Abstract

This document describes the method of slave reception in clock synchronous serial I/O mode using high speed communication with DMAC for the M16C Group. When the clock synchronous serial I/O mode is in slave reception mode, and continuous reception mode is used, data may not be received correctly when the transmit/receive clock becomes high speed. When communicating in high speed, data can be received correctly using DMAC instead of continuous receive mode.

Products

MCUs: M16C/63, 64A, 64C, 65, 65C, 6C, 5LD, 56D, 5L, 56, 5M, and 57 Groups

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.
## Contents

1. Specifications ........................................................................................................ 3  
   1.1 Notes on Clock Synchronous Serial I/O Mode ...................................................... 5  
2. Operation Confirmation Conditions ..................................................................... 6  
3. Software .................................................................................................................. 7  
   3.1 Operation Overview ............................................................................................. 8  
   3.2 Required Memory Size ......................................................................................... 9  
   3.3 Constant ............................................................................................................... 9  
   3.4 Variables ............................................................................................................. 9  
   3.5 Functions ............................................................................................................ 10  
   3.6 Function Specifications ....................................................................................... 10  
   3.7 Flowcharts ......................................................................................................... 11  
      3.7.1 Main Processing .......................................................................................... 11  
      3.7.2 Peripheral Function Initialization .............................................................. 12  
4. Sample Code ........................................................................................................... 14  
5. Reference Documents ............................................................................................. 14
1. Specifications

In this application note, the clock synchronous serial I/O mode, and two channels of DMAC are used. Data is received continually 1 byte at a time, until all 256 bytes of data are received. Continuous receive mode is not used. Write dummy data to the UiTB register, and read the received data from the UiRB register using DMAC.

If continuous receive mode is used, when the transmit/receive clock becomes high speed, the read timing for the UiRB register may not adhere to the notes in section 1.1 Notes on Clock Synchronous Serial I/O Mode, thus data may not be received correctly (refer to Figure 1.1). In using DMAC for communication, even if the transmit/receive clock operates in high speed, data can be received correctly (refer to Figure 1.2).

Table 1.1 lists the Peripheral Functions and Their Applications. Figure 1.1 shows the Timing Diagram of Continuous Receive Mode, and Figure 1.2 shows Timing Diagram of Continuous Data Reception with DMAC.

Table 1.1  Peripheral Functions and Their Applications

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial interface (UART0)</td>
<td>Receiving data</td>
</tr>
<tr>
<td>DMAC (DMA0)</td>
<td>Writing dummy data</td>
</tr>
<tr>
<td>DMAC (DMA1)</td>
<td>Reading received data</td>
</tr>
</tbody>
</table>
Serial Interface: Using High Speed Communication when Performing Clock Synchronous Slave Reception with DMAC

When continuous receive mode can be used for receiving (when the transmit/receive clock operates at low speed)

When continuous receive mode cannot be used for receiving (when the transmit/receive clock operates in high speed)

Figure 1.1 Timing Diagram of Continuous Receive Mode
**1.1 Notes on Clock Synchronous Serial I/O Mode**

This section describes notes on clock synchronous serial I/O mode for the M16C/63, 64A, 64C, 65, 65C, 6C, 5LD, 56D, 5L, 56, 5M, and 57 Groups.

When starting transmission/reception while an external clock is selected, and the TXEPT bit in the UiC0 register is 1 (no data present in transmit register), the reception start condition must be met when the level of an external clock is the following:

- When the CKPOL bit in the UiC0 register is 0 (transmit data is output at the falling edge of transmit/receive clock and receive data is input at the rising edge), an external clock is high.
- When the CKPOL bit in the UiC0 register is 1 (transmit data is output at the rising edge of transmit/receive clock and receive data is input at the falling edge), an external clock is low.

The TXEPT bit becomes 0 at the next clock as dummy data is written to the UiTB register during communication. (The CLKI pin can be low as the TXEPT bit is 0.)

The received data in the UiRB register is transferred to the RAM using DMA1 transfer.

The above diagram assumes the following:
- The CKDIR bit in the UiMR register is 1 (external clock).
- The CRD bit in the UiC0 register is 1 (CTS/RTS function disabled).
- The CKPOL bit in the UiC0 register is 0 (transmit data is output at the falling edge of transmit/receive clock and receive data is input at the rising edge).
- The UiIRS bit in the UCON register is 0 (transmit buffer empty).
2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>M16C/65 Group</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td>• XIN Clock: 8 MHz&lt;br&gt;• CPU clock: 32 MHz (PLL clock: divided by 2, multiplied by 8)</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5 V (available between 3.3 to 5 V)</td>
</tr>
<tr>
<td>Integrated development</td>
<td>Renesas Electronics Corporation&lt;br&gt;High-performance Embedded Workshop Version 4.09</td>
</tr>
</tbody>
</table>
| environment                 | C compiler<br>Renesas Electronics Corporation<br>M16C Series/R8C Family C Compiler V.5.45 Release 01<br>Compile options<br>-c -Iinfo -dir "$(CONFIGDIR)"
(The default setting is used in the integrated development environment.) |
| Operating mode              | Single-chip mode                                                                             |
| Sample code version         | Version 1.00                                                                                 |
3. Software

The sample code uses a combination of UART0 clock synchronous I/O mode, DMAC0, and DMAC1 to receive a total of 256 bytes of data. Table 3.1 to Table 3.3 list the settings for UART0, DMAC0, and DMAC1.

Table 3.1 UART0 Setting Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation mode</td>
<td>Clock synchronous serial I/O mode</td>
</tr>
<tr>
<td>Transmit/receive clock</td>
<td>External clock</td>
</tr>
<tr>
<td>CTS/RTS function</td>
<td>Disabled</td>
</tr>
<tr>
<td>CLK polarity selection</td>
<td>Transmit data is output at the falling edge of the transmit/receive clock and receive data is input at the rising edge.</td>
</tr>
<tr>
<td>Bit order</td>
<td>LSB first</td>
</tr>
<tr>
<td>Transmit interrupt source</td>
<td>Transmit buffer empty (TI = 1)</td>
</tr>
<tr>
<td>Continuous receive mode</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

Table 3.2 DMAC0 Setting Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer unit</td>
<td>8 bits</td>
</tr>
<tr>
<td>Transfer mode</td>
<td>Single transfer</td>
</tr>
<tr>
<td>Transfer source address (RAM address)</td>
<td>Fixed</td>
</tr>
<tr>
<td>Transfer destination address (the address of U0TB register)</td>
<td>Fixed</td>
</tr>
<tr>
<td>DMA request source</td>
<td>UART0 transmission</td>
</tr>
<tr>
<td>Number of transfers</td>
<td>255</td>
</tr>
</tbody>
</table>

Table 3.3 DMAC1 Setting Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer unit</td>
<td>8 bits</td>
</tr>
<tr>
<td>Transfer mode</td>
<td>Single transfer</td>
</tr>
<tr>
<td>Transfer source address (the address of the U0RB register)</td>
<td>Fixed</td>
</tr>
<tr>
<td>Transfer destination address (RAM address)</td>
<td>Forward</td>
</tr>
<tr>
<td>DMA request source</td>
<td>UART0 reception</td>
</tr>
<tr>
<td>Number of transfers</td>
<td>256</td>
</tr>
</tbody>
</table>
### 3.1 Operation Overview

1. **Setting the reception start condition**
   Set the RE bit in the U0C1 register to 1 (reception enabled), and the TE bit to 1 (transmission enabled). Verify the CLK0 pin is high by a program, then write dummy data to the U0TB register.

2. **Starting DMA0 transfer**
   When the data in the U0TB register is transferred to the transmit register, the TI bit in the U0C1 register becomes 1 (no data present in U0TB register), and the TXEPT bit in the U0C0 register becomes 0 (data present in transmit register). Then the DMA0 transfer is started by a UART0 transmit interrupt request, and dummy data is transferred to the U0TB register from the RAM.

3. **Setting the reception start condition during transmitting (when TXEPT = 0)**
   When dummy data is written to the U0TB register by the DMA0 transfer, the TI bit becomes 0 (data present in UI0TB register). (1.1 Notes on Clock Synchronous Serial I/O Mode does not apply to this case as the TXEPT bit is 0.)

4. **Starting DMA1 transfer**
   When the reception is completed, the RI bit in the U0C1 register becomes 1 (data present in U0RB register). Then the DMA1 transfer is started by the UART0 receive interrupt request, and the received data is transferred from the U0RB register to the RAM. Repeat steps (2) to (4) until a total of 256 bytes of data is received.

5. **When 256-byte data reception is completed**
   When the total of 256 bytes of data is received, the receive operation is not performed for the subsequent data.

Figure 3.1 shows the Timing Diagram.
3.2 Required Memory Size

Table 3.4 lists the Required Memory Size.

<table>
<thead>
<tr>
<th>Memory Used</th>
<th>Size</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>229 bytes</td>
<td>In the r01an0732_src module</td>
</tr>
<tr>
<td>RAM</td>
<td>257 bytes</td>
<td>In the r01an0732_src module</td>
</tr>
<tr>
<td>Maximum user stack usage</td>
<td>14 bytes</td>
<td></td>
</tr>
<tr>
<td>Maximum interrupt stack usage</td>
<td>0 bytes</td>
<td></td>
</tr>
</tbody>
</table>

The required memory size varies depending on the C compiler version and compile options.

3.3 Constant

Table 3.5 lists the Constant Used in the Sample Code.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMMY_DATA</td>
<td>FFh</td>
<td>Dummy data</td>
</tr>
</tbody>
</table>

3.4 Variables

Table 3.6 lists the Global Variables.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsigned char</td>
<td>trn_data</td>
<td>Transmitted data</td>
<td>peripheral_init</td>
</tr>
<tr>
<td>unsigned char</td>
<td>rcv_data[256]</td>
<td>For received data storage</td>
<td>main, peripheral_init</td>
</tr>
</tbody>
</table>
3.5 Functions

Table 3.7 lists the Functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>mcu_init</td>
<td>CPU initialization</td>
</tr>
<tr>
<td>peripheral_init</td>
<td>Peripheral function initialization</td>
</tr>
</tbody>
</table>

3.6 Function Specifications

The following tables list the sample code function specifications.

**mcu_init**

<table>
<thead>
<tr>
<th>Outline</th>
<th>CPU initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void mcu_init(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Set the PLL clock divided by 2, and multiplied by 8 as the CPU clock.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Remark</td>
<td></td>
</tr>
</tbody>
</table>

**peripheral_init**

<table>
<thead>
<tr>
<th>Outline</th>
<th>Peripheral function initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void peripheral_init(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Initialize UART0, DMAC0, and DMAC1.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Remark</td>
<td></td>
</tr>
</tbody>
</table>
3.7 Flowcharts

3.7.1 Main Processing

Figure 3.2 shows the Main Processing.

```
main

- Disable maskable interrupts
  I flag ← 0

- CPU initialization
  mcu_init()

- Peripheral function initialization
  peripheral_init()

  Initialize UART0, DMAC0, and DMAC1.

- Initialize the array to store the received data

- Read dummy data
  tmp ← U0RB register

- Check the CLK0 pin
  Wait until the CLK0 pin becomes high.

- Write dummy data
  U0TB register ← FFh: Start continuous reception
```

Figure 3.2 Main Processing
### 3.7.2 Peripheral Function Initialization

Figure 3.3 and Figure 3.4 show Peripheral Function Initialization.

![Diagram showing Peripheral Function Initialization](image)

**Figure 3.3 Peripheral Function Initialization (1/2)**

- **Peripheral Init**
  - Set the UART0 transmit/receive mode register
    - U0MR register ← 09h
      - Bits SMD2 to SMD0 = 001b: Clock synchronous serial I/O mode
      - CKDIR bit = 1: External clock
  - Set UART0 transmit/receive control register 0
    - U0C0 register ← 10h
      - Bits CLK1 and CLK0 = 00b: f1SIO selected
      - CRD bit = 1: CTS/RTS function disabled
      - NCH bit = 0: CMOS output
      - CKPOL bit = 0: Transmit data is output at the falling edge of transmit/receive clock and receive data is input at the rising edge
      - UFORM bit = 0: LSB first
  - Set UART0 transmit/receive control register 1
    - U0C1 register ← 00h
      - TE bit = 0: Transmission disabled
      - RE bit = 0: Reception disabled
      - U0LCH bit = 0: No reverse
  - Select the UART0 transmit interrupt source
    - U0CON register ← 00h
      - U0IRS bit = 0: Transmit buffer empty (TI = 1)
  - Set the interrupt control register
    - S0TIC register ← 00h
      - Bits ILVL2 to ILVL0 = 000b: Level 0 (interrupt disabled)
    - S0RIC register ← 00h
      - Bits ILVL2 to ILVL0 = 000b: Level 0 (interrupt disabled)
  - Enable transmit/receive operation
    - U0C1 register ← 05h
      - TE bit = 1: Transmission enabled
      - RE bit = 1: Reception enabled
Serial Interface: Using High Speed Communication when Performing
Clock Synchronous Slave Reception with DMAC

Figure 3.4 Peripheral Function Initialization (2/2)

Set the DMA0 source pointer
SAR0 register ← &tm_data: Set the address of transmit data variable in the DMA0 source pointer
Set the DMA0 destination pointer
DAR0 register ← 024Ah: Set the address of the U0TB register in the DMA0 destination pointer
Select the DMA request source for DMA0
DM0SL register ← 0Ah
Bits DSEL4 to DSEL0 = 01010b: UART0 transmission
Set the DMA0 control register
DM0CON register ← 01h
DMBIT bit = 1: 8 bits
DMASL bit = 0: Single transfer
DSD bit = 0: Source address direction is fixed
DAD bit = 0: Destination address direction is fixed
Set the number of dummy data transfers
TCR0 register ← 254: Transfer dummy data 256 times. Set the number of transfers minus 1 in the TCR0 register as the number of DMA0 transfers. The first dummy data is written by a program, thus the set value is 254.
Set the DMA1 source pointer
SAR1 register ← 024Eh: Set the address of the U0RB register in the DMA1 source address pointer.
Set the DMA1 destination pointer
DAR1 register ← &rcv_data[0]: Set the top address of the reception data array in the DMA1 destination pointer.
Select the DMA request source for DMA1
DM1SL register ← 0Bh
Bits DSEL4 to DSEL0 = 01011b: UART0 reception
Set the DMA1 control register
DM1CON register ← 21h
DMBIT bit = 1: 8 bits
DMASL bit = 0: Single transfer
DSD bit = 0: Source address direction is fixed
DAD bit = 1: Destination address direction is forward
Set the number of transfers for the received data
TCR1 register ← 255: Receive data 256 times. Set the number of transfers minus 1 in the TCR1 register as the number of DMA1 transfers.
Set the interrupt control register
DMiIC register ← 00h
Bits ILVL2 to ILVL0 = 000b: Level 0 (interrupt disabled)
IR bit = 0: Interrupt not requested
Enable DMA0 and DMA1
DM0CON register ← 09h
DMAE bit = 1
DM1CON register ← 29h
DMAE bit = 1

return
4. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

5. Reference Documents

M16C/63 Group User’s Manual: Hardware Rev. 2.00
M16C/64A Group User’s Manual: Hardware Rev. 2.00
M16C/64C Group User’s Manual: Hardware Rev. 1.00
M16C/65 Group User’s Manual: Hardware Rev. 2.00
M16C/65C Group User’s Manual: Hardware Rev. 1.00
M16C/6C Group User’s Manual: Hardware Rev. 2.00
M16C/5L Group, M16C/56 Group User’s Manual: Hardware Rev. 1.10
M16C/5LD Group, M16C/56D Group User’s Manual: Hardware Rev. 1.10
M16C/5M Group, M16C/57 Group User’s Manual: Hardware Rev. 1.10

The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual
M16C Series/R8C Series C Compiler Package V.5.45
C Compiler User’s Manual Rev. 2.00

The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Nov. 30, 2011</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>

M16C/63, 64A, 64C, 65, 65C, 6C, 5LD, 56D, 5L, 56, 5M, and 57 Groups
Serial Interface: Using High Speed Communication when Performing
Clock Synchronous Slave Reception with DMAC

All trademarks and registered trademarks are the property of their respective owners.
General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on
   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   - The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products
   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.
   - The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.
Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.

2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.

4. 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.

6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors or omissions from the information included herein.

7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depend on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. You may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics products for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.