Abstract

This document describes the setting method and application example for using terminal software on a personal computer (hereinafter referred to as PC terminal software) to perform UART communication with the XIN clock in the M16C/63, 64A, 64C, 65, 65C, 6C, 5LD, 56D, 5L, 56, 5M, and 57 Groups. The M16C/65C MCU is used in this application note.

Products

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
2. Operation Confirmation Conditions ................................................................. 4
3. Hardware .............................................................................................................. 4
   3.1 Hardware Configuration ..................................................................................... 4
   3.2 Pins Used ........................................................................................................... 4
4. Software ............................................................................................................... 5
   4.1 Operation Overview .......................................................................................... 6
   4.2 Required Memory Size ..................................................................................... 7
   4.3 Constants .......................................................................................................... 7
   4.4 Variables .......................................................................................................... 8
   4.5 Functions .......................................................................................................... 8
   4.6 Function Specifications ...................................................................................... 9
   4.7 Flowcharts ....................................................................................................... 12
      4.7.1 Main Processing ............................................................................................ 12
      4.7.2 Peripheral Function Initialization ................................................................. 13
      4.7.3 Data Calculation .......................................................................................... 14
      4.7.4 Data Transmission ....................................................................................... 14
      4.7.5 Processing for Transmit Data Storage ........................................................ 15
      4.7.6 UART1 Transmit Interrupt Handler ............................................................. 16
      4.7.7 UART1 Receive Interrupt Handler ............................................................... 16
      4.7.8 Input Data Calculation Transmission .......................................................... 17
5. Sample Code ....................................................................................................... 18
6. Reference Documents .......................................................................................... 18
1. Specifications

Data transmission and reception are performed between the PC terminal software and the M16C/65C Group MCU using clock asynchronous serial I/O (UART) mode.

Table 1.1 lists the Peripheral Function and Its Application and Table 1.2 lists PC Terminal Software Settings.

Table 1.1 Peripheral Function and Its Application

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial interface (UART1)</td>
<td>Data transmission and reception with the PC terminal software</td>
</tr>
</tbody>
</table>

Table 1.2 PC Terminal Software Settings

<table>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits per second</td>
<td>115200 bps</td>
</tr>
<tr>
<td>Data bit</td>
<td>8 bits</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
<tr>
<td>Stop bit</td>
<td>1 bit</td>
</tr>
<tr>
<td>Flow control</td>
<td>None</td>
</tr>
</tbody>
</table>

Two numerical values are transmitted from the PC terminal software and received by the M16C/65C Group MCU. Then the M16C/65C Group MCU adds the two numerical values and the calculation result is transmitted to the PC terminal software.

Figure 1.1 shows the Operation Outline.

![Figure 1.1 Operation Outline](image)
2. **Operation Confirmation Conditions**

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>M16C/65C Group</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td>• XIN clock: 20 MHz</td>
</tr>
<tr>
<td></td>
<td>• CPU clock: 20 MHz (main clock: no division)</td>
</tr>
<tr>
<td></td>
<td>• Peripheral clock (UART): f1</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5.0 V (available from 2.7 to 5.0 V)</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics Corporation M16C Series, R8C Family C Compiler V.5.45 Release 01</td>
</tr>
<tr>
<td></td>
<td>Compile options -c -finfo -dir &quot;$(CONFIGDIR)&quot; (The default setting is used in the integrated development environment.)</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Single-chip mode</td>
</tr>
<tr>
<td>Sample code version</td>
<td>Version 1.00</td>
</tr>
<tr>
<td>Tool used</td>
<td>Terminal software</td>
</tr>
</tbody>
</table>

3. **Hardware**

3.1 **Hardware Configuration**

The PC terminal software is connected to the M16C/65C MCU.

Figure 3.1 shows a Connection Example.

![Connection Example Diagram](image)

3.2 **Pins Used**

Table 3.1 lists the Pins Used and Their Functions.

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P6_7/TXD1</td>
<td>Output</td>
<td>ASCII code output</td>
</tr>
<tr>
<td>P6_6/RXD1</td>
<td>Input</td>
<td>ASCII code input</td>
</tr>
</tbody>
</table>
4. Software

In the sample code, data transmission and reception are performed between the PC terminal software and the M16C/65C Group MCU using UART1 clock asynchronous serial I/O mode. First, the MCU receives the ‘single-digit number’, ‘+’, and ‘single-digit number’ transmitted by the PC terminal software. Then, the MCU adds the two single-digit numbers received and transmits the result to the PC terminal software.

The operation example is described below.

1. Use the keyboard to enter a single-digit number, press the ‘+’ key, enter a second single-digit number, and then press the ‘Enter’ key. For example, press the following keys: 5 + 7 Enter

The following table shows the ASCII code for the example just entered by the keyboard.

<table>
<thead>
<tr>
<th>Input from a PC</th>
<th>5</th>
<th>+</th>
<th>7</th>
<th>Enter key</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII code</td>
<td>35h</td>
<td>2Bh</td>
<td>37h</td>
<td>0Dh</td>
</tr>
</tbody>
</table>

2. The MCU adds the two single digit numbers entered in step (1) (i.e. 5 + 7 in the example).
3. The MCU transmits the ‘LF/NL’, ‘=’, ‘calculation result’ (i.e. 12 in the example), ‘CR’ and ‘LF/NL’ to the PC terminal software.

The following table shows the data transmitted from the MCU and the corresponding ASCII code.

<table>
<thead>
<tr>
<th>Data transmitted from the MCU</th>
<th>LF/NL (line feed)</th>
<th>=</th>
<th>1</th>
<th>2</th>
<th>CR (carriage return)</th>
<th>LF/NL (line feed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII code</td>
<td>0Ah</td>
<td>3Dh</td>
<td>31h</td>
<td>32h</td>
<td>0Dh</td>
<td>0Ah</td>
</tr>
</tbody>
</table>

Table 4.1 lists the UART1 Setting Conditions.

Table 4.1 UART1 Setting Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation mode</td>
<td>Clock asynchronous serial I/O mode</td>
</tr>
<tr>
<td>U1BRG count source</td>
<td>f1SIO</td>
</tr>
<tr>
<td>Character bits</td>
<td>8 bits</td>
</tr>
<tr>
<td>Bit rate</td>
<td>115200 bps</td>
</tr>
<tr>
<td></td>
<td>(20 MHz (f1SIO) / (16 × (10 + 1)) ≈ 115200)</td>
</tr>
<tr>
<td>Transmit/receive clock</td>
<td>Internal clock</td>
</tr>
<tr>
<td>Stop bit</td>
<td>1 bit</td>
</tr>
<tr>
<td>Parity</td>
<td>Disabled</td>
</tr>
<tr>
<td>TXD, RXD I/O polarity</td>
<td>Not inverted</td>
</tr>
<tr>
<td>Error signal</td>
<td>No output</td>
</tr>
<tr>
<td>Data output</td>
<td>CMOS output for the TXD1 pin</td>
</tr>
<tr>
<td>CTS/RTS function</td>
<td>Disabled</td>
</tr>
<tr>
<td>Bit order</td>
<td>LSB first</td>
</tr>
<tr>
<td>Interrupt source for transmission</td>
<td>Transmit buffer is empty (TI = 1)</td>
</tr>
<tr>
<td>Transmit interrupt priority level</td>
<td>Level 1</td>
</tr>
<tr>
<td>Receive interrupt priority level</td>
<td>Level 2</td>
</tr>
</tbody>
</table>
4.1 Operation Overview

The sample code operations are described below.

1. The first single-digit number is received by the MCU. The MCU then transmits it back to the PC terminal software. (1)
2. The addition symbol (2Bh) is received by the MCU. The MCU then transmits it back to the PC terminal software. (1)
3. The second single-digit number is received by the MCU. The MCU then transmits it back to the PC terminal software. (1)
4. The 'enter key' (0Dh) is received by the MCU. The MCU then transmits a 'CR' (0Dh) back to the PC terminal software. (1)
5. The MCU adds the first and second numerical values.
6. The MCU transmits the calculation result and accompanying data (LF/NL = 'calculation result' CR LF/NL which is 0Ah 3Dh 'calculation result' CR LF/NL in ASCII code).

Note:
1. The characters (ASCII code) received from the PC are transmitted back to the PC as they are for displaying them in the terminal software.

Figure 4.1 shows the Overview of the PC Terminal Software and Sample Code Operations.
4.2 Required Memory Size

Table 4.2 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>436 bytes</td>
<td>In the r01an0816_src.c module</td>
</tr>
<tr>
<td>RAM</td>
<td>13 bytes</td>
<td>In the r01an0816_src.c module</td>
</tr>
<tr>
<td>Maximum user stack usage</td>
<td>13 bytes</td>
<td></td>
</tr>
<tr>
<td>Maximum interrupt stack usage</td>
<td>23 bytes</td>
<td></td>
</tr>
</tbody>
</table>

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

4.3 Constants

Table 4.3 lists the Constants Used in the Sample Code.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1BRG_115200</td>
<td>10</td>
<td>Bit rate</td>
</tr>
<tr>
<td>INPUT_1ST_NUM</td>
<td>0</td>
<td>First value input mode</td>
</tr>
<tr>
<td>INPUT_2ND_NUM</td>
<td>1</td>
<td>Second value input mode</td>
</tr>
<tr>
<td>TRN_ST_MODE</td>
<td>2</td>
<td>Transmit start mode</td>
</tr>
<tr>
<td>TRN_BSY_MODE</td>
<td>3</td>
<td>Transmitting mode</td>
</tr>
<tr>
<td>ASCII_LF_NF</td>
<td>0Ah</td>
<td>LF/NL (line feed)</td>
</tr>
<tr>
<td>ASCII_CR</td>
<td>0Dh</td>
<td>CR (carriage return)</td>
</tr>
<tr>
<td>ASCII_PLUS</td>
<td>2Bh</td>
<td>+ (plus)</td>
</tr>
<tr>
<td>ASCII_EQUAL</td>
<td>3Dh</td>
<td>= (equal)</td>
</tr>
</tbody>
</table>
4.4 Variables

Table 4.4 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>mode</td>
<td>Mode</td>
<td>main, input_data_calc_trn, transmit, _uart1_transmit</td>
</tr>
<tr>
<td>unsigned char</td>
<td>rcv_buf</td>
<td>Receive buffer</td>
<td>input_data_calc_trn, _uart1_receive</td>
</tr>
<tr>
<td>unsigned char</td>
<td>digit_num</td>
<td>Digit number</td>
<td>input_data_calc_trn</td>
</tr>
<tr>
<td>unsigned char</td>
<td>num_1st</td>
<td>First numerical value</td>
<td>input_data_calc_trn</td>
</tr>
<tr>
<td>unsigned char</td>
<td>num_2nd</td>
<td>Second numerical value</td>
<td>input_data_calc_trn</td>
</tr>
<tr>
<td>unsigned char</td>
<td>num_sum</td>
<td>Calculation result (first value plus second value)</td>
<td>calculation</td>
</tr>
<tr>
<td>unsigned char</td>
<td>trn_data[ ]</td>
<td>Transmit data storage array</td>
<td>transmit, trn_data_storage, _uart1_transmit</td>
</tr>
<tr>
<td>unsigned char</td>
<td>cnt_trn</td>
<td>Transmit counter</td>
<td>transmit, trn_data_storage, _uart1_transmit</td>
</tr>
</tbody>
</table>

4.5 Functions

Table 4.5 lists the Functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>Main processing</td>
</tr>
<tr>
<td>mcu_init</td>
<td>CPU clock initialization</td>
</tr>
<tr>
<td>peripheral_init</td>
<td>Peripheral function initialization</td>
</tr>
<tr>
<td>calculation</td>
<td>Data calculation</td>
</tr>
<tr>
<td>transmit</td>
<td>Data transmission</td>
</tr>
<tr>
<td>trn_data_storage</td>
<td>Processing for transmit data storage</td>
</tr>
<tr>
<td>_uart1_transmit</td>
<td>UART1 transmit interrupt handler</td>
</tr>
<tr>
<td>_uart1_receive</td>
<td>UART1 receive interrupt handler</td>
</tr>
<tr>
<td>input_data_calc_trn</td>
<td>Input data calculation transmission</td>
</tr>
</tbody>
</table>
# 4.6 Function Specifications

The following tables list the sample code function specifications.

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Header</th>
<th>Declaration</th>
<th>Description</th>
<th>Argument</th>
<th>Returned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>Main processing</td>
<td>None</td>
<td>void main(void)</td>
<td>Call the calculation function and transmit function when in transmit start mode.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>mcu_init</td>
<td>CPU clock initialization</td>
<td>None</td>
<td>void mcu_init(void)</td>
<td>Set the main clock (no division) as the CPU clock.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>peripheral_init</td>
<td>Peripheral function initialization</td>
<td>None</td>
<td>void peripheral_init(void)</td>
<td>Initialize SFRs associated with UART1.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>calculation</td>
<td>Data calculation</td>
<td>None</td>
<td>void calculation(void)</td>
<td>Add the two received values. Call the trn_data_storage function to prepare the transmit data to be sent to the PC.</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
### transmit

<table>
<thead>
<tr>
<th>Outline</th>
<th>Data transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void transmit(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Set the first character (LF/NL) of the transmit data storage array to the U1TB register and transmit. Then set the mode to transmitting mode.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
</tbody>
</table>

### trn_data_storage

<table>
<thead>
<tr>
<th>Outline</th>
<th>Processing for transmit data storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void trn_data_storage(unsigned char sum_data)</td>
</tr>
<tr>
<td>Description</td>
<td>Store the transmit data in the transmit data storage array in order. The values of the calculation result are converted to ASCII code.</td>
</tr>
<tr>
<td>Argument</td>
<td>unsigned char sum_data: Calculation result</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
</tbody>
</table>

### _uart1_transmit

<table>
<thead>
<tr>
<th>Outline</th>
<th>UART1 transmit interrupt handler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void _uart1_transmit(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Transmit the data stored in the transmit data storage array in 1-byte units when in transmitting mode. When all data in the transmit data storage array is transmitted, set the mode to first value input mode.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
</tbody>
</table>

### _uart1_receive

<table>
<thead>
<tr>
<th>Outline</th>
<th>UART1 receive interrupt handler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void _uart1_receive(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Read the received data from the U1RB register. If an overrun error does not occur, store the data in the receive buffer, and call the input_data_calc_trn function.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
</tbody>
</table>
### `input_data_calc_trn`

<table>
<thead>
<tr>
<th><strong>Outline</strong></th>
<th>Input data calculation transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Header</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Declaration</strong></td>
<td><code>void input_data_calc_trn(void)</code></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Verify that the data received from the PC terminal software is correct data. If the data is verified, keep the data and transmit it to the PC terminal software. After 'CR' is transmitted, set the mode to transmit start mode.</td>
</tr>
<tr>
<td><strong>Argument</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Returned value</strong></td>
<td>None</td>
</tr>
</tbody>
</table>
4.7 Flowcharts

4.7.1 Main Processing

Figure 4.2 shows the Main Processing.

![Flowchart Diagram]

Figure 4.2 Main Processing
4.7.2 Peripheral Function Initialization

Figure 4.3 shows the Peripheral Function Initialization.

```
peripheral_init

Disable transmission and reception
U1C1 register ← 00h
TE bit = 0: Transmission disabled
RE bit = 0: Reception disabled

Set the UART1 transmit/receive mode register
U1MR register ← 05h
Bits SMD2 to SMD0 = 101b: UART mode character bit length is 8 bits
CKDIR = 0: Internal clock
STPS bit = 0: 1 stop bit
PRYE bit = 0: Parity disabled
IOPOL bit = 0: Not inverted

Set UART1 transmit/receive control register 0
U1C0 register ← 10h
Bits CLK1 and CLK0 = 00b: f1SIO
CRD bit = 1: CTS/RTS function disabled
NCH bit = 0: The TXD1 pin is CMOS output
UFORM bit = 0: LSB first

Set UART1 transmit/receive control register 1
U1C1 register ← 00h
U1LCH bit = 0: No reverse
U1ERE bit = 0: Error output disabled

Set UART1 transmit/receive control register 2
UCON register ← 00h
U1IRS bit = 0: Transmit buffer empty (TI = 1)

Set the bit rate to 115200 bps
U1BRG register ← 10h

Set the UART1 special mode registers
U1SMR register ← 00h
U1SMR2 register ← 00h
U1SMR3 register ← 00h
U1SMR4 register ← 00h

Enable the UART1 transmit interrupt
S1TIC register ← 01h
Bits ILVL2 to ILVL0 = 001b: Level 1
IR bit = 0: Interrupt not requested

Enable the UART1 receive interrupt
S1RIC register ← 02h
Bits ILVL2 to ILVL0 = 010b: Level 2
IR bit = 0: Interrupt not requested

Enable transmission and reception
U1C1 register ← 05h
TE bit = 1: Transmit enabled
RE bit = 1: Reception enabled

return
```

Figure 4.3 Peripheral Function Initialization
4.7.3 Data Calculation
Figure 4.4 shows the Data Calculation.

```
calculation
Add the first and second received values
Processing for transmit data storage
trn_data_storage()
return
```

**Figure 4.4 Data Calculation**

4.7.4 Data Transmission
Figure 4.5 shows the Data Transmission.

```
transmit
Wait for the transmit buffer to become empty
Transmit ‘LF/NL’
U1TB register ← trn_data[cnt_trn]
Set the mode to transmitting mode
return
```

**Figure 4.5 Data Transmission**
4.7.5 Processing for Transmit Data Storage

Figure 4.6 shows the Processing for Transmit Data Storage.

```
trn_data_storage

Argument
unsigned char sum_data: Calculation result

Initialize the storage array

Store 'LF/NL' in the array

Store 'CR' in the array

Store the lowest digit of the calculation result in the array

Convert the stored values to ASCII code

Update the lowest digit of the calculation result

All values of the calculation result stored in the array?

Yes

Store '=' in the array

Store 'LF/NL' in the array

Set the number of transmit data to the transmit counter

return

No
```

Figure 4.6 Processing for Transmit Data Storage
4.7.6 UART1 Transmit Interrupt Handler

Figure 4.7 shows the UART1 Transmit Interrupt Handler.

```plaintext
_uart1_transmit

Transmitting mode?

Yes

Update the transmit counter

Transmit the data

U1TB register ← trn_data[ cnt_trn ]

All data transmitted?

No

Yes

Set the mode to first value

input mode

return
```

Figure 4.7 UART1 Transmit Interrupt Handler

4.7.7 UART1 Receive Interrupt Handler

Figure 4.8 shows the UART1 Receive Interrupt Handler.

```plaintext
_uart1_receive

Read the received data

Overrun error occurred?

No

Yes

Error processing (1)

Store the received data in the receive buffer

Input data calculation

input_data_calc_trn()

return
```

Note:
1. The processing when an overrun error occurs is not performed in this application note. Add the processing as required.

Figure 4.8 UART1 Receive Interrupt Handler
4.7.8 Input Data Calculation Transmission

Figure 4.9 shows the Input Data Calculation Transmission.

Figure 4.9 Input Data Calculation Transmission
5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. 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/5LD Group, M16C/56D Group User’s Manual: Hardware Rev. 1.20
M16C/5L Group, M16C/56 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/contact
<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>May. 31, 2012</td>
<td>First edition issued</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>

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 for 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 manipulate 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 in 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. Further, 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.

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 range.

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 them 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.

SALES OFFICES
Renesas Electronics Corporation
http://www.renesas.com

Refer to “http://www.renesas.com” for the latest and detailed information.

Renesas Electronics America Inc.,
2890 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1701 McPhee Road, Nepean, Ontario L3Y 9C3, Canada
Tel: +1-613-898-5441, Fax: +1-613-898-3200

Renesas Electronics Europe Limited
Dukes Meadow, Milford Road, Basingstoke, Hampshire, RG24 8PH, U.K.
Tel: +44-1256-856-900, Fax: +44-1256-856-900

Renesas Electronics Europe GmbH
Am Schlosspark 12, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
29 Tai Ping Quan Zhen, No.22 Zhi Chu Lu, Haidian District, Beijing 100083, P.R.China
Tel: +86-10-6230-1110, Fax: +86-10-6230-7070

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, A31, No.1233 Ling Long Rd, Putong District, Shanghai 200012, China
Tel: +86-21-6577-7094, Fax: +86-21-6577-7898

Renesas Electronics Hong Kong Limited
Unit 1901-1913, 16/F., Tower 2, Grand Century Place, 133 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-0318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8753-9080, Fax: +886-2-8753-9587

Renesas Electronics Singapore Pte. Ltd.
1 HarbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6213-0101

Renesas Electronics Malaysia Sdn.Bhd.
Unit 902, Block B, Menara Amancon, Ampang Trade Centre, No. 18, Jln Persiaran Barat, 46500 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-2755-9300, Fax: +60-3-2755-9201

Renesas Electronics Korea Co., Ltd.
11F, Samseong-dong, Gangnam-gu, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141