RL78/G23
32-Bit Interval Timer (8-bit counter mode)

Introduction

This application note describes how to use the 32-bit interval timer channels in 8-bit counter mode. Using this mode, the application reverses LED indications based on the settings of the compare match detection flags when a timer interrupt occurs. The application also changes the timer interrupt generation intervals based on the number of times the switch has been previously pressed.

Target Device

RL78/G23

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.
Contents

1. Specifications .....................................................................................................................................3
   1.1 Overview of Specifications ........................................................................................................3
   1.2 Outline of Operation ...............................................................................................................6

2. Operation Confirmation Conditions ..............................................................................................7

3. Hardware Descriptions ....................................................................................................................8
   3.1 Example of Hardware Configuration .......................................................................................8
   3.2 List of Pins to be Used ............................................................................................................8

4. Software Explanation .......................................................................................................................9
   4.1 Setting of Option Byte ..........................................................................................................9
   4.2 List of Constants ..................................................................................................................9
   4.3 List of Variables ..................................................................................................................10
   4.4 List of Functions ................................................................................................................10
   4.5 Specification of Functions ...................................................................................................10
   4.6 Flowcharts ...........................................................................................................................12
      4.6.1 Initial Settings by the User in Main Processing .................................................................13
      4.6.2 Interrupt Processing for Interval Timer Channel 0 .........................................................14
      4.6.3 Interrupt Processing for Interval Timer Channel 1 .........................................................15
      4.6.4 Interrupt Processing for Interval Timer Channel 2 .........................................................16
      4.6.5 Interrupt Processing for Interval Timer Channel 3 .........................................................17
      4.6.6 External Interrupt (INTP0) Processing ...........................................................................18

5. Sample Code ......................................................................................................................................19

6. Reference Documents ...................................................................................................................19

All trademarks and registered trademarks are the property of their respective owners. ..........19
1. Specifications

1.1 Overview of Specifications

In this application note, the application uses timer interrupts generated by the 32-bit interval timer (INTITL) and interrupts received by the edge-detecting interrupt input pin (INTP0).

When an INTITL timer interrupt occurs, the application checks the compare match detection flags (ITF00 to ITF03) and reverses the indications of the LEDs associated with the flags. In addition, the application changes the frequency division ratios for the timer channels to modify the timer interrupt generation intervals based on the number of times the switch (SW) has been previously pressed.

Table 1-1 lists the peripheral functions to be used and their uses, Figure 1-1 gives an overview of the relationship between the timer and interrupts, and Figure 1-2 gives an overview of the timer interrupt operation.

Table 1-1 Peripheral Functions Used and Their Uses

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-bit interval timer</td>
<td>Sets the interval of the 32-bit interval timer.</td>
</tr>
<tr>
<td>External interrupt</td>
<td>Used as a interrupt generated on pin input edge detection (INTP0) by switch input.</td>
</tr>
<tr>
<td></td>
<td>Receives switch input interrupts on the edge-detecting interrupt input pin (INTP0).</td>
</tr>
<tr>
<td>Port output</td>
<td>Controls the LEDs (LED1 to LED4) connected to P03, P02, P43, and P42 pins.</td>
</tr>
</tbody>
</table>
Figure 1-1 Overview of the Relationship Between the Timer and Interrupts (Operation Example of Channel 0 in 8-bit Counter Mode)

Caution: This timing chart is for channel 0. For channels 1 to 3, check their setting values and replace the names of the relevant registers with the appropriate ones.
Figure 1-2  Overview of the Timer Interrupt Operation (8-bit Counter Mode)

Caution  This timing chart is for channels 0 and 1. For channels 2 and 3, check their setting values and replace the names of the relevant registers with the appropriate ones.
1.2 Outline of Operation

This section describes how to set the 8-bit counter mode for the 32-bit interval timer.

After the mode is set, the timer interrupt (INTITL) processing for the interval timer counts the number of compare matches for each timer channel. When any of the counts reaches 2000, the CPU reverses the corresponding LED indication. Another processing changes the frequency division ratios for the timer channels, as shown in Table 1-2, based on the number of times the switch has been previously pressed (the operation number increments as the number of switch presses increases).

Table 1-2  LED On/Off Interval Changes

<table>
<thead>
<tr>
<th>Operation</th>
<th>Frequency division ratios for timer channels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Channel 0 (P03)</td>
</tr>
<tr>
<td>(1)</td>
<td>ITLO</td>
</tr>
<tr>
<td>(2)</td>
<td>ITLO/2</td>
</tr>
<tr>
<td>(3)</td>
<td>ITLO/4</td>
</tr>
<tr>
<td>(4)</td>
<td>ITLO/8</td>
</tr>
<tr>
<td>(5)</td>
<td>ITLO/16</td>
</tr>
<tr>
<td>(6)</td>
<td>ITLO/32</td>
</tr>
<tr>
<td>(7)</td>
<td>ITLO/64</td>
</tr>
<tr>
<td>(8)</td>
<td>ITLO/128</td>
</tr>
</tbody>
</table>

Operations (1) to (8) cycle each time the switch is pressed.

(1) Initialize the I/O ports.
   ● P03, P02, P43, and P42 pins: Set as output ports (to be used for LED on/off control).
   ● P137 / INTP0 pin: Set as an input port (to be used for switch input).

(2) Initialize the 32-bit interval timer.
   ● Place the timer in 8-bit counter mode.
   ● Set the compare values (INTCMP0n) for channels 0 to 4.
     ITLCMP000 = 3FH
     ITLCMP001 = 7FH
     ITLCMP012 = BFH
     ITLCMP013 = FFH
   ● Enable timer interrupts by the 32-bit interval timer (INTITL).

(3) Initialize the edge-detecting external interrupt pin.
   ● Set the falling edge as the valid edge for the INTP0 pin.
   ● Enable INTP0 interrupts.

(4) Execute a HALT instruction and wait for a timer interrupt (INTITL).

(5) When a timer interrupt (INTITL) cancels HALT mode, the number of compare matches (the number of interrupts) for each timer channel is counted.

(6) When any of the timer interrupt counts reaches 2000, the indication of the applicable LED is reversed.

(7) The INTP0 interrupt processing increments the number of switch inputs (the number of INTP0 interrupts) and changes the frequency division ratio for each timer channel as shown in Table 1-2.
2. Operation Confirmation Conditions

The operation of the sample code provided with this application note has been tested under the following conditions.

Table 2-1 Operation Confirmation Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>RL78/G23 (R7F100GLG)</td>
</tr>
<tr>
<td>Board used</td>
<td>RL78/G23-64p Fast Prototyping Board (RTK7RLG230CLG000BJ)</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>High-speed on-chip oscillator clock (fIH): 32 MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5.0 V (can be operated at 2.0 V to 5.5 V)</td>
</tr>
<tr>
<td></td>
<td>LVD0 operations (VLVD0): Reset mode</td>
</tr>
<tr>
<td></td>
<td>At rising edge TYP. 1.90V (1.84 V to 1.95 V)</td>
</tr>
<tr>
<td></td>
<td>At falling edge TYP. 1.86V (1.80 V to 1.91 V)</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>CS+ for CC E8.05.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>(CS+)</td>
<td></td>
</tr>
<tr>
<td>C compiler (CS+)</td>
<td>CC-RL V1.10.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>e2studio V2021-04 (21.4.0) from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>(e2studio)</td>
<td></td>
</tr>
<tr>
<td>C compiler (e2studio)</td>
<td>CC-RL V1.10.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>IAR Embedded Workbench for Renesas RL78 V4.21.1 from IAR Systems Corp.</td>
</tr>
<tr>
<td>(IAR)</td>
<td></td>
</tr>
<tr>
<td>C compiler (IAR)</td>
<td>IAR C/C++ Compiler for Renesas RL78 V4.21.1.2260 from IAR Systems Corp.</td>
</tr>
<tr>
<td>Smart configurator (SC)</td>
<td>V1.0.1 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Board support package (BSP)</td>
<td>V1.00 from Renesas Electronics Corp.</td>
</tr>
</tbody>
</table>
3. Hardware Descriptions

3.1 Example of Hardware Configuration

Figure 3-1 shows an example of the hardware configuration used in the application note.

![Figure 3-1 Hardware Configuration](image)

Note 1. This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes appropriate pin handling and meets electrical characteristic requirements (connect each input-only port to VDD or VSS through a resistor).

Note 2. Connect any pins whose name begins with EVSS to VSS, and any pins whose name begins with EVDD to VDD, respectively.

Note 3. VDD must not be lower than the reset release voltage (VLVD0) that is specified for the LVD0.

3.2 List of Pins to be Used

Table 3-1 lists the pins to be used and their functions.

<table>
<thead>
<tr>
<th>Pin name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P03</td>
<td>Output</td>
<td>LED1 control</td>
</tr>
<tr>
<td>P02</td>
<td>Output</td>
<td>LED2 control</td>
</tr>
<tr>
<td>P43</td>
<td>Output</td>
<td>LED3 control</td>
</tr>
<tr>
<td>P42</td>
<td>Output</td>
<td>LED4 control</td>
</tr>
<tr>
<td>P137 / INTP0</td>
<td>Input</td>
<td>Input pin for the switch (SW) (external interrupt request input pin)</td>
</tr>
</tbody>
</table>

Caution In this application note, only the used pins are processed. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.
4. Software Explanation

4.1 Setting of Option Byte

Table 4-1 shows the option byte settings.

<table>
<thead>
<tr>
<th>Address</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>000C0H / 040C0H</td>
<td>1110111B</td>
<td>Disables the watchdog timer. (Counting stopped after reset)</td>
</tr>
<tr>
<td>000C1H / 040C1H</td>
<td>11111110B</td>
<td>LVD0 detection voltage: reset mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>At rising edge TYP. 1.90V (1.84 V to 1.95 V)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>At falling edge TYP. 1.86V (1.80 V to 1.91 V)</td>
</tr>
<tr>
<td>000C2H / 040C2H</td>
<td>11101000B</td>
<td>HS mode, High-speed on-chip oscillator clock (fIH): 32 MHz</td>
</tr>
<tr>
<td>000C3H / 040C3H</td>
<td>10000100B</td>
<td>Enables on-chip debugging</td>
</tr>
</tbody>
</table>

4.2 List of Constants

Table 4-2 lists the constants that are used in the sample code.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT_COUNT</td>
<td>2000</td>
<td>Timer interrupt count for reversing LED indications</td>
</tr>
<tr>
<td>WAITCOUNT_32M</td>
<td>8000</td>
<td>Wait count for 5 ms when the MCU operates in HS mode at 32 MHz</td>
</tr>
</tbody>
</table>

4.3 List of Variables

Table 4-3 lists global variables.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Description</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint16_t</td>
<td>g_transition_status</td>
<td>Transition status variable</td>
<td>r_Config_INTC_intp0_interrupt</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_itldiv_table</td>
<td>Frequency division ratio table for each timer counter</td>
<td>r_Config_INTC_intp0_interrupt</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_inttm00_counter</td>
<td>Timer interrupt counter for channel 0</td>
<td>R_Config_ITL000_Callback_Shared_Interrupt</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_inttm01_counter</td>
<td>Timer interrupt counter for channel 1</td>
<td>R_Config_ITL000_Callback_Shared_Interrupt</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_inttm02_counter</td>
<td>Timer interrupt counter for channel 2</td>
<td>R_Config_ITL000_Callback_Shared_Interrupt</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_inttm03_counter</td>
<td>Timer interrupt counter for channel 3</td>
<td>R_Config_ITL000_Callback_Shared_Interrupt</td>
</tr>
</tbody>
</table>
4.4 List of Functions

Table 4-4 shows a list of functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserInit()</td>
<td>Initial settings by the user in main processing</td>
</tr>
<tr>
<td>R_Config_ITL000_Callback_Shared_Interrupt()</td>
<td>Interrupt processing for interval timer channel 0</td>
</tr>
<tr>
<td>R_Config_ITL001_Callback_Shared_Interrupt()</td>
<td>Interrupt processing for interval timer channel 1</td>
</tr>
<tr>
<td>R_Config_ITL012_Callback_Shared_Interrupt()</td>
<td>Interrupt processing for interval timer channel 2</td>
</tr>
<tr>
<td>R_Config_ITL013_Callback_Shared_Interrupt()</td>
<td>Interrupt processing for interval timer channel 3</td>
</tr>
</tbody>
</table>

4.5 Specification of Functions

The function specifications of the sample code are shown below.

**UserInit()**

Outline: Initial settings by the user in main processing

Header: r_cg_macrodriver.h, r_cg_userdefine.h, r_cg_itl_common.h, Config_INTC.h, Config_ITL000.h, Config_ITL001.h, Config_ITL012.h, Config_ITL013.h

Declaration: void UserInit (void)

Description: Initialization required for running the application

Argument: None

Return Value: None

**R_Config_ITL000_Callback_Shared_Interrupt()**

Outline: Interrupt processing for interval timer channel 0

Header: r_cg_macrodriver.h, r_cg_userdefine.h, Config_ITL000.h

Declaration: void R_Config_ITL000_Callback_Shared_Interrupt (void)

Description: Reverses the LED1 indication each time the number of channel 0 interrupts reaches 2000.

Argument: None

Return Value: None

**R_Config_ITL001_Callback_Shared_Interrupt()**

Outline: Interrupt processing for interval timer channel 1

Header: r_cg_macrodriver.h, r_cg_userdefine.h, Config_ITL001.h

Declaration: void R_Config_ITL001_Callback_Shared_Interrupt (void)

Description: Reverses the LED2 indication each time the number of channel 1 interrupts reaches 2000.

Argument: None

Return Value: None
### R_Config_ITL012_Callback_Shared_Interrupt()

**Outline**  
Interrupt processing for interval timer channel 2

**Header**  
r_cg_macrodriver.h, r_cg_userdefine.h, Config_ITL012.h

**Declaration**  
void R_Config_ITL012_Callback_Shared_Interrupt (void)

**Description**  
Reverses the LED3 indication each time the number of channel 2 interrupts reaches 2000.

**Argument**  
None

**Return Value**  
None

---

### R_Config_ITL013_Callback_Shared_Interrupt()

**Outline**  
Interrupt processing for interval timer channel 3

**Header**  
r_cg_macrodriver.h, r_cg_userdefine.h, Config_ITL013.h

**Declaration**  
void R_Config_ITL013_Callback_Shared_Interrupt (void)

**Description**  
Reverses the LED4 indication each time the number of channel 3 interrupts reaches 2000.

**Argument**  
None

**Return Value**  
None

---

### r_Config_INTC_intp0_interrupt()

**Outline**  
External interrupt processing

**Header**  
r_cg_macrodriver.h, r_cg_userdefine.h, Config_INTC.h, r_cg_itl.h,  
r_cg_itl_common.h, Config_ITL000h, Config_ITL001.h, Config_ITL012.h,  
Config_ITL013.h

**Declaration**  
static void __near r_Config_INTC_intp0_interrupt (void)

**Description**  
Changes the frequency division ratio for each interval timer channel when the switch is pressed.

**Argument**  
None

**Return Value**  
None
4.6 Flowcharts

Figure 4-1 shows the flowchart of the main processing in this application note.

Figure 4-1 Main Processing

```
main()

Main initial settings
UserInit()

: while(1) loop

HALT

External interrupt (INTP0)
Interval time interrupt (INTITL)
```
4.6.1 Initial Settings by the User in Main Processing

Figure 4-2 shows the flowchart of the initial settings by the user in main processing.

Figure 4-2  Initial Settings by the User in Main Processing

- **UserInit()**
  - INTP0 interrupt enable processing
    - R_Config_INTC_INTP0_Start()
  - 8-bit counter mode
    - Channels 0 to 3 start operation
  - 32-bit interval timer interrupt enable processing
    - R_ITL_Start_Interrupt()
  - return

- PIF0 bit ← 0
- PMK0 bit ← 0
- ITLS0 register
  - ITF03 to ITF00 bits ← 0
- ITLMKF0 register
  - MKF03 to MKF00 bits ← 0
- ITLCTL0 register
  - ITLEN03 to ITLEN00 bits ← 1
- ITLIF0 bit ← 0
- ITLMK0 bit ← 0
4.6.2 Interrupt Processing for Interval Timer Channel 0

Figure 4-3 shows the flowchart of the interrupt processing for interval timer channel 0.

Figure 4-3 Interrupt Processing for Interval Timer Channel 0

- R_Config_ITL000_Callback_Shared_Interrupt()
- Clear the compare match detection flag for channel 0
- Increment the g_inttm00_counter interrupt counter for the interval timer (channel 0)
- g_inttm00_counter == 500?
  - YES: P03 (LED1): Reverse output
  - NO: ITLS0 register ITF00 bit ← 0
- g_inttm00_counter = 0
- return
4.6.3 Interrupt Processing for Interval Timer Channel 1

Figure 4-4 shows the flowchart of the interrupt processing for interval timer channel 1.

Figure 4-4  Interrupt Processing for Interval Timer Channel 1

```
R_Config_ITL001_Callback_Shared_Interrupt()

Clear the compare match detection flag for channel 1
ITLS0 register
ITF01 bit ← 0

Increment the g_inttm01_counter interrupt counter for the interval timer (channel 1)

g_inttm01_counter == 500 ?

YES

P02 (LED2): Reverse output

NO

g_inttm01_counter = 0

return
```
4.6.4 Interrupt Processing for Interval Timer Channel 2

Figure 4-5 shows the flowchart of the interrupt processing for interval timer channel 2.

Figure 4-5 Interrupt Processing for Interval Timer Channel 2

```
R_Config_ITL012_Callback_Shared_Interrupt()

Clear the compare match detection flag for channel 2
ITLS0 register ITF02 bit ← 0

Increment the g_inttm02_counter interrupt counter for the interval timer (channel 2)

g_inttm02_counter == 500 ?
YES

P43 (LED3): Reverse output

NO

g_inttm02_counter = 0

return
```
4.6.5 Interrupt Processing for Interval Timer Channel 3

Figure 4-6 shows the flowchart of the interrupt processing for interval timer channel 3.

Figure 4-6  Interrupt Processing for Interval Timer Channel 3

```
R_Config_ITL013_Callback_Shared_Interrupt()

Clear the compare match detection flag for channel 3
ITLS0 register
ITF03 bit ← 0

Increment the g_inttm03_counter interrupt counter for the interval timer (channel 3)

g_inttm03_counter == 500 ?

YES

P43 (LED4): Reverse output

NO

g_inttm03_counter = 0

return
```
4.6.6 External Interrupt (INTP0) Processing

Figure 4-7 shows the flowchart of the external interrupt (INTP0) processing.

Figure 4-7 External Interrupt (INTP0) Processing

```plaintext
r_Config_INTC_intp0_interrupt()

Chattering protection processing

Increment the SW input count
g_transition_status = 0→1→2→⋯→7→0

8-bit counter mode
Channels 0 to 3 stop operation

32-bit interval timer interrupt disable processing
R_ITL_Stop_Interrupt()

Change frequency division ratio for generating timer interrupts (channel 0)

Change frequency division ratio for generating timer interrupts (channel 1)

Change frequency division ratio for generating timer interrupts (channel 2)

Change frequency division ratio for generating timer interrupts (channel 3)

8-bit counter mode
Channels 0 to 3 start operation

32-bit interval timer interrupt enable processing
R_ITL_Start_Interrupt()

return

Go to return when the interrupt is caused by chattering

ITLMKF0 register
MKF03 to MKF00 bits ← 1

ITLS0 register
ITF03 to ITF00 bits ← 0
ITLCTL0 register
ITLEN03 to ITLEN00 bits ← 0

ITLFDIV00 register
FDIV002 to FDIV000 bits: A value is set from the frequency division ratio table

ITLFDIV00 register
FDIV012 to FDIV010 bits: A value is set from the frequency division ratio table

ITLFDIV01 register
FDIV022 to FDIV020 bits: A value is set from the frequency division ratio table

ITLFDIV01 register
FDIV032 to FDIV030 bits: A value is set from the frequency division ratio table

ITLS0 register
ITF03 to ITF00 bits ← 0
ITLMKF0 register
MKF03 to MKF00 bits ← 0
ITLCTL0 register
ITLEN03 to ITLEN00 bits ← 1
```
5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

RL78/G23 User’s Manual: Hardware (R01UH0896J)
RL78 family user’s manual software (R01US0015J)
The latest versions can be downloaded from the Renesas Electronics website.

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

All trademarks and registered trademarks are the property of their respective owners.
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>2021.04.13</td>
<td>—</td>
<td>First Edition</td>
<td></td>
</tr>
<tr>
<td>1.01</td>
<td>2021.07.12</td>
<td>7</td>
<td>Updated the Operation Confirmation Conditions</td>
<td></td>
</tr>
<tr>
<td>1.02</td>
<td>2021.08.18</td>
<td>—</td>
<td>Fix the sample code</td>
<td></td>
</tr>
</tbody>
</table>
General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

   Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

   Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between \( V_{IL} \) (Max.) and \( V_{IH} \) (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between \( V_{IL} \) (Max.) and \( V_{IH} \) (Min.).

7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.
RL78/G23

32-Bit Interval Timer (8-bit counter mode)

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

6. Renesas Electronics products are classified according to the following reliability grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

- **“Standard”:** Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment, industrial robots, etc.
- **“High Quality”:** Automobiles, traffic control (traffic lights), large-scale communication equipment, key financial terminal systems, safety control equipment, etc.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to, any unauthorized access or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. Renesas Electronics does not warrant or guarantee that Renesas Electronics products will be invulnerable or free from corruption, attack, viruses, interference, hacking, data loss or theft, or other security intrusions (“Vulnerability Issues”). Renesas Electronics disclaims any and all responsibility or liability arising from or related to any vulnerability issues. Furthermore, to the extent permitted by applicable law, Renesas Electronics disclaims any and all warranties, express or implied, with respect to this document and any related or accompanying software or hardware, including but not limited to, the implied warranties of merchantability, or fitness for a particular purpose.

8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

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

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

(Rev. 5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

www.renesas.com/contact/

Trademarks

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

© 2021 Renesas Electronics Corporation. All rights reserved.