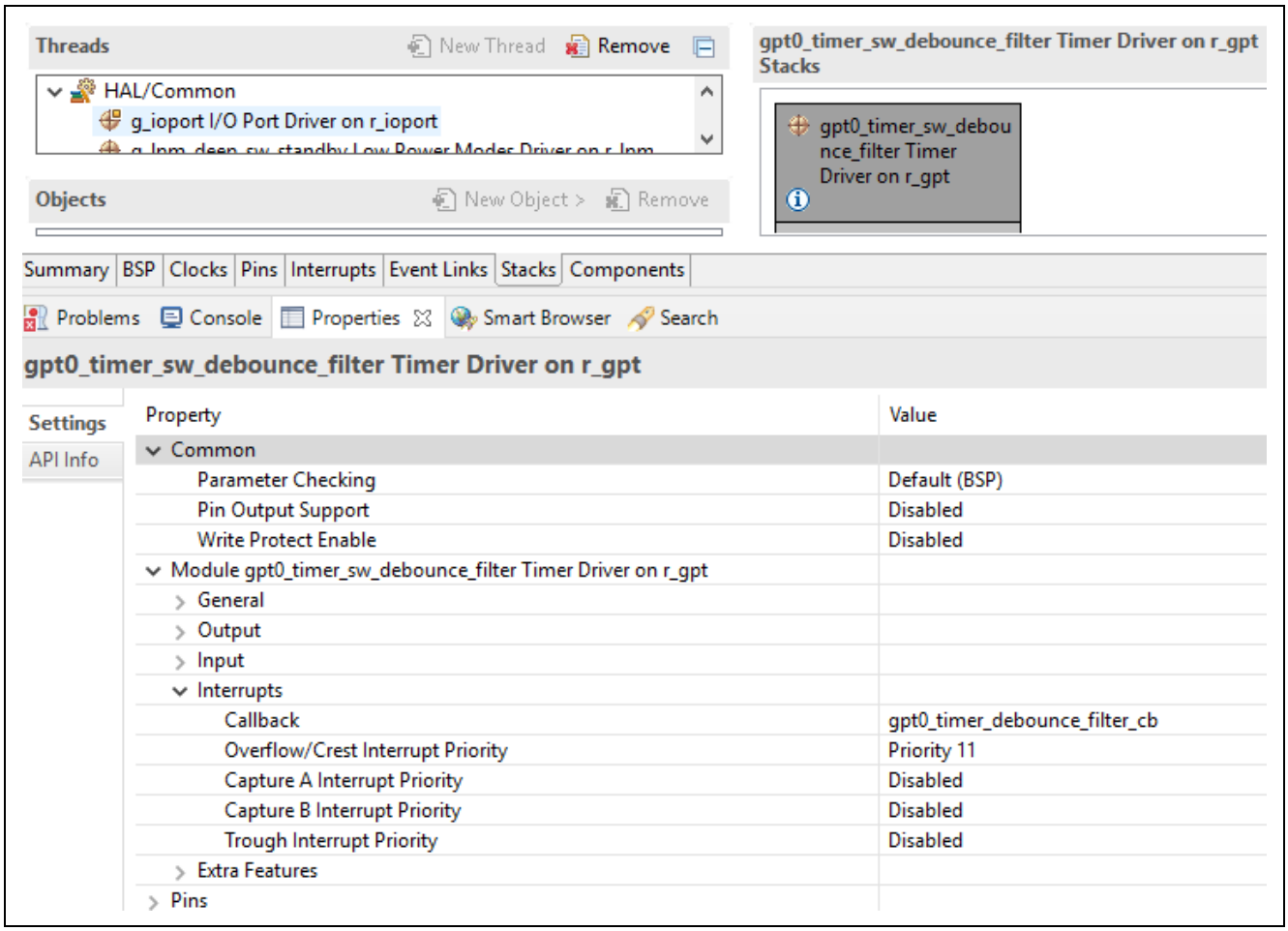

Figure 30. Enable GTP0 Overflow Interrupt and Set User Callback Functions Invoked by Interrupt Service Routine

13. Low Power Consumption

The RA6 devices have several functions for reducing power consumption. These include setting clock dividers, EBCLK output control, stopping modules, selecting power control mode in Normal mode, and transitions to low power modes. Refer to the *Low Power Modes* chapter in the Hardware User's Manual for more details.

RA6 MCUs support four different types of LPM depending on the MCU Group. These are:

- Sleep mode
- Software Standby mode
- Snooze mode
- Deep Software Standby mode.

The following table is an overview of the functions available for reducing power consumption.

Table 15. Specifications of the Lower Power Mode Functions

Item	Specification
Reducing power consumption by modifying clock signals	The frequency division ratio can be selected independently for the system clock (ICLK), peripheral module clock (PCLKH, PCLKL, PCLKADC, PCLKGPT), external bus clock (BCLK), and flash interface clock (FCLK).*1
EBCLK output control	BCLK output or high-level output can be selected.
Module stop	Functions can be stopped independently for each peripheral module.
Low-power modes	• Sleep mode • Software Standby mode • Snooze mode • Deep Software Standby mode
Power control modes	Three operating power control modes: • High-speed mode • Low-speed mode • Subosc-speed mode
TrustZone Filter*2	Security attribution can be set for each register

Notes: 1. For details, see the *Clock Generation Circuit* chapter in the Hardware User's Manual.
2. For devices that support Arm® TrustZone® security features.

The following table lists the conditions to transition to low power modes, the states of the CPU and the peripheral modules, and the method for cancelling each mode.

Table 16. Low Power Consumption Modes

State of operation*1	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition condition	WFI instruction while SBYCR.SSBY=0	WFI instruction while SBYCR.SSBY=1 and DPSBYCR.DPSBY=0	Snooze request trigger in Software Standby mode. SNZCR.SNZE=1	WFI instruction while SBYCR.SSBY=1 and DPSBYCR.DPSBY=1
Canceling method	All interrupts. Any reset available in the mode.	Interrupts defined for this mode. Any reset available in the mode.	Interrupts defined for this mode. Any reset available in the mode.	Interrupts defined for this mode. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Reset state
State after cancellation by a reset	Reset state	Reset state	Reset state	Reset state

Notes: 1. Refer to the *Operating conditions of each low power mode* table in the Hardware User's Manual for additional details.

RA6 devices include register settings that allow the MCU to operate with lower power consumption in Normal mode and Sleep mode. These modes are referred to as the Operating Power Control Modes and are controlled by the OPCCR register.

The following is a summary of the Operating Power Consumption Control modes and the maximum permissible clocking and voltage levels under each mode.

Table 17. Available Oscillators in each Operating Power Consumption Control Mode

Mode	Oscillator						
	PLL, PLL2	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator	IWDT-dedicated on-chip oscillator
High-speed	Available	Available	Available	Available	Available	Available	Available
Low-speed	N/A	Available	Available	Available	Available	Available	Available
Subosc-speed	N/A	N/A	N/A	Available	N/A	Available	Available

Note: While it may be possible to set the value in the OPCCR register to any of the low power operating modes, clocking and voltage levels must also be set to meet the requirements of the desired mode. Otherwise, the settings in the OPCCR register will not have any effect in lowering power consumption.

In order to achieve the lowest power numbers, use the maximum possible dividers in the clock generation circuits.

Low power modes are canceled by various interrupt sources such as RES pin reset, power-on reset, voltage monitor reset, and peripheral interrupts. Refer to the *Low Power Modes* section in the Hardware User's Manual for a list of interrupt sources for different LPMs.

Only Snooze mode is triggered by a Snooze request to enter Snooze mode from Software Standby mode. The transitions to other LPMs are done by executing a WFI instruction with appropriate settings in the Standby Control register (SBYCR).

Renesas FSP provides a low power mode (LPM) driver and driver configurator to set up low power mode, wake source/cancel source, and so forth.

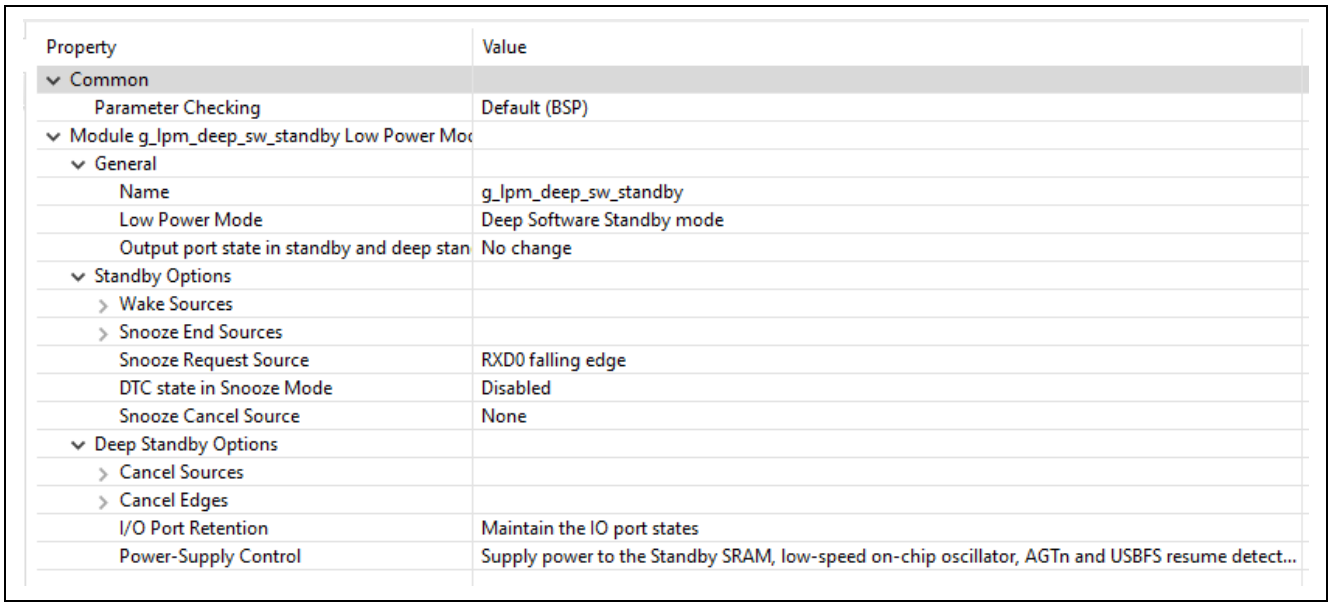

Figure 31. Set up Low Power Mode Using Renesas FSP Configurator

After a specific LPM mode is set up by FSP Configurator, LPM driver's API can be used to initialize LPM driver and place MCU in configured LPM mode, as shown in the following example:

```
/* 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. External Buses

RA6 devices include an external bus controller. Some RA6 devices have built-in SDRAM controllers.

14.1 Bus Width and Multiplexing

The access width of external memory areas can be set to 8-bit or16-bit. Width settings are set on a per-chip-select basis by setting the BSIZE bits in the CSnCR register or the SDC Control Register (SDCCR). The address and data lines of chip-select regions can be multiplexed by setting the MPXEN bit in the CSnCR register.

14.2 Drive Strength for Bus Signals

When an external memory area is used, pins that control the bus signals should be set for high-drive capacity output in high-speed setting. See the section *Port mn Pin Function Select Register* in the *I/O Ports* chapter, and the *Electrical Characteristics* chapter in MCU Hardware User's Manual for more information on setting the drive capacity of a pin.

14.3 Bus Errors

The following types of errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- TrustZone® Filter error (MCUs that support Trustzone only).
- Bus error transmitted from each slave IP

When a bus error occurs, operation is not guaranteed, and the error is returned to the requesting master IP. The bus errors that occur for each master are stored in the BUSnERRADD and BUSnERRSTAT registers. These registers must only be cleared by a reset. For more information, see section *Bus Error Address Register (BUSnERRADD)* and *Bus Error Status Register (BUSnERRSTAT)* in the Hardware User's Manual.

Note: The DMAC and DTC do not receive bus errors, so their operation is not affected by bus errors.

15. General Layout Practices

15.1 Digital Domain vs. Analog Domain

Renesas RA6 Microcontroller devices have three primary types of pin functions: Power, Digital, and Analog.

Generally, power pins are dedicated for voltage and reference input and do not have multiple functions. Power pins are typically dedicated to specific portions, or domains, within the MCU. For example, the main supply voltage for the MCU will provide power to the digital core, many of the digital peripheral functions and many of the digital I/O pins. The digital domain can be defined as the digital circuitry, digital I/O pins, and the related power pins. Power pins which are designated for analog functions (such as AVCC0 and the associated AVSS0) supply specific analog circuitry within the MCU, which is separate from the digital domain circuitry. The analog domain can be defined as the analog circuitry, analog I/O pins, and the related power pins.

Digital signals are typically repetitive, switched patterns that are associated with periodic clocks. The transitions on digital signals tend to be relatively sharp edges, with stable levels of high or low between the transitions. Each signal must be stable at an acceptable voltage level, referred to as a logic state, within a specified timeframe. The state of the signal is typically sampled at predetermined clock intervals, using the edge transition of a clock to evaluate the associated data signals. Small variations in the voltage level of digital signals are typically acceptable, as long as the level remains within a specified range. However, large external influences on digital signals can have an acute influence on a digital signal, which can result in an incorrect logic state at the moment when the data is sampled.

Analog signals are usually quite different. Analog signals may be periodic, but the evaluation of an analog signal is typically a measurement of voltage over a range instead of logic state. The voltage level of an analog signal is sampled based on a specific trigger event, and the resulting measurement is processed using the analog circuitry in the MCU. The accuracy of an analog measurement is directly related to the accuracy of the sampled voltage level. Any unwanted external influence which may change the voltage level of an analog input signal, even slightly, can influence the accuracy of the measurement.

Due to the highly multiplexed nature of the I/O pins on Renesas RA6 MCU devices, many I/O pins can be used for either Analog or Digital functions. This can result in situations where digital and analog functions may overlap and result in data errors.

To minimize potential problems between digital and analog signal domains, consider the following guidelines.

- When assigning I/O pin functions, select pin functions such that analog pins and digital pins are physically separated as much as possible.
- Each analog signal should be separated from all other signals as much as possible.
- PCB routing should isolate each analog signal as much as possible. Avoid routing any other signals, either analog or digital, in the same area.
- Ensure that analog supply voltages and analog reference voltages include appropriate AC filters. This may be in the form of recommended capacitors located near the MCU voltage pin, or appropriate inductive filters. The goal is to provide voltage supply and reference voltage with little or no voltage ripple.
- When using dedicated power layers in a PCB design, avoid routing digital signals in the areas of analog voltages, and avoid routing analog signals in the areas of digital voltages.

For highly sensitive applications, it is highly recommended to evaluate the specific design using simulation tools to understand the effect that circuit design has on the performance. For example, this may include applications such as precision sensor designs, or very high-speed digital bus interfaces. Refer to the *Electrical Characteristics* chapter in the Hardware User's Manual for the specific requirements for each peripheral function.

15.2 High Speed Signal Design Considerations

As clock speeds for digital signals increase, the influence of external stimuli on those signals can become more significant. Some peripheral functions can be classified as High Speed digital signals. Additional design considerations should be made for high-speed digital signals.

Crosstalk is a condition where transitions on one signal have an inductive influence on another nearby signal. When this crosstalk effect is strong enough, the first signal may cause errors on the second signal. To reduce the effects of crosstalk, use the following general PCB routing guidelines.

- Provide sufficient space between routed signals on the same routing layer. Generally, keep a minimum of one trace width space between signals of the same digital group, and a minimum of 3-5 trace widths space between signals of different digital groups.
- Provide extra space between clock signals and data signals on the same routing layer. Generally, keep a minimum of 3-5 trace widths space between clocks and any other digital signals.
- Avoid parallel routing of digital signals on any adjacent routing layers. If signals must be routed on adjacent signals layers, try to use only orthogonal crossings wherever possible.

If possible, separate PCB signal layers using power or ground layers between signal layers. The solid copper of the power or ground layer can act as a "shield" for the digital signals.

Each standardized interface will have specific requirements. To ensure that the PCB is designed to avoid signal crosstalk problems, it is strongly suggested to refer to the relevant standards for each interface in the design.

15.3 Signal Group Selections

Some pin names have an added _A, _B, or _C suffix to indicate signal groups. When assigning certain peripheral functions, such as IIC, SPI, SSIE, ETHERC, and SDHI, select the functional pins having the same suffix. In some cases, the AC timing characteristics shown in the "Electrical Characteristics" chapter of the Hardware User's Manual are measured for each signal group. If the signal groups are mixed, the peripheral is not guaranteed to function, and the stated AC timing characteristics may not apply.

If the pin names for a peripheral function to not have a signal group suffix, it is safe to select the most convenient pin assignment for each function signal.

Refer to the sections *Peripheral Select Settings for each Product* and *Notes on the PmnPFS Register Setting* in the *I/O Ports* chapter of the Hardware User's Manual.

16. References

The following documents were used in creating this Quick Design Guide. Visit the Renesas website for the latest version of each of these documents.

Reference	Document Number	Description
1	R01UH0884	Renesas RA6M1 Group, User's Manual: Hardware
2	R01UH0885	Renesas RA6M2 Group, User's Manual: Hardware
3	R01UH0886	Renesas RA6M3 Group, User's Manual: Hardware
4	R01UH0890	Renesas RA6M4 Group, User's Manual: Hardware
5	R01UH0891	Renesas RA6M5 Group, User's Manual: Hardware
6	R20AN0577	RA Arm® TrustZone® Tooling Primer
7	R11AN0467	Security Design with Arm® TrustZone® - IP Protection
8	R11AN0468	Securing Data at Rest Using the Arm® TrustZone®
9	R11AN0475	Establishing and Protecting Device Identity using SCE9 and Arm® TrustZone®
10	R11AN0469	Device Lifecycle Management Key Installation

Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb.18.21	—	First release document
1.01	Apr.28.21	—	Minor fixes to differentiate from Cortex-M33 devices. Fix statements on MOCO.
1.02	Apr.07.23	5-10, 12 21	Removed references to external pull up resistors on emulator signals Added note regarding requirement for serial programming on TrustZone® devices.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.