Abstract

In this application note, an example of receiving two kinds of remote control signal formats is shown using pattern match mode of the remote control signal receiver.

Products

MCUs: M16C/63 Group
    M16C/64A Group
    M16C/64C Group
    M16C/65 Group
    M16C/65C Group

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. Reference Application Notes ...................................................................................... 4

4. Hardware Description .................................................................................................. 5
   4.1 Used pins .................................................................................................................. 5
   4.2 Reference Circuits .................................................................................................... 6

5. Software Description .................................................................................................. 7
   5.1 Operation Overview ................................................................................................ 7
   5.1.1 Operation of Receiving Pattern A format ............................................................ 7
   5.1.2 Operation of Receiving Pattern B format ............................................................ 10
   5.2 Required Memory Size .......................................................................................... 13
   5.3 Invariable Table ...................................................................................................... 14
   5.4 Variable Tables ....................................................................................................... 15
   5.5 Function Table ......................................................................................................... 15
   5.6 Function Specifications ......................................................................................... 16
   5.7 Flowchart ............................................................................................................... 18
   5.7.1 Main Processing .................................................................................................. 18
   5.7.2 Initialization Processing of MCU ......................................................................... 19
   5.7.3 Initialization Processing of PMC and Timer B2 .................................................. 20
   5.7.4 Interrupt Processing of PMC0 and Timer B2 ....................................................... 23
   5.7.5 Interrupt Processing of PMC1 ............................................................................. 25

6. Sample Code ............................................................................................................. 26

7. Reference Documents .............................................................................................. 26
1. Specifications

This application note describes receiving two separate formats of remote control signals: "pattern A (with header pattern and repeat code)" and "pattern B (with special header pattern)".

Table 1.1 shows the peripheral functions and their applications. Figure 1.1 and Figure 1.2 show the waveforms of pattern A and pattern B, respectively.

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMC0 circuit</td>
<td>Receive pattern A header pattern and data (see section 4.2)</td>
</tr>
<tr>
<td>PMC1 circuit</td>
<td>Receive pattern B header pattern and data (see section 4.3)</td>
</tr>
<tr>
<td>Timer B2</td>
<td>Receive pattern A repeat code (see section 4.2)</td>
</tr>
</tbody>
</table>

![Figure 1.1 Pattern A: Remote Control Format with Header Pattern and Repeat Code](image1)

![Figure 1.2 Pattern B: Remote Control Format with Special Header Pattern](image2)
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/65 Group (Program ROM 1: 256 KB)</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>20 MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5 V</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>Renesas Electronics products</td>
</tr>
<tr>
<td></td>
<td>High-performance Embedded Workshop V.4.08.00</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics products</td>
</tr>
<tr>
<td></td>
<td>M16C Series, R8C Family C Compiler V.5.45 Release 01</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Single-chip mode</td>
</tr>
</tbody>
</table>

3. Reference Application Note

The application note associated with this application note is listed below. Refer to this application note for additional information.

- M16C/63, 64A, 65 Groups (R01AN0390EJ0100) Remote Control Signal Receiver Setting by Format Type
4. Hardware

4.1 Pins Used

Table 4.1 lists the used pins and their functions.

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P9_2/TB2IN/PMC0</td>
<td>Input</td>
<td>Input of pattern A remote control signal</td>
</tr>
<tr>
<td>P9_1/PMC1</td>
<td>Input</td>
<td>Input of pattern B remote control signal</td>
</tr>
</tbody>
</table>
4.2 Reference Circuits

Figure 4.1 shows a connection example.

As voltage at point B is low, connecting PMC0/TB2 directly to PMC1 may make it difficult to differentiate with a high. By connecting a transistor, point C, PMC0/TB2, and PMC1 are connected, and the signals can be distinguished. By doing this, the PMC0/TB2 or PMC1 signal is actually an inverted signal of the remote control signal.

Figure 4.1  Connection Example
5. Software

5.1 Operation Overview

5.1.1 Receiving Pattern A Format

The PMC0 circuit receives the header and data of the pattern A format. Timer B2 receives the repeat code of the pattern A format. The settings are listed below.

Table 5.1 PMC0 Circuit Settings

<table>
<thead>
<tr>
<th>Item</th>
<th>PMC0 circuit</th>
<th>Timer B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count source</td>
<td>fC</td>
<td>f1</td>
</tr>
<tr>
<td>Division</td>
<td>No division</td>
<td>Divided by 64</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Pattern match mode</td>
<td>Pulse period/pulse width measurement mode</td>
</tr>
<tr>
<td>Pattern match mode</td>
<td>Header</td>
<td>Repeat code</td>
</tr>
<tr>
<td></td>
<td>Data 0 or data 1 match</td>
<td></td>
</tr>
<tr>
<td>Interrupt</td>
<td>Completion of data reception</td>
<td>Active edge of measurement pulse</td>
</tr>
<tr>
<td>Selected function</td>
<td>Input signal not inverted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Digital filter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error flag hold</td>
<td></td>
</tr>
<tr>
<td>Input pin</td>
<td>P9_2</td>
<td></td>
</tr>
</tbody>
</table>
Operation

(1) Reception begins at the first rising edge of the header pattern.
(2) When receiving, data is sequentially stored bit by bit in the PMC0DATi register (i = 0 to 3).
(3) After 32-bit data is received, the data reception completion interrupt is generated if there is no change in the signal of time which is longer than the setting value in registers PMCiHDPMAX, PMCiD0PMAX, and PMCiD1PMAX (i = 0, 1).
(4) After reading the error flag in the PMC0 data reception complete interrupt, if the REFLG bit is 0 (no error occurs), disable the PMC0 circuit (set the EN bit in the PMC0CON0 register to 0) \(^{(3)}\), and timer B2 starts counting after setting the initial value to timer B2. If the REFLG bit is 1 (error occurs), keep the PMC0 circuit enabled and timer B2 disabled, and exit the interrupt handler.
(5) If a repeat signal is received during the setting period, perform the following in the timer B2 interrupt routine: stop timer B2, reset the amount of time until the next repeat signal comes, and restart the timer B2 count.
(6) If there is no repeat signal during the setting period, the MCU enters the timer B2 overflow interrupt, the PMC0 circuit is reenabled \(^{(1)}\), and timer B2 is disabled \(^{(2)}\).

Notes:
1. The PMCi circuit starts operating by setting the EN bit to 1 (operation enabled) and the ENFLG bit becomes 1 (operating) (i = 0, 1). After setting the EN bit to 1, it takes up to two cycles of the count source before the ENFLG bit becomes 1. During this period, do not access bits or registers associated with the PMCi circuit except for the ENFLG bit. When the EN bit is set to 0 (operation disabled), the PMCi circuit stops operating and the ENFLG bit becomes 0 (operation stopped). After setting the EN bit to 0, it takes up to one cycle of the count source before the ENFLG bit becomes 0.
2. The MR3 bit (timer Bi overflow flag) is undefined after reset. The MR3 bit is cleared to 0 (no overflow) by writing to the TBiMR register (i = 0 to 5). The MR3 bit cannot be set to 1 by a program.
Figure 5.1 shows the status operation and interrupt generation timing of the PMC0 circuit remote control signal during reception.

**Notes:**
1. Because a transistor is connected between IR remote receiver and the PMC0 circuit, the actual input signal of the PMC0 circuit is an inverted signal.
2. Because the receiving mode of the pattern A signal is the pulse period measurement mode, the end signal can be ignored.
3. The MR3 bit becomes to 0 (no overflow) when writing to TB2MR register.
5.1.2 Receiving Pattern B Format

The PMC1 circuit receives the header and data of the pattern B format. The settings are listed below.

Table 5.2 PMC1 Circuit Settings

<table>
<thead>
<tr>
<th>Item</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count source</td>
<td>Clock source fC</td>
</tr>
<tr>
<td></td>
<td>Division No division</td>
</tr>
<tr>
<td>Operating mode</td>
<td>Pattern match mode</td>
</tr>
<tr>
<td>Pattern match mode</td>
<td>Detection patterns Header</td>
</tr>
<tr>
<td></td>
<td>Data 0 or data 1 match</td>
</tr>
<tr>
<td></td>
<td>Interrupt Header pattern match</td>
</tr>
<tr>
<td></td>
<td>Data 0 or data 1 match</td>
</tr>
<tr>
<td></td>
<td>Receive error</td>
</tr>
<tr>
<td></td>
<td>Completion of data reception</td>
</tr>
<tr>
<td></td>
<td>Selected function Input signal not inverted</td>
</tr>
<tr>
<td></td>
<td>Digital filter</td>
</tr>
<tr>
<td></td>
<td>Error flag hold</td>
</tr>
<tr>
<td></td>
<td>Input pin P9_1</td>
</tr>
</tbody>
</table>
The measurement condition of the remote control signal in pattern B is selected by setting bits TYP1 to TYP0 to 10b (pulse width measurement (between rising edge and falling edge, and falling edge and rising edge)).

The low level width and high level width of each bit are measured. Therefore, a data 0 or data 1 match interrupt is generated twice when 1 bit is received.

In this application note, the signal whose width is 0.6 ms is judged as data 0, and the signal whose width is 1.2 ms is judged as data 1.

The determination above should be performed at every data 0 and data 1 match interrupt.

The received data is encoded to bit 0 or bit 1 based on the low width and high width of each bit in the reception completion interrupt routine.

**Operation**

1. Start the reception operation at the first rising edge of the header.
2. In the header interrupt routine, enable the data 0 and data 1 match interrupts, data reception completion interrupt, and reception error interrupt.
3. In the data 0/data1 match interrupt routine, whether the data is valid or invalid is determined by the reception count value. Data is invalid at an even number of reception times, and data is valid at an odd number of reception times. When data 0/data1 is judged as valid, it is stored sequentially
4. In the data reception completion interrupt, received data is encoded to bit 0 or bit 1 according to the low width and high width of each bit.
5. When a signal is affected by noise and causes an error, to prevent further data from being received, the data 0/data1 match interrupt and reception error interrupts are disabled in the error interrupt processing.

**Note:**

1. As there is no PMC1 receive data store register i (PMC1DATi), the user must self-define it.
Figure 5.2 shows the status operation and interrupt generation timing when the PMC1 circuit remote control signal during reception.

Figure 5.2  PMC1 Reception Timing of the Remote Control with Header and Repeat Code Format

Notes:
1. Because the transistor is connected between IR remote receiver and PMCO, the actual input signal of PMCO is the reversed signal.
2. Because the header of pattern B is special, bits TYP1 and TYP0 are set to 10b (pulse width measurement mode) for the measurement.
3. Disable data 0/data 1 match interrupt and the reception error interrupt in the reception error interrupt routine.
5.2 Required Memory Size

Table 5.3 lists the required memory size.

Table 5.3  Required Memory Size

<table>
<thead>
<tr>
<th>Memory Used</th>
<th>Size</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>1528 Bytes</td>
<td></td>
</tr>
<tr>
<td>RAM</td>
<td>1543 Bytes</td>
<td></td>
</tr>
<tr>
<td>Maximum user stack</td>
<td>23 Bytes</td>
<td></td>
</tr>
<tr>
<td>Maximum interrupt stack</td>
<td>23 Bytes</td>
<td></td>
</tr>
</tbody>
</table>

The required memory size varies depending on the C compiler version and compiler options.
### 5.3 Invariable Table

Table 5.4 lists the invariables used in the sample code.

<table>
<thead>
<tr>
<th>Invariable Name</th>
<th>Setting Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISABLE</td>
<td>0</td>
<td>Disable</td>
</tr>
<tr>
<td>ENABLE</td>
<td>1</td>
<td>Enable</td>
</tr>
<tr>
<td>PMC0_HEAD_MIN</td>
<td>397</td>
<td>PMC0 header pattern set (MIN)</td>
</tr>
<tr>
<td>PMC0_HEAD_MAX</td>
<td>486</td>
<td>PMC0 header pattern set (MAX)</td>
</tr>
<tr>
<td>MEASURE_VALUE_MAX</td>
<td>0x0F2C</td>
<td>Maximum value of the pattern A special data (repeat) (MAX)</td>
</tr>
<tr>
<td>MEASURE_VALUE_MIN</td>
<td>0x0C6A</td>
<td>Minimum value of the pattern A special data (repeat) (MIN)</td>
</tr>
<tr>
<td>PMC0_DATA0_MIN</td>
<td>31</td>
<td>PMC0 data 0 pattern set (MIN)</td>
</tr>
<tr>
<td>PMC0_DATA0_MAX</td>
<td>39</td>
<td>PMC0 data 0 pattern set (MAX)</td>
</tr>
<tr>
<td>PMC0_DATA1_MIN</td>
<td>67</td>
<td>PMC0 data 1 pattern set (MIN)</td>
</tr>
<tr>
<td>PMC0_DATA1_MAX</td>
<td>82</td>
<td>PMC0 data 1 pattern set (MAX)</td>
</tr>
<tr>
<td>PMC1_HEAD_MIN</td>
<td>70</td>
<td>PMC1 header pattern set (MIN)</td>
</tr>
<tr>
<td>PMC1_HEAD_MAX</td>
<td>86</td>
<td>PMC1 header pattern set (MAX)</td>
</tr>
<tr>
<td>PMC1_DATA0_MIN</td>
<td>17</td>
<td>PMC1 data 0 pattern set (MIN)</td>
</tr>
<tr>
<td>PMC1_DATA0_MAX</td>
<td>21</td>
<td>PMC1 data 0 pattern set (MAX)</td>
</tr>
<tr>
<td>PMC1_DATA1_MIN</td>
<td>34</td>
<td>PMC1 data 0 pattern set (MIN)</td>
</tr>
<tr>
<td>PMC1_DATA1_MAX</td>
<td>42</td>
<td>PMC1 data 0 pattern set (MAX)</td>
</tr>
<tr>
<td>EN_PMC</td>
<td>0x01</td>
<td>Enable PMCi (i = 0, 1)</td>
</tr>
<tr>
<td>COUNT_TB2</td>
<td>0x5000</td>
<td>Timer B2 counts 0x5000</td>
</tr>
<tr>
<td>TB2S_EN</td>
<td>0x80</td>
<td>Enable timer B2 count</td>
</tr>
<tr>
<td>f64TIMAB</td>
<td>0x03</td>
<td>Count source of timer B2</td>
</tr>
</tbody>
</table>
5.4 Variable Table

Table 5.5 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>int</td>
<td>odd_even_bit</td>
<td>Count the number of bit received from the PMC1 circuit. Use the count value to determine if the number of received bits is even or odd.</td>
<td>_remote_control_1</td>
</tr>
<tr>
<td>char</td>
<td>PMC1_data[2]</td>
<td>Store the received data using the PMC1 circuit.</td>
<td>_remote_control_1</td>
</tr>
<tr>
<td>int</td>
<td>bits</td>
<td>Offset value to store received data in PMC1_data[].</td>
<td>_remote_control_1</td>
</tr>
<tr>
<td>_Bool</td>
<td>PMC1_ERFLG</td>
<td>PMC1 error flag 0: No error 1: Error</td>
<td>_remote_control_1</td>
</tr>
</tbody>
</table>

5.5 Function Table

Table 5.6 lists the functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock_Init</td>
<td>Initialize the clock</td>
</tr>
<tr>
<td>PMC_Init</td>
<td>Initialize PMC1 and PMC0</td>
</tr>
<tr>
<td>TB2_Init</td>
<td>Initialize timer B2</td>
</tr>
<tr>
<td>_remote_control_0</td>
<td>Interrupt function of PMC0</td>
</tr>
<tr>
<td>_remote_control_1</td>
<td>Interrupt function of PMC1</td>
</tr>
<tr>
<td>_timer_b2</td>
<td>Interrupt function of timer B2</td>
</tr>
</tbody>
</table>
## 5.6 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>Clock_Init</td>
<td>Clock initialization function</td>
<td>None</td>
<td>void Clock_Init(void)</td>
<td>Initialize the CPU clock and sub clock.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>PMC_Init</td>
<td>Remote control initialization function</td>
<td>None</td>
<td>void PMC_Init(void)</td>
<td>Initialize the PMC0 and PMC1 circuits.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>TB2_Init</td>
<td>Timer B2 initialization function</td>
<td>None</td>
<td>void TB2_Init(void)</td>
<td>Initialize timer B2.</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>_remote_control_0</td>
<td>PMC0 interrupt function</td>
<td>None</td>
<td>void _remote_control_0(void)</td>
<td>Receive the header and data of pattern A format.</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
### _timer_b2

<table>
<thead>
<tr>
<th>Outline</th>
<th>Timer B2 interrupt function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void _timer_b2(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Receive the repeat code of pattern A format.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
</tbody>
</table>

### _remote_control_1

<table>
<thead>
<tr>
<th>Outline</th>
<th>PMC1 interrupt function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void _remote_control_1(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Receive the header and data of pattern B format.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
</tbody>
</table>
5.7 Flowchart

5.7.1 Main Processing

Figure 5.3 shows the main processing.

![Flowchart](image_url)

Figure 5.3 Main Processing

- **main**
  - Disable maskable interrupts
  - Initial setting of CPU Clock_Init()
    - Main clock (no division) set as the CPU clock, and sub clock (fC) setting.
  - Initial setting of remote control PMC_Init()
    - Remote control initial setting
  - Initial setting of timer B2 TB2_Init()
    - Timer B2 initial setting: Pulse period measurement mode
  - Enable maskable interrupts
5.7.2 MCU Initialization Processing

Figure 5.4 shows the MCU initialization processing.

Figure 5.4  MCU Initialization Processing

Note:
1. Main clock is 20 MHz, and fC (32.768 kHz) is initialized.
5.7.3 PMCI and Timer B2 Initialization Processing

Figure 5.5 to Figure 5.7 show the PMCI and timer B2 initialization processing.

```
PMCI_Init

- Switch interrupt source
  - IFSR24 bit ← 1: Remote signal receive function 0
  - IFSR25 bit ← 1: Remote signal receive function 1

- Select interrupt priority level
  - PMCI1C register ← 01h
    - Bits ILVL2 to ILVL0 = 001b: Level 1
  - PMCO1C register ← 01h
    - Bits ILVL2 to ILVL0 = 001b: Level 1
  - PMCO0CON3 register ← 20h
    - Bits CSRC1 and CSRC0 = 10b: fC
  - PMCO0CON2 register ← 40h
    - Bits PSEL1 and PSEL0 = 01b: Input pin - PMCO pin
  - PMCO0CON0 register ← 9Ch
    - Bits DRINT1 and DRINT0 = 10b: Interrupt is generated when no receive error occurs and reception is completed
    - HDEN bit = 1: Header enabled
    - EHANDLE = 1: State of the REFLG bit in the PMCOSTS register - Hold even after next data received
    - FIL bit = 1: Filter enabled
  - PMCO0CON1 register ← 00h
    - Bits TYP1 and TYP0 = 00b: Period measurement (between rising edge and rising edge)
    - EXHDEN bit = 0: Special pattern detect block select bit - PMCO
  - PMCO0INT register ← 04h
    - DRINT bit = 1: Enable data reception complete interrupt

- PMC0 initial setting
  - PMC0CON3 register ← 20h
    - Bits CSRC1 and CSRC0 = 10b: fC
  - PMC1CON2 register ← 80h
    - Bits PSEL1 and PSEL0 = 10b: Input pin - PMC1 pin
  - PMC1CON0 register ← 14h
    - HDEN bit = 1: Header enabled
    - FIL bit = 1: Filter enabled
  - PMC1CON1 register ← 02h
    - Bits TYP1 and TYP0 = 10b: Pulse width measurement (between rising edge and falling edge, and falling edge and rising edge)
  - PMC1INT register ← 36h
    - PTDINT bit = 1: Enable data 0/data 1 match flag interrupt
    - DRINT bit = 1: Enable data reception complete interrupt
    - REINT bit = 1: Enable receive error flag interrupt
```

Note:
1. Bits TYP1 and TYP0 are set to 10b (pulse width measurement) for the header measurement of pattern B.

Figure 5.5 PMCI Initialization Processing (1/2)
M16C/63, 64A, 64C, 65, 65C Groups  Remote Control Signal Receiver: Receiving Two Separate Formats

Figure 5.6  PMCi Initialization Processing (2/2)

Returns

Notes:
1. Pattern B takes two measurements (high width and low width) of a single waveform. The low width (an odd number of times) is determined as data 0 pattern (0.6 ms) or data 1 pattern (1.2 ms), and the data is encoded.
2. The PMCi circuit starts operating by setting the EN bit to 1 (operation enabled) and the ENFLG bit becomes 1 (operating). After setting the EN bit to 1, it takes up to two cycles of the count source before the ENFLG bit becomes 1. During this period, do not access bits or registers associated with the PMCi except for the ENFLG bit.

Set the PMC0 pattern

Set the PMC1 pattern

Enable PMC0 operation

Enable PMC1 operation

Wait for PMC0 to start

Wait for PMC0 to start

Read PMC0CON2 register

Read PMC1CON2 register

EN bit = 1: Enable PMC0 operation

EN bit = 1: Enable PMC1 operation

ENFLG bit = 0; PMC0 stops

ENFLG bit = 0; PMC1 stops

ENFLG bit = 1: PMC0 operating

ENFLG bit = 1: PMC1 operating
M16C/63, 64A, 64C, 65, 65C Groups  Remote Control Signal Receiver: Receiving Two Separate Formats

Figure 5.7  Timer B2 Initialization Processing

- **TB2_Init**
  - Enable writing to PRCR register
  - Timer B2 initial setting
    - PRCR register ← 01h
      - PRC0 bit = 1: Enable writing to registers CM0, CM1, CM2, PLC0, PCLKR, and FRA0.
    - CM21 bit ← 0: Main clock or PLL clock
    - TCDIV00 bit ← 0: Clock select prior to timer AB division bit - fOCO-F
    - TBCS1 register ← 0Bh
      - Bits TCS2 to TCS0 = 011b: Timer B2 count source - f64TIMAB
      - TCS3 bit = 1: TCK0 to TCK1 disabled, TCS2 to TCS0 enabled
    - TB2MR register ← 06h
      - Bits TMOD1 to TMOD0 = 10b: Pulse period/pulse width measurement modes
      - Bits MR1 to MR0 = 01b: Pulse period measurement (measurement between a rising edge and the next rising edge of a measured pulse)
    - TB2 register ← 5000h (1): Set the initial value.
    - TB2IC register ← 01h
      - Bits ILVL2 to ILVL0 = 001b: Level 1
  - PRCR register ← 00h
  - Disable writing to PRCR register
  - return

Note:
1. The reception interval for the pattern A repeat code is 108 ms (< 144 ms) which is why the timer B2 overflow time is set to 144 ms.
   The timer B2 overflow time is 1 × 20 MHz (0xFFFFh + 1 - 5000h (timer B2 register value)) × 64 = 144 ms.
   In the timer B2 interrupt handler, determine if repeat code exists by confirming if overflow has occurred or not.
5.7.4 PMC0 and Timer B2 Interrupt Handling

Figure 5.8 and Figure 5.9 show the interrupt handling of PMC0 and Timer B2, respectively.

---

**Figure 5.8 PMC0 Interrupt Handling**

- **remote_control_0 (1)**
  - Clear the IR bit in the PMC0IC register
  - PMC0IC register ← 01h
  - IR bit ← 1: No interrupt request

- **Receive error?**
  - No error
  - Read received data
  - See Note 2
  - Disable PMC0 operation (3)
  - Disable timer B2 count
  - Set initial value
  - Start timer B2 count
  - return

**Notes:**
1. _remote_control_0 receives the header, data 0, and data 1.
2. Add communication error processing if necessary.
3. When the EN bit is set to 0 (operation disabled), the PMCi circuit stops operating and the ENFLG bit becomes 0 (operation stopped). After setting the EN bit to 0, it takes up to one cycle of the count source before the ENFLG bit becomes 0.

---
No overflow

Overflow ?

Overflow

Enable PMC0 operation (2)

EN bit ← 1: Enable PMC0 operation

Wait for PMC0 to start

Read the PMC0CON2 register.
ENFLG bit = 0: PMC0 stops
= 1: PMC0 operating

Disable timer B2 counting

TB2S bit ← 0: Stop counting

Write to TB2MR register

TB2MR &< 0FFh: The MR3 bit becomes 0
(no overflow) when writing to the TB2MR
register.

The range of 0C6Ah to 0F2Ch includes
the pulse period of special data of pattern
A.

Disable timer B2 counting

TB2S bit ← 0: Stop counting

Set initialization value

TBS register ← 5000h: Set initialization
value

Start timer B2 count

TB2S bit ← 1: Start counting

return

Notes:
1. _timer_b2() receives the repeat code.
2. The PMC0 circuit starts operating by setting the EN bit to 1 (operation enabled)
   and the ENFLG bit becomes 1 (operating). After setting the EN bit to 1, it takes
   up to two cycles of the count source before the ENFLG bit becomes 1. During
   this period, do not access bits or registers associated with the PMC0 circuit
   except for the ENFLG bit.

Figure 5.9 Timer B2 Interrupt Handling
5.7.5 PMC1 Interrupt Handling

Figure 5.10 shows the interrupt handling of PMC1.

```
Figure 5.10  PMC1 Interrupt Handling
```

- **remote_control_1**
- **global variable**
  - `bool PMC1_ERFLG = 0: PMC1 error flag`
  - `char PMC1_data[]: Store the data received on PMC1.`
  - `int odd_even_bit = 0: Received data is valid when even, and invalid when odd.`
  - `int bits = 0: Each time data is received, store it in PMC1_data[].`

- **local variable**
  - `int iNumber: Group of bits currently received`
  - `int iOffset: Positional location of the group of bits currently received`
  - `BitBuffer: Value of the bits currently received`
  - `int m: Use this variable to clear the PMC1_data[] in order.```

1. Reception error? 
   - Error
   - No error

2. Data reception completed? 
   - Yes
   - No

3. Header pattern matched? 
   - Yes
   - No

```
odd_even_bit++
odd_even_bit = odd? 
Process the received data
return
```

```
1
Disabling error interrupt
Disable receive error interrupt
```

```
1. Disable receive error interrupt
   Disable data 0/data 1 match interrupt
```

```
If an error occurs, set PMC1_ERFLG to 1.
```

```
2
PMCC1_ERFLG = 0 ?
```

```
If an error does not occur, encode the PMC1_data[] data.
```

```
3
Enable data 0/data 1 error interrupt
Enable receiving error interrupt
```

```
When the header is received, enable the data 0 match interrupt, data 1 match interrupt, and receive error interrupt.
```

```
Clear the buffer
```

```
odd_even_bit = 0
```

```
PMC1_ERFLG = 0
Clear PMC1_data[]
```

```
return
```

Note:
1. Add communication error processing where necessary.
6. Sample Code
Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents
M16C/63 Group User’s Manual: Hardware Rev.1.00
M16C/64A Group User’s Manual: Hardware Rev.1.10
M16C/64C Group User’s Manual: Hardware Rev.1.00
M16C/65 Group User’s Manual: Hardware Rev.1.10
M16C/65C Group User’s Manual: Hardware Rev.1.00
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 and R8C Family C Compiler Package V.5.45
C Compiler User’s Manual Rev.3.00
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
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Mar. 31, 2011</td>
<td>First edition issued</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 type number, confirm that the change will not lead to problems.
   - The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type 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 by 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 depends 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. 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 product 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others. You should not disclose by Renesas Electronics such as that disclosed through our website.

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.