Introduction
As the use of electronics accelerates in automobiles, the scope of the development of software to run the electronics is becoming increasingly diverse and larger-scale. In response to the needs for quickly and efficiently developing high-quality software of this type, Renesas provides a lineup of various development tools. This application note describes the R-Car S4 series development environment.

Target Device
R-Car S4

Contents

  1.1 Structure of the R-Car S4 Software .......................................................... 3
  1.2 Software and Development Environment .............................................. 5
  1.3 Boards and Kits ..................................................................................... 7
  1.4 Development Environment for the R-Car S4 ....................................... 8

2. Lineup of Tools ...................................................................................... 9

3. Integrated Development Environment (IDE) ......................................... 10
  3.1 e² studio for R-Car* .............................................................................. 10
  3.2 Features .............................................................................................. 10

4. Emulators ............................................................................................. 11
  4.1 On-chip Debugging Emulator: E2 ......................................................... 11
  4.1.1 Features .......................................................................................... 11
  4.1.2 Multi-core Synchronous Debugging ................................................. 12
  4.1.3 Multi-core Synchronous Tracing ...................................................... 13
  4.2 Full-spec Emulator: IE850A ................................................................. 14
  4.2.1 Features .......................................................................................... 14
  4.2.2 Optional Product ............................................................................. 14

5. Compilers ............................................................................................ 15
  5.1 Compilers for the Arm Cores ............................................................... 15
  5.2 Compilers for the RH850 Cores ........................................................... 15

6. Coding Tools ....................................................................................... 16
  6.1 Smart Configurator (Pin Settings) ....................................................... 16
  6.2 Region ID Configurator ...................................................................... 16

7. Analysis Tools ...................................................................................... 17
  7.1 Bus Monitor ....................................................................................... 17
7.2 Synchronous Logging ................................................................. 18

8. Virtual Platform ............................................................................ 19
8.1 VDK ......................................................................................... 19
8.2 QEMU ...................................................................................... 19

Revision History .................................................................................. 20
1. Software and Tools for the R-Car

1.1 Structure of the R-Car S4 Software

R-Car S4 enables to launch Car Server/CoGW with high performance, high-speed networking, high security and high functional safety levels that are required as E/E architectures evolve into domains and zones. The R-Car S4 solution allows designers to re-use up to 88 percent of software code developed for 3rd generation R-Car SoCs and RH850 MCU applications. The software package supports software development for R-Car S4, including the real-time cores with various drivers and basic software such as Linux BSP and hypervisors.
1.2 Software and Development Environment

- R-Car software development kit (SDK)

See the product page. >>
See the download page. >>

The SDK for R-Car S4 is a software development kit that includes platform software which is required for connected cars. It operates in two environments: the R-Car S4 reference board (Spider) and R-Car S4 starter kit.

- Software optimized for automotive applications
- Includes the e2 studio for R-Car integrated software development environment
- Includes sample applications which are helpful in development

---

### R-CAR S4 SDK ARCHITECTURE

**SOFTWARE DEVELOPMENT KIT**

![R-CAR S4 SDK Architecture Diagram]

<table>
<thead>
<tr>
<th>CPU</th>
<th>Software</th>
<th>Description</th>
<th>Provider</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA55</td>
<td>Linux</td>
<td>Linux OS</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>BSP / Libraries</td>
<td>For Linux</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>OP-TEE/ TEE Client</td>
<td>Trusted OS</td>
<td>Renesas</td>
<td>OSS (BSD2)</td>
</tr>
<tr>
<td></td>
<td>TF-A</td>
<td>Trusted firmware</td>
<td>Renesas</td>
<td>OSS (BSD2)</td>
</tr>
<tr>
<td></td>
<td>XEN</td>
<td>Hypervisor</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>Guest Linux</td>
<td>for QNX Hypervisor</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>ICCOM</td>
<td>for Linux</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>IIUCOM</td>
<td>for QNX</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>AES ACC Library</td>
<td>Security Library</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>ICUMX Service Handler</td>
<td>Security Handler</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>SHIP-S Library</td>
<td>Security Library</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>IPL</td>
<td>Boot</td>
<td>Renesas</td>
<td>Proprietary</td>
</tr>
</tbody>
</table>
# R-Car S4 Series

<table>
<thead>
<tr>
<th>CPU</th>
<th>Software</th>
<th>Description</th>
<th>Provider</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA55</td>
<td>Linux</td>
<td>Linux OS</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>BSP / Libraries</td>
<td>For Linux</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>OP-TEE/ TEE Client</td>
<td>Trusted OS</td>
<td>Renesas</td>
<td>OSS (BSD2)</td>
</tr>
<tr>
<td></td>
<td>TP-A</td>
<td>Trusted firmware</td>
<td>Renesas</td>
<td>OSS (BSD2)</td>
</tr>
<tr>
<td></td>
<td>XEN</td>
<td>Hypervisor</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>Guest Linux</td>
<td>for QNX Hypervisor</td>
<td>Renesas</td>
<td>OSS (GPLv2)</td>
</tr>
<tr>
<td></td>
<td>ICCOM</td>
<td>for Linux</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>IIICOM</td>
<td>for QNX</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>AES ACC Library</td>
<td>Security Library</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>ICUMX Service Handler</td>
<td>Security Handler</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>SHIP-S Library</td>
<td>Security Library</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>IPL</td>
<td>Boot</td>
<td>Renesas</td>
<td>Proprietary</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU</th>
<th>Software</th>
<th>Description</th>
<th>Provider</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICUMX</td>
<td>Firmware</td>
<td>Security</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>FLS driver</td>
<td>Flash driver</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>IPL</td>
<td>Bootloader</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td>ICUMH</td>
<td>Firmware</td>
<td>Security</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
<tr>
<td></td>
<td>FLS driver</td>
<td>Flash driver</td>
<td>Renesas</td>
<td>REL Proprietary</td>
</tr>
</tbody>
</table>

## Tool

<table>
<thead>
<tr>
<th>Tool</th>
<th>Provider</th>
<th>License</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td></td>
</tr>
<tr>
<td>FLS driver</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td></td>
</tr>
<tr>
<td>Firmware</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td></td>
</tr>
<tr>
<td>FLS driver</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td></td>
</tr>
<tr>
<td>Secure Tool</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td></td>
</tr>
<tr>
<td>Secure Boot Signing</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td></td>
</tr>
<tr>
<td>MCU Secure Boot</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td></td>
</tr>
<tr>
<td>Flash Writter</td>
<td>Renesas</td>
<td>OSS(BSD2)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tool</th>
<th>Provider</th>
<th>License</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>e2 studio</td>
<td>Renesas</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Smart Configurator</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td>e2 studio Plug-in</td>
</tr>
<tr>
<td>Bus Monitor</td>
<td>Renesas</td>
<td>REL Proprietary</td>
<td>e2 studio Plug-in</td>
</tr>
</tbody>
</table>
1.3 Boards and Kits

- **R-Car S4 Reference Board (Spider)**
  This board has been developed to assist in evaluating the R-Car S4 device in a complete development environment with hardware and software for automotive gateway applications that support new E/E architectures. It allows the evaluation of all functions of the R-Car S4.

- **R-Car S4 Starter Kit**
  This is a compact and reasonably priced evaluation board for the R-Car S4. It allows the debugging of software for the IP modules and processors of the R-Car S4. Limitations apply to certain functions of the R-Car S4, such as the number of channels for LIN or CAN.
1.4 Development Environment for the R-Car S4

**IDE**
- Coding tool
- Editor
- Build tool
- Compiler
- Debugger and emulator

**Emulator**
- On-chip debugging emulator: E2
- Full-spec emulator: IE850A

**Software**
- Driver, etc. provided by the SDK
- Analysis tool
- Synchronous logging and bus monitor

**Virtual platform**
- Virtual development environment: R-Car Virtual Platform (VPF)*

*: Some tools are sold or provided by partners.
## 2. Lineup of Tools

Lineup of tools for the R-Car S4 series

<table>
<thead>
<tr>
<th>CPU</th>
<th>OS</th>
<th>Compiler</th>
<th>Emulator</th>
<th>Programming Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>On-chip Debugging</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Emulator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Full-spec Emulator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICUMX</td>
<td>—</td>
<td>Renesas compiler and Green Hills compiler</td>
<td><strong>E2 emulator</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*1, *2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G4MH/ICUMH</td>
<td>Autosar OS</td>
<td>Renesas compiler and Green Hills compiler</td>
<td><strong>E2 emulator</strong></td>
<td><strong>IE850A</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*1, *2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cortex-R52</td>
<td>Autosar OS</td>
<td>Arm compiler, etc.</td>
<td><strong>E2 emulator</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cortex-A55</td>
<td>Linux (Yocto), BlackBerry QNX</td>
<td>GCC compiler, etc.</td>
<td><strong>E2 emulator</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*1</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
1. The R-Car SDK includes the e² studio for R-Car and emulator software.
2. These are not included in the R-Car SDK, but only CS+ supports the emulator functions.
3. The flash writer (programming control program) operates on the ICUMX.
4. These are sold or provided by partners.
5. For Renesas compilers, you can purchase licenses or use free-evaluation editions.

- Tools for the R-Car S4 series from partners
  The R-Car S4 is supported by many tools from partners.

[R-Car Consortium >>]
3. Integrated Development Environment (IDE)

3.1 e² studio for R-Car*

The e² studio for R-Car is an IDE based on the open-source Eclipse IDE and the CDT plug-in to support development in C/C++.

Since the interface of the e² studio for R-Car is based on that of Eclipse, it will be easy to be use for customers who are familiar with Eclipse. It is also suitable in cases where a customer is preparing an appropriate environment for collaborative development with overseas companies.

Note: The e² studio for R-Car is an IDE for automotive SOCs from Renesas. Use the e² studio for R-Car that is part of the R-Car SDK.

3.2 Features

- The e² studio IDE covers all aspects of development.
- High-spec debugging functions in combination with the standard GNU debugger (GDB)
- High expandability as an Eclipse-based IDE
- Reuse of existing resources by importing projects from other IDEs

Editor

- Editor providing superior functions including code complementation according to the syntax of the C++ programming language

Code generation

- Support for the pin-setting feature of the Smart Configurator, easing the settings of pins

Building

- Support for an environment for building with the use of CMake

Debugging

- Support for the E2 and IE850A emulators from Renesas and for emulators from partners
- Debugging of multiple cores or multiple devices
- Debugging of Linux applications
- Support for virtual platform environments

Analysis

- Measuring bus loads in the SoC (bus monitoring)
4. Emulators

4.1 On-chip Debugging Emulator: E2

See the product page. >>

The E2 emulator (RTE0T00020KCE00000R) from Renesas is an advanced emulator developed with the concept of greater efficiency in development. The combination of basic debugging functions and various software and hardware solutions will contribute to reducing development times.

![E2 Emulator Image](image.jpg)

### 4.1.1 Features

- The E2 emulator can be used to debug the new next-generation MCUs and SoCs such as the RH850-family U2x series and R-Car S4 series that mount the G4MH core.
- The E2 emulator can be used to debug all processor cores among the Cortex-A55, Cortex-R52, G4MH, ICUMH, and ICUMX that are mounted on R-Car S4 series devices.
- For the R-Car S4-8 and R-Car S4N-8, both of which have two JTAG ports, using two E2 emulator units enables the simultaneous debugging of two types among the processor cores mounted on the R-Car S4 series.
- For the R-Car S4 series, multi-core synchronous debugging, that is, the synchronized debugging of multiple cores, is available. In addition, multi-core synchronous tracing, that is, tracing the execution times for multiple cores on the same time axis, is also available. Both facilities are further described later in this document.
4.1.2 Multi-core Synchronous Debugging

See the video. »>

See the blog. »>

The multiple heterogeneous processor cores that are mounted on the R-Car S4 can be debugged and traced simultaneously. This reduces the time required to identify the causes of problems and to analyze them in systems where multiple cores work together to achieve functionality.
4.1.3 Multi-core Synchronous Tracing

Multi-core synchronous tracing is a feature for synchronizing and checking trace information on each of multiple cores. Comparing and verifying the operations of software makes it easier to identify the problems that may arise in a multi-core system.

Execution history of the Arm cores/RH850 cores can be seen at the same time
4.2 Full-spec Emulator: IE850A
See the product page. >>

The IE850A (RTE0T0850AKCT00000J) full-spec emulator from Renesas is used to efficiently debug software of the G4MH and ICUMH cores of the R-Car S4. It is equipped with a rich variety of debugging functions, such as events, tracing, and time measurement.

4.2.1 Features
- The IE850A enables debugging of each of the processor cores (G4MH, ICUMH, or ICUMX) which are mounted on R-Car S4 series devices.
- Collect long-term program execution histories
  The emulator has 9 GB of memory for collecting histories of program execution, such as branches and data access. The emulator handles the memory as a set of trace frames, each with a capacity of up to 1 GB. The total capacity allows the saving and checking of histories of program execution over longer periods. Connecting the IE850A to an OCuLink connector which has been connected to the PCIe external pins of the R-Car S4 allows the acquisition of histories of program execution over long times.
- Rich variety of debugging functions
  The IE850A is equipped with a rich variety of debugging functions. Along with basic functions such as stepped execution, breaks, multi-core synchronous breaks, and multi-core synchronous tracing, it also supports time measurement for high-performance analysis and tracing to enable more efficient system development.

4.2.2 Optional Product
A tracing function is available by connecting the following optional product to the IE850A.

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Part Number for Ordering</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-Car Nexus Aurora Conversion Adapter</td>
<td>RTP8A779F0ASKB0SR0SN080</td>
<td>This product is used in conjunction with the IE850A to acquire Nexus Aurora tracing data from the R-Car S4.</td>
</tr>
</tbody>
</table>
### 5. Compilers

#### 5.1 Compilers for the Arm Cores

<table>
<thead>
<tr>
<th>CPU</th>
<th>OS</th>
<th>Supported Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex-R52</td>
<td>Autosar OS</td>
<td>Arm compiler, etc.*</td>
</tr>
<tr>
<td>Cortex-A55</td>
<td>Linux (Yocto),</td>
<td>GCC compiler, etc.*</td>
</tr>
<tr>
<td></td>
<td>BlackBerry QNX*</td>
<td></td>
</tr>
</tbody>
</table>

Note: The compilers are sold or provided by partners.

#### 5.2 Compilers for the RH850 Cores

<table>
<thead>
<tr>
<th>CPU</th>
<th>OS</th>
<th>Supported Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>G4MH/ICUMH</td>
<td>—</td>
<td>Renesas compiler</td>
</tr>
<tr>
<td>ICUMX</td>
<td>Autosar OS</td>
<td>Green Hills compiler*</td>
</tr>
</tbody>
</table>

Note: The compilers are sold or provided by partners.
6. Coding Tools

6.1 Smart Configurator (Pin Settings)

The Smart Configurator is a tool for setting the assignments of pins through a GUI. It enables visual checking and offers solutions for cases of contention for the same pin by multiplexed functions.

[Pin configuration] window (displayed per peripheral module)

Specifying a peripheral module in the [Pin configuration] window displays the pin functions for use by the module and allows you to set the assignment of the functions to pins.

[MCU/MPU Package] window

The [MCU/MPU Package] window displays a symbolic view of the pins, indicating by colors which pins are and are not in use, which have been assigned contending multiplexed functions, and so on. This makes it easy to check the overall situation.

Automatic resolution of contention between pin functions

In cases of contention for a pin, clicking on the pin brings up a list allowing automatic changing of the assignment of the pin to resolve the contention.

6.2 Region ID Configurator

The region ID configurator is a tool for generating header files for specifying the operations of each of the IP modules and setting the access attributes for IP resources.

- A region ID can be read when an application is operating under the Linux OS which runs on the R-Car S4.
- Generating header files for defining each register of the region ID is possible.
- An external xml file can be used to filter specific settings for each IP.
7. Analysis Tools

7.1 Bus Monitor

The bus monitor is a tool for verifying the states of bus occupancy by using a QoS (quality of service) tracing function. It can record and display statistics regarding DRAM access by various IPs.
7.2 Synchronous Logging

The synchronous logging tool is used to measure the changes over time of the processing loads and performance at the same time in a system consisting of multiple cores and devices. This eases identifying the bottlenecks for performance and the reasons for defects, in addition to the verification of system performance.
8. Virtual Platform

8.1 VDK

See the blog. >>

The “Virtual Turnkey Platform” an application software development environment for R-Car, is a pretested R-Car SDK on the R-Car VPF which enables immediate virtual development of application software. Since this platform reproduces the actual chip with a high degree of accuracy, users don’t need to use an actual board to start with the development environment, and multiple users can develop software on their own PCs or servers at the same time.

Use Cases and Expected Benefits

- You can start software design while pre-silicon, and seamlessly transition to software design using device samples.
- System verification is possible in pre-silicon when devices and control software can cooperate with each other.
- This can also be used for post-silicon software regression testing.

Main Features

- A CPU model is included. This allows the execution of software with the same binary code as that intended to run on the actual device.
- R/W access to memory space such as DRAM is possible.
- The address map and arithmetic precision will match perfectly with the corresponding devices.
- We are sequentially adding models for various IPs in response to market requirements.

Collaboration

We work closely with our partners and can leverage their tools, model libraries and other design assets.

- Renesas provides our partners with SoC specifications, design information and IP models.
- The partners themselves are also developing IP models and assembling R-Car VPFs.
- We are building an ecosystem in which many partners can participate, and the completed R-Car VPFs will be sold by the partners.

Please contact the following partners (to be added to in due course) for purchase information.

VDK (synopsys.com) (Product information)
Contact Synopsys (Contact)

8.2 QEMU

- The demand for virtual environments for software development is increasing due to the Software First philosophy.
- QEMU is an open-source virtual environment that dynamically converts programs for R-Car and simulates them on the host machine at high speed close to that of the actual machine.
- Providing a high-speed virtual environment based on QEMU will reduce the time for verification of software for Renesas devices in a virtual environment.
- Please contact a sales representative or distributor for product availability.
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Jun.30.23</td>
<td>First Edition issued</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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, sale, 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 electronics 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.

7. Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics products, 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 implants; 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.

8. 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 INDESTRUCTIBLE 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.

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 prevent 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 any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

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.

© 2023 Renesas Electronics Corporation. All rights reserved.