RZ/T2

Flexible Software Package v3.0 Compatibility Notice for Major Updates

Introduction

This document will describe the compatibility notice for Flexible Software Package (FSP) major updates that are being actively made to FSP v3.0 for Renesas RZ/T series in relation to the addition of new features or removal of deprecated features. As a reminder, FSP follows semantic versioning which means that backward incompatible changes should only occur in major releases. Bugs do cause backward incompatibility, but they are obviously not planned.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

Contents

1. What is considered a compatibility notice?

FSP defines a compatibility notice as an issue that requires users to make manual changes to files or settings they did not modify in their project when migrating FSP versions. The "files or settings they did not modify" is important. If the user made manual modifications to a FSP source file, migrated to a new FSP version, and they then run into an issue related to that source file, then that is not a FSP compatibility notice.

2. List of Compatibility Notices

2.1 BSP

2.1.1 Renamed CPU reset APIs to camel case

Issue

The following APIs have been renamed.

```
R_BSP_CPUReset()                           -> R_BSP_CpuReset()
R_BSP_CPUResetAutoRelease()                -> R_BSP_CpuResetAutoRelease()
R_BSP_CPUResetRelease()                    -> R_BSP_CpuResetRelease()
R_BSP_CPUClusterResetAutoReleaseControl() -> R_BSP_CpuClusterResetAutoReleaseControl()
```

Workaround

Existing projects using CPU reset APIs must be updated to replace it with new API names.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.1.2 Renamed internal FIQ setting functions

Issue

The following internal functions have been renamed.

```
__enable_all_exception()  -> r_bsp_enable_all_exception()
__disable_fiq()           -> r_bsp_disable_fiq()
__enable_fiq()            -> r_bsp_enable_fiq()
```

Workaround

Existing projects using internal FIQ setting functions must be updated to replace them with new function names.

Target Device

RZ/T2H

2.1.3 Changed the program location for a secondary or later project using RZ/T2H CR52 in multi-core processing from System SRAM to TCM

Issue

The program placement policy for a secondary or later project using RZ/T2H CR52 in multi-core processing was changed from placing everything in System SRAM to placing Second Stage Boot Loader (SSBL) to BTCM and others to ATCM.

Workaround

Existing projects specifying the System SRAM address directly for a secondary or later project using RZ/T2H CR52 in multi-core processing must be changed to a TCM address.

Target Device

RZ/T2H

2.1.4 Unified to device-independent noncache section addresses

Issue

The addresses of noncache sections have been unified to eliminate differences in addresses for each device and improved the readability and maintainability of linker scripts.

Old noncache section address

- Shared noncache address: the end address of the Mirror area of System SRAM*1 minus 0x60000

New noncache section address

- Noncache address: the end address of the Mirror area of System SRAM*1 minus 0x80000.
- Shared noncache address: the end address of the Mirror area of System SRAM*1 minus 0x20000

*1 Since the RZ/T2H CR52 does not have a mirror area of System SRAM, calculation is performed using the end address of System SRAM.

The actual address of each device is shown on the table below.

Table 1 noncache section address of each device

Device	Core type	Ordinal number of cores in multi-core processing	Noncache section		Shared_noncache_buffer section	
			Old start address	New start address	Old start address	New start address
RZ/T2M RZ/T2ME	CR52	Primary	0x3019_0000	0x3018_0000	0x301A_0000	0x301EA_0000
		Secondary or later	0x3001_0000	*4		
RZ/T2L	CR52		0x3009_0000	0x3008_0000	-	-
RZ/T2H	CR52	Primary	0x1019_0000	0x1018_0000	0x101A_0000	0x101E_0000
		Secondary or later	*2	*4		
	CA55	Primary	0x1039_0000	0x1038_0000	0x103A_0000	0x103E_0000
		Secondary or later	0x1023_0000*3	*4		

*2 The end address of the program placed in the cacheable area is aligned with 0x40.

*3 The address varies depending on the memory usage on the primary core.

*4 This address is aligned at 0x20 from the end address of the program placed in the primary core noncache section.
e.g. For RZ/T2H CR52, if the primary core uses up to 0x1018_0050, the secondary core will start at address 0x1018_0060.

Workaround

Existing projects specifying the noncache section address directly must be changed the address to new one.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.1.5 The part of noncache sections was generated as an independent file from a binary file for secondary or later project

Issue

The noncache sections and other binary files are generated separately. Due to the change in the address of the noncache sections (Refer to 2.1.4 Unified to device-independent noncache section addresses for details), if the binary files are not separated, very large binary files will be generated.

Workaround

Binary files generated by the existing project for a secondary or later project need to be replaced with two newly generated binary files.

Target Device

RZ/T2M, RZ/T2ME, RZ/T2H

2.1.6 Renamed macro used for timeout processing

Issue

The macro used for timeout processing renamed from `BSP_HARDWARE_REGISTER_WAIT_WTIH_TIMEOUT` to `BSP_HARDWARE_REGISTER_WAIT_WITH_TIMEOUT`. It is defined in `bsp_common.h`.

Workaround

Existing projects using `BSP_HARDWARE_REGISTER_WAIT_WTIH_TIMEOUT` must be updated to replace it with `BSP_HARDWARE_REGISTER_WAIT_WITH_TIMEOUT`.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.2 Analog

2.2.1 Removed mode setting in DSMIF configuration

Issue

General > [DEPRECATED]Mode in DSMIF configuration removed.

Workaround

Since RZ/T2 FSP v2.3.0 does not use this configuration, no action is required.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.3 Connectivity

2.3.1 Changed Interrupt settings of PCIE_EP and PCIE_RC from required to optional.

Issue

PCIE_EP and PCIE_RC interrupt events could be disabled on the FSP Configuration and were disabled by default.

Workaround

Existing projects using PCIE_EP and PCIE_RC modules should re-configure interrupt events on FSP Configuration.

Target Device

RZ/T2H

2.3.2 Removed Software RTS Port configuration item for SCI_UART

Issue

Module > Flow > Control > Software RTS Port on FSP Configuration was removed to set only available pins on each device.

Workaround

Existing projects using Software RTS as SCI_UART flow control should re-configure it on FSP Configuration.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.3.3 Renamed macro used to switch the behavior of SPI using DMAC

Issue

The macro used to switch the behavior of SPI using DMAC renamed from `SPI_DMAC_SUPPORT_ENABLE` to `SPI_DMA_SUPPORT_ENABLE`. It is output to `r_spi_cfg.h`.

Workaround

Existing projects using `SPI_DMAC_SUPPORT_ENABLE` must be updated to replace it with `SPI_DMA_SUPPORT_ENABLE`.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.3.4 Disabling interrupts of SPI when using DMAC have been changed to mandatory

Issue

Previously, it was not possible to choose to disable interrupts of SPI, but those settings have been added. Then, when using SPI with DMAC, these settings must be set to disable.

Workaround

Existing projects that use the SPI module must be updated to disable `Receive Interrupt Priority` and `Transmit Buffer Empty Interrupt Priority` when using DMAC.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.4 Networking

2.4.1 Removed Reset Port configuration item for Ethernet

Issue

`Module > Reset Port` on FSP Configuration was removed to set only available pins on each device.

Workaround

Existing projects using Ethernet module should re-configure reset pin on FSP Configuration.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.5 Security

2.5.1 Restructured RSIP APIs about key management

Issue

APIs and structure for key management have been reconstructed.

i. `rsip_wrapped_key_t`
   Old specifications:

```
typedef struct st_rsip_wrapped_key
{
    uint8_t alg;              ///< Internal algorithm ID
    uint8_t subtype;          ///< Internal key type ID
    uint8_t info[2];          ///< Reserved area
    uint8_t value[];          ///< Variable length array to store the key value
} rsip_wrapped_key_t;
```

- `rsip_wrapped_key_t` is defined as a variable-length structure. User declares an arbitrary buffer and casts it to a pointer of `rsip_wrapped_key_t*`.
- No initialization of `rsip_wrapped_key_t` is required.

New specifications:

```
typedef struct st_rsip_wrapped_key
{
    rsip_key_type_t type;     ///< Key type
    void          * p_value;  ///< Key value
} rsip_wrapped_key_t;
```

- User declares an arbitrary buffer and initializes `rsip_wrapped_key_t->p_value` as a pointer to the buffer.
- Each member (`rsip_wrapped_key_t->type` and `rsip_wrapped_key_t->p_value`) must be initialized.

ii. The following API arguments, `rsip_key_type_t const key_type` and `rsip_key_pair_type_t const key_pair_type`, have been replaced by `rsip_wrapped_key_t->type`.
   - `R_RSIP_KeyGenerate()`
   - `R_RSIP_KeyPairGenerate()`
   - `R_RSIP_EncryptedKeyWrap()`
   - `R_RSIP_RFC3394_KeyUnwrap()`
   - `R_RSIP_KDF_MACKeyImport()`
   - `R_RSIP_KDF_ECDHSecretKeyImport()`
   - `R_RSIP_KDF_DerivedKeyImport()`
   - `R_RSIP_InitialKeyWrap()`
   - `R_RSIP_PKI_RootCertKeyImport()`

iii. `R_RSIP_InjectedKeyImport()` have been obsoleted.

iv. `R_RSIP_InitialKeyUpdateKeyWrap()` have been merged to `R_RSIP_InitialKeyWrap()`.

v. The following API argument types used for key injection or key update have been changed to `void*`.
   - `R_RSIP_EncryptedKeyWrap()`
   - `R_RSIP_InitialKeyWrap()`
   - `R_RSIP_SB_InitialDecryptionKeyWrap()`
   - `R_RSIP_AuthPasswordHashCompute()`
   - `R_RSIP_PKI_InitialRootCertWrap()`

Workaround

i. Existing projects using RSIP must be updated to match the new specifications of `rsip_wrapped_key_t`.

ii. Existing projects using RSIP must update API arguments to `rsip_wrapped_key_t->type`.

iii. Existing projects using RSIP must initialize `rsip_wrapped_key_t` instead of using `R_RSIP_InjectedKeyImport()`.

iv. Existing projects using RSIP must implement the equivalent behavior of `R_RSIP_InitialKeyUpdateKeyWrap()` using `RSIP_KEY_TYPE_KUK` added to `rsip_key_type_t` and `R_RSIP_InitialKeyWrap()`.

v. Existing projects using RSIP must update API arguments to `void*`.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.5.2 Renamed APIs and structure of RSIP module

Issue

The following APIs have been renamed.

```
R_RSIP_KDF_MACKeyImport()              -> R_RSIP_KDF_HMAC_DKMKeyImport()
R_RSIP_KDF_ECDHSecretKeyImport()       -> R_RSIP_KDF_HMAC_ECDHSecretKeyImport()
R_RSIP_KDF_HMAC_MACUpdate()            -> R_RSIP_KDF_HMAC_DKMUpdate()
R_RSIP_KDF_MACConcatenate()            -> R_RSIP_KDF_DKMConcatenate()
R_RSIP_SB_InitialDecryptionKeyWrap()   -> R_RSIP_SB_InitialCommonKeyWrap()
```

The following structure has been renamed.

```
rsip_wrapped_mac_t                     -> rsip_wrapped_dkm_t
```

This structure is the argument of the following function.

```
R_RSIP_KDF_HMAC_DKMKeyImport()
R_RSIP_KDF_HMAC_DKMUpdate()
R_RSIP_KDF_HMAC_SignFinish()
R_RSIP_KDF_DKMConcatenate()
R_RSIP_KDF_DerivedKeyImport()
R_RSIP_KDF_DerivedIVWrap()
```

Workaround

Existing projects using the above APIs and structure must be updated to replace them with new names.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.5.3 Changed arguments of R_RSIP_SB_InitialCommonKeyWrap() and R_RSIP_AuthPasswordHashCompute()

Issue

The following APIs have changed their arguments. The last argument of each API has been changed from uint8_t * to rsip_sb_common_key_t*/rsip_hashed_auth_password_t* to clarify the output length. For other changes to these APIs, refer to 2.5.1 Restructured RSIP APIs about key **management**.

Old APIs:

```
R_RSIP_SB_InitialDecryptionKeyWrap (
      rsip_ctrl_t *const                   p_ctrl,
      rsip_wufpk_t const *const            p_wrapped_user_factory_programming_key,
      uint8_t const *const                 p_initial_vector,
      uint8_t const *const                 p_encrypted_key,
      uint8_t *const                       p_injected_key )
R_RSIP_AuthPasswordHashCompute (
      rsip_ctrl_t *const                   p_ctrl,
      rsip_wufpk_t *const                  p_wrapped_user_factory_programming_key,
      uint8_t const *const                 p_initial_vector,
      rsip_auth_type_t const               authentication_type,
      uint8_t const *const                 p_encrypted_password,
      uint8_t *const                       p_hashed_password )
```

New APIs:

```
R_RSIP_SB_InitialCommonKeyWrap (
      rsip_ctrl_t *const                    p_ctrl,
      void const *const                     p_wrapped_user_factory_programming_key,
      void const *const                     p_initial_vector,
      void const *const                     p_encrypted_key,
      rsip_sb_common_key_t *const           p_injected_key )
fsp_err_t R_RSIP_AuthPasswordHashCompute (
      rsip_ctrl_t *const                    p_ctrl,
      void const *const                     p_wrapped_user_factory_programming_key,
      void const *const                     p_initial_vector,
      rsip_auth_type_t const                authentication_type,
      void const *const                     p_encrypted_password,
      rsip_hashed_auth_password_t *const    p_hashed_password )
```

Workaround

Existing projects using `R_RSIP_SB_InitialCommonKeyWrap()` and `R_RSIP_AuthPasswordHashCompute()` must update their arguments to match the new specifications.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.6 Storage

2.6.1 Renamed address space configuration item of xSPI_HYPER, xSPI_OSPI and xSPI_QSPI

Issue

The configuration item for address space renamed from `Custom Address Space > Custom Address Space Enable` (`Disable`, `Enable`) to `Memory Mapping Address Space > Memory Mapping Address Space Configuration Support` (`Supported`, `Unsupported` (Fixed per device)).

Workaround

Existing projects that set `Custom Address Space Enable` to `Disable` must be updated to set the items that are placed under `Memory Mapping Address Space` on FSP Configuration.

Target Device

RZ/T2ME, RZ/T2H

2.6.2 Added DMAC Support configuration for SDHI

Issue

For SDHI module, `DMAC Support` configuration has been added in FSP Configuration. It is possible to select SDHI operation without using DMAC.

Workaround

Existing projects using SDHI module should be changed `DMAC Support` configuration from the default Disabled to Enabled for RZ/T2 FSP v2.3.0 equivalent operation.

Target Device

RZ/T2H

2.7 Timers

2.7.1 Changed arguments of R_MTU3_PeriodSet(), R_MTU3_StatusGet() and R_MTU3_InfoGet()

Issue

The following APIs have changed their arguments.

Old APIs

```
R_MTU3_PeriodSet (timer_ctrl_t * const p_ctrl, mtu3_counter_t * const p_counter)
R_MTU3_StatusGet (timer_ctrl_t * const p_ctrl, mtu3_status_t * const p_status)
R_MTU3_InfoGet (timer_ctrl_t * const p_ctrl, mtu3_info_t * const p_info)
```

New APIs

```
R_MTU3_PeriodSet (timer_ctrl_t * const p_ctrl, uint32_t const period)
R_MTU3_StatusGet (timer_ctrl_t * const p_ctrl, timer_status_t * const p_status)
R_MTU3_InfoGet (timer_ctrl_t * const p_ctrl, timer_info_t * const p_info)
```

Workaround

Existing projects using `R_MTU3_PeriodSet()`, `R_MTU3_StatusGet()` and `R_MTU3_InfoGet()` must update their arguments to match the new specifications.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.7.2 Changed configuration items of MTU3

Issue

To adapt the usage to GPT, the configuration items of MTU3 were changed. The changes made are shown below in red.

```
Module
|---General
    |--- Compare Match
         |--- Stauts
         |--- Compare match value
    |--- Name
    |--- Channel
    |--- Mode
    |--- ~~TGRA(Output Compare or Input Capture Value)~~
    |--- ~~TGRB(Output Compare or Input Capture Value)~~
    |--- ~~Time Prescaler~~
    |--- Buffer Operation Support
    |--- Period
    |--- Period Unit
    |--- Clock Edge
    |--- Counter Clear Source
|---Output
    |--- ~~Initial Output A~~
    |--- ~~Initial Output B~~
    |--- Duty Cycle Percent (only applicable in PWM mode)
    |--- MTIOCA Output Enabled
    |--- MTIOCA Stop Level
    |--- Initial Output Level (only applicable in PWM mode)
    |--- MTIOCB Output Enabled
    |--- MTIOCB Stop Level
    |--- Retain Output Level at Count Stop
|---Input
    |--- Input Capture
         |--- MTIOCnA Source
         |--- MTIOCnB Source
    |--- Phase Count
         |--- Bit Mode
         |--- Counting Mode
         |--- Clock Pin Select
|--- Custom Waveform
       |--- Custom Waveform Enable
       |--- TGRA(Ouput Compare or Input Capture Value)
       |--- TGRB(Ouput Compare or Input Capture Value)
       |--- TGRC(Ouput Compare or Input Capture Value)
       |--- TGRD(Ouput Compare or Input Capture Value)
       |--- MTIOCnA Pin Function
       |--- MTIOCnB Pin Function
       |--- Time Prescaler
```

Workaround

Existing projects using MTU3 module should configure new items on FSP Configuration.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.7.3 Removed MTU3 channel 5

Issue

MTU3 channel 5, which is not available in normal mode, was excluded from the channels supported in FSP.

Workaround

Existing projects using MTU3 channel 5 must be changed to another channel.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.7.4 Changed APIs used to start and stop the timer in the MTU3 input capture function

Issue

To adapt the usage to GPT, the flow for using MTU3 input capture function was changed.

Workaround

Existing projects using MTU3 channel 5 must be changed to another channel.

For starting a timer : `R_MTU3_Open`, `R_MTU3_Start` -> `R_MTU3_Open`, `R_MTU3_Enable`

For stopping a timer : `R_MTU3_Stop` -> `R_MTU3_Disable`

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.7.5 Changed where to configure MTU DSMIF error requests

Issue

The locations to set MTU DSMIF error requests were changed from MTUX Pin Control > Additional MTU0 pin control request condition to MTU0 Pin Control > Additional MTU0 pin control DSMIF Error X request condition (X = 0, 1). This is the same for MTU3/4 and MTU6/7.

Workaround

Existing projects using MTU DSMIF error requests must re-configure them on FSP Configuration.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME

2.7.6 Changed definition values of poe3_active_level_t

Issue

The enumeration(enum) member values of the `poe3_active_level_t` have been changed.

```
POE3_ACTIVE_LEVEL_HIGH: 0U -> 1U
POE3_ACTIVE_LEVEL_LOW : 1U -> 0U
```

Workaround

Existing projects using the above enum members should be checked to see if the operation is affected.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.8 Transfer

2.8.1 Renamed a member of dmac_extended_cfg_t

Issue

A structure member of DMAC module has been renamed as follows.

```
dmac_register_select_reverse_t next_register_operaion
-> dmac_register_select_reverse_t next_register_operation
```

Workaround

Existing projects using DMAC module should be updated to use the new member name.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.9 OS/Middleware

2.9.1 Changed configuration for FreeRTOS Port

Issue

When using FreeRTOS related sources, it is necessary to add FreeRTOS Port on Stacks tab of FSP Configuration as same as other HAL drivers.

Workaround

Existing projects using FreeRTOS related sources must be updated to add RTOS > FreeRTOS Port stack on FSP Configuration.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

2.10 FSP Configuration

2.10.1 Renamed ETHER_ETHn operation mode

Issue

ETHER_ETHn operation modes have been renamed to remove the voltage notations. The voltage is fixed depending on the mode.

```
Custom(1.8V)       : No change
Custom(3.3V)       : No change
MII mode(3.3V)    -> MII mode
RMII mode(3.3V)   -> RMII mode
RGMII mode(1.8V)  -> RGMII mode
```

Workaround

Existing projects using ETHER_ETHn pin must be re-configured on Pins tab of FSP Configuration.

Target Device

RZ/T2H

2.10.2 Changed default value for CPU MPU configuration

Issue

The default value for the CPU MPU memory region enable setting has been changed to Enabled. Therefore, when FSP is replaced by RZ/T2 FSP v3.0.0 in a project created with RZ/T2 FSP v2.X.X, the settings are changed from Disabled to Enabled. The target regions are as follows:

RZ/T2M : Region 13 to 21 (Used with default settings)

RZ/T2L : Region 13 to 20 (Used with default settings)

RZ/T2ME : Region 13 to 21 (Used with default settings)

RZ/T2H : Region 16 to 21 (Used with default settings)

Workaround

Existing projects using above regions with default value of CPU MPU configuration must change RZT2x Memory Config > CPU MPU > Region > Region (number) > Region enable to Disabled on BSP tab of FSP Configuration.

Target Device

RZ/T2M, RZ/T2L, RZ/T2ME, RZ/T2H

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul.16.25	-	First Edition issued.

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

1. 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

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/.

Trademarks

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

© 2025 Renesas Electronics Corporation. All rights reserved.