Introduction

This application note describes how the trash can with automatic operating lid is implemented by using the infrared LED, infrared receiver, and a motor.

This application uses the RL78/G10 timer array unit, 12-bit interval timer, and external interrupts.

Target Device

RL78/G10

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.
1. Specifications

This application uses the RL78/G10 timer array unit, 12-bit interval timer, and external interrupts. The system initially controls the motor to close the lid; turns on the LED (CLOSE) for one second; and executes standby mode. After that, the system branches either to standby mode, manual mode, or automatic mode depending on the operation state.

- Standby mode

In this mode, the system performs the infrared detection process every 200 ms. If the system has detected infrared light, the system branches to automatic mode. If the system has detected no infrared light, the system turns on the LED (CLOSE) every four seconds for 5 ms.

When the switch (OPEN) is pressed, the system controls the motor to open the lid; turns on the LED (OPEN) for one second; and shifts to manual mode.

- Manual mode

In this mode, the system turns on the LED (OPEN) every four seconds for 5 ms.

When the switch (CLOSE) is pressed, the system controls the motor to close the lid; turns on the LED (CLOSE) for one second; and shifts to standby mode.

- Automatic mode

In this mode, the system initially controls the motor to open the lid, and turns on the LED (OPEN). The system then performs the infrared detection process every 200 ms. If the system has detected no infrared light for five seconds, the system controls the motor to close the lid; turns off the LED (OPEN); and shifts to standby mode.

When the switch (OPEN) is pressed, the system turns off the LED (OPEN) and shifts to manual mode. When the switch (CLOSE) is pressed, the system controls the motor to close the lid; turns on the LED (CLOSE) for one second; and shifts to standby mode.

For motor control, the port outputs and timer array units are used to provide PWM output. Similarly, in the infrared detection process, the timer array units are used to output 25.6-us rectangular waves to the infrared LED.

Table 1.1 shows the required peripheral functions and their uses. Figure 1.1 shows the outline of the operations.

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer array unit channel 0</td>
<td>Used for PWM output of motor control signal 2.</td>
</tr>
<tr>
<td>Timer array unit channel 1</td>
<td>Used for PWM output of motor control signal 2.</td>
</tr>
<tr>
<td>Timer array unit channel 2</td>
<td>Counts the output period for the infrared LED and others.</td>
</tr>
<tr>
<td>Timer array unit channel 3</td>
<td>Counts the output width of rectangular wave for the infrared LED.</td>
</tr>
<tr>
<td>12-bit interval timer</td>
<td>Counts 200 ms.</td>
</tr>
<tr>
<td>P137/INTP0</td>
<td>Switch input to close the lid</td>
</tr>
<tr>
<td>P41/INTP2</td>
<td>Switch input to open the lid</td>
</tr>
<tr>
<td>P00</td>
<td>Output port for LED (CLOSE) display</td>
</tr>
<tr>
<td>P01</td>
<td>Output port for LED (OPEN) display</td>
</tr>
<tr>
<td>P02</td>
<td>Turns on/off the infrared receiver.</td>
</tr>
<tr>
<td>P03</td>
<td>Inputs the signal from the infrared receiver.</td>
</tr>
<tr>
<td>P04/TO01</td>
<td>Outputs motor control signal 2.</td>
</tr>
<tr>
<td>P05</td>
<td>Outputs motor control signal 1.</td>
</tr>
<tr>
<td>P07/TO03</td>
<td>Outputs the signal to the infrared LED.</td>
</tr>
</tbody>
</table>
Initial setting

**Initial operations**
- Closes the lid.
- Turns on LED (CLOSE) for 1 s.

**Standby mode**
- Performs the infrared detection process every 200 ms.
- Turns on LED (CLOSE) every 4 s for 5 ms.

**Infrared light detected**
- Opens the lid.
- Turns on LED (OPEN).

**Switch (CLOSE) pressed down**
- Closes the lid.
- Turns on LED (CLOSE) for 1 s.

**Switch (OPEN) pressed down**
- Opens the lid.
- Turns on LED (OPEN) for 1 s.

**Automatic mode**
- Performs the infrared detection process every 200 ms.
- Always turns on LED (OPEN).

**Manual mode**
- Turns on LED (OPEN) every 4 s for 5 ms.

**No infrared light detected for 5 s**
- Closes the lid.
- Turns off LED (OPEN).

**Switch (CLOSE) pressed down**
- Closes the lid.
- Turns on LED (CLOSE) for 1 s.

**Switch (OPEN) pressed down**
- Closes the lid.
- Turns on LED (OPEN).

**Infrared light detected**
- Opens the lid.
- Turns on LED (OPEN).

**Manual mode**
- Turns on LED (OPEN) every 4 s for 5 ms.

Figure 1.1  Overview of the Operation
1.1.1 Infrared Detection

With the application described in this note, the rectangular wave output function of the timer array unit is used to output the approximately 38-kHz rectangular wave to suit to the 38-kHz center frequency of the infrared receiver.

When the infrared receiver receives 38-kHz infrared light, it sets the output signal to the low level. This signal is input to P03 and is used to determine whether infrared light has been detected.

![Infrared Detection Diagram]

Figure 1.2 Infrared Detection

1.1.2 Motor Control

Figure 1.3 shows the motor control signal for closing the lid, and figure 1.4 shows the motor control signal for opening the lid. The motor control signals in the figure are assumed to be output to the motor driver that involves the operations listed in table 1.2. Motor control signal 1 is output from the port, and motor control signal 2 is output by using the PWM output function of the timer array unit. The duty cycle of the PWM output of motor control signal 2 is changed by changing the setting values of TDR01H and TDR01L registers after generation of the interrupt request of the timer array unit that is currently performing the motor control process.
Figure 1.3  Motor Control to Close the Lid

Figure 1.4  Motor Control to Open the Lid

Table 1.2  Operation of motor driver

<table>
<thead>
<tr>
<th>Operation</th>
<th>IN1</th>
<th>IN2</th>
<th>OUT1</th>
<th>OUT2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>0</td>
<td>0</td>
<td>Hi-Z</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>Forward</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Backward</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Brake</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
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>RL78/G10 (R5F10Y47Y)</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td>- High-speed on-chip oscillator clock: 1.25MHz</td>
</tr>
<tr>
<td></td>
<td>- CPU/peripheral hardware clock: 1.25MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>3.6V (operating range 2.2V to 5.5V)</td>
</tr>
<tr>
<td></td>
<td>SPOR detection voltage: Falling edge: 2.11V</td>
</tr>
<tr>
<td></td>
<td>Rising edge: 2.16V</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>CS+ for CC V6.00.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>(CS+)</td>
<td></td>
</tr>
<tr>
<td>C compiler (CS+)</td>
<td>CC-RL V1.05.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td></td>
</tr>
<tr>
<td>(e² studio)</td>
<td>e² studio V5.4.0.018 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>C compiler (e² studio)</td>
<td>CC-RL V1.05.00 from Renesas Electronics Corp.</td>
</tr>
</tbody>
</table>

3. Related Application Notes

The application notes that are related to this application note are listed below for reference.

RL78/G10 Initialization CC-RL (R01AN2668J) Application Note
RL78/G10 Timer Array Unit (Interval Timer) CC-RL (R01AN3839J) Application Note
RL78/G10 Timer Array Unit (PWM Output) CC-RL (R01AN2667J) Application Note
4. Hardware Explanation

4.1 Hardware Configuration Example

Figure 4.1 shows an example of the hardware configuration used in this application note.

![Figure 4.1 Hardware Configuration](image)

Cautions: 1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to VDD or VSS via a resistor).

2. VDD must supply not lower than the reset release voltage (VSPOR) that is specified as SPOR.

4.2 Used Pin List

Table 4.1 provides List of Pins and Functions.

<table>
<thead>
<tr>
<th>Pin/ Name</th>
<th>Input/Output</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P00</td>
<td>Output</td>
<td>Controls LED (CLOSE).</td>
</tr>
<tr>
<td>P01</td>
<td>Output</td>
<td>Controls LED (OPEN).</td>
</tr>
<tr>
<td>P02</td>
<td>Output</td>
<td>Turns on/off the infrared receiver.</td>
</tr>
<tr>
<td>P03</td>
<td>Input</td>
<td>Inputs the signal from the infrared receiver.</td>
</tr>
<tr>
<td>P04/TO01</td>
<td>Output</td>
<td>Motor control signal 2</td>
</tr>
<tr>
<td>P05</td>
<td>Output</td>
<td>Motor control signal 1</td>
</tr>
<tr>
<td>P07</td>
<td>Output</td>
<td>Controls the infrared LED.</td>
</tr>
<tr>
<td>P41/INTP2</td>
<td>Input</td>
<td>Switch input (to close the lid).</td>
</tr>
<tr>
<td>P137/INTP0</td>
<td>Input</td>
<td>Switch input (to open the lid).</td>
</tr>
</tbody>
</table>
5. Software Explanation

5.1 Operation Outline

With this application, after making the initial settings (timer array unit, 12-bit interval timer, and external interrupts), the system initially controls the motor to close the lid; turns on the LED (CLOSE) for one second; and shifts to STOP mode.

The system returns from STOP mode by an external interrupt or a 12-bit interval timer interrupt, which is generated every 200 ms. After returning from STOP mode, the system executes standby mode. After that, the system branches either to standby mode, manual mode, or automatic mode depending on the operation state.

- Standby mode (operation status: STANDBY)
  After returning from STOP mode, the system performs the infrared detection process. If the system has detected infrared light, the system sets the operation status to AUTOMATIC. If the system has detected no infrared light and the 200-ms counter value is 20 (four seconds have passed), the system turns on the LED (CLOSE) for 5 ms and clears the 200-ms counter.

- Manual mode (operation status: MANUAL)
  After returning from STOP mode, the system turns on the LED (OPEN) for 5 ms and clears the 200-ms counter.

- Automatic mode (operation status: AUTOMATIC)
  After returning from STOP mode, the system performs the infrared detection process. If the system has detected infrared light, the system clears the 200-ms counter. If the system has detected no infrared light and the 200-ms counter value is 25 (five seconds have passed), the system closes the lid; turns off the LED (OPEN); clears the 200-ms counter; and sets the operation status to STANDBY.
  After shifting from standby mode, the system turns on the LED (OPEN) and opens the lid.

If the switch (OPEN) is pressed while the lid is closed, the system turns on the LED (OPEN); opens the lid; turns off the LED (OPEN); and sets the operation status to MANUAL.

If the switch (OPEN) is pressed while the lid is open, the system turns off the LED (OPEN) and sets the operation status to MANUAL.

If the switch (CLOSE) is pressed, the system turns on the LED (CLOSE); closes the lid, turns off the LED (CLOSE); and sets the operation status to STANDBY. After that, the system waits for 600 ms to prevent erroneous infrared detection.
1. Makes the initial settings for the timer array unit.
   
   <Conditions for setting channel 0>
   - Sets the timer operation mode to PWM output (master) mode.
   - Sets timer data register 00 (TDR00) to 20 ms as the initial value.
   - Sets the timer output enable register to disable operation.
   - Uses the timer channel 0 timer interrupt (INTTM00).

   <Conditions for setting channel 1>
   - Sets the timer operation mode to PWM output (slave) mode.
   - Sets timer data register 01 (TDR01) to 18 ms as the initial value.
   - Sets the timer output enable register to enable operation.
   - Does not use the timer channel 1 timer interrupt (INTTM01).

   <Conditions for setting channel 2>
   - Sets the timer operation mode to interval timer mode.
   - Sets timer data register 02 (TDR02) to 10 ms as the initial value.
   - Sets the timer output enable register to disable operation.
   - Uses the timer channel 2 timer interrupt (INTTM02).

   <Conditions for setting channel 3>
   - Sets the timer operation mode to interval timer mode.
   - Sets timer data register 03 (TDR03) to 25.6 us as the initial value.
   - Sets the timer output enable register to enable operation.
   - Does not use the timer channel 3 timer interrupt (INTTM03).

2. Makes the initial settings for the external interrupts.
   
   <Conditions for setting the external interrupts>
   - Uses the P137/INTP0 pin.
   - Sets the falling edge as the valid edge for the INTP0 pin.
   - Uses the P41/INTP2 pin.
   - Sets the falling edge as the valid edge for the INTP2 pin.

3. Makes the initial settings for the external interrupts of the 12-bit interval timer.
   
   <Conditions for setting the 12-bit interval timer>
   - Sets 0BB7H as the compare value so that a compare-match interrupt request is generated approximately every 200 ms.

4. Turns on the LED (CLOSE) after making the initial settings.
5. Controls the motor to close the lid.
6. Turns off the LED (CLOSE).
7. Starts the 12-bit interval timer operation.
8. Enables the external interrupts.
9. Enables the maskable interrupts.
10. Shifts to STOP mode.
11. Returns from STOP mode upon generation of the 12-bit interval timer interrupt after 200 ms or upon generation of the external interrupt. In the 12-bit interval timer interrupt process, increments the 200-ms counter.
12. Disables the maskable interrupts.
13. Disables the external interrupts.
14. Operates as described below according to the operation state.

<Standby mode (operation status: STANDBY)>
A) Performs the infrared detection process.
B) If having detected infrared light, sets the operation status to AUTOMATIC.
C) If having detected no infrared light and the 200-ms counter value is 20 (approximately four seconds have passed), turns on the LED (CLOSE) for 5 ms and clears the 200-ms counter.

<Manual mode (operation status: MANUAL)>
A) If the 200-ms counter value is 20 (approximately four seconds have passed), turns on the LED (OPEN) for 5 ms and clears the 200-ms counter.

<Automatic mode (operation status: AUTOMATIC)>
A) Initially turns on the LED (OPEN), controls the motor to open the lid, and clears the 200-ms counter.
B) Performs the infrared detection process.
C) If having detected infrared light, clears the 200-ms counter.
D) If having detected no infrared light and the 200-ms counter value is 25 (approximately five seconds have passed), controls the motor to close the lid; turns off the LED (OPEN); and clears the 200-ms counter.
The motor control process is described below.

1. Sets motor control signal 1. While the lid is closed, sets motor control signal 1 to the low level; and sets motor control signal 2 to the high level while the lid is open.
2. Sets the width and duty cycle of the PWM output of timer array unit channels 0 and 1.
3. Disables the 12-bit interval timer interrupt.
4. Starts the PWM output operation of timer array unit channels 0 and 1.
5. Shifts to HALT mode.
6. Returns from HALT mode by an interrupt request of timer array unit channel 0.
7. Increments the count and changes the duty cycle of the PWM output according to the count.
8. Repeats steps 5 to 7 until the count reaches 20 while the lid is closed, and until 51 while the lid is open.
9. Stops the PWM output of timer array unit channels 0 and 1.
10. Sets motor control signals 1 and 2 to the low level.

The infrared detection process is described below.

1. Sets P02 to the high level and starts power supply to the infrared receiver.
2. Starts the operation of timer array unit channel 2.
3. Shifts to HALT mode.
4. Returns from HALT mode after 5 ms by an interrupt request of timer array unit channel 2.
5. Stops the operation of timer array unit channel 2.
6. Checks the P03 input level. When low, determines that infrared light has been erroneously detected and terminates the infrared detection process.
7. Sets the interval time of timer array unit channel 2 to 620 us.
8. Starts the operation of timer array unit channels 2 and 3.
9. Shifts to HALT mode.
10. Returns from HALT mode after 620 us by an interrupt request of timer array unit channel 2.
11. Checks the P03 input level. When low, determines that infrared light has been detected.
12. When the P03 input level is low, repeats steps 8 to 10 five times, determining that infrared light has been detected. When high, terminates the infrared detection process.
13. Stops the operation of timer array unit channels 2 and 3.
14. Sets the interval time of timer array unit channel 2 to 5 ms.
15. If having detected infrared light, repeats steps 8 to 12 five times. If having detected no infrared light, terminates the process.
16. Sets P02 to the low level and stops power supply to the infrared receiver.
The operation when the switch (OPEN) is pressed is described below.

1. An external interrupt occurs.
2. Disables the external interrupts.
3. Executes either of the following operations depending on whether the lid is open or closed.

<When lid is open>
A) Turns on the LED (OPEN).
B) Controls the motor to open the lid.
C) Turns off the LED (OPEN).
D) Changes the operation status to MANUAL.
E) Clears the 200-ms counter.
F) Enables the 12-bit interval timer interrupt.
G) Shifts to HALT mode.
H) Returns from HALT mode upon generation of an interrupt request of the 12-bit interval timer.

<When lid is closed>
A) Turns off the LED.
B) Changes the operation status to MANUAL.

4. Clears the external interrupt request.

The operation when the switch (CLOSE) is pressed is described below.

1. An external interrupt occurs.
2. Disables the external interrupts.
3. Turns on the LED (CLOSE).
4. Controls the motor to close the lid.
5. Turns off the LED (CLOSE).
6. Changes the operation status to STANDBY.
7. Enables the 12-bit interval timer interrupt.
8. Shifts to HALT mode.
9. Returns from HALT mode upon generation of an interrupt request of the 12-bit interval timer.
10. Performs steps 7 and 8 three times.
11. Sets the 200-ms counter to 2.
12. Clears the external interrupt request flag.
5.2 Option Byte Settings

Table 5.1 lists the option byte settings.

<table>
<thead>
<tr>
<th>Address</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>000C0H/010C0H</td>
<td>11101111B</td>
<td>Watchdog timer operation is stopped (count is stopped after reset)</td>
</tr>
<tr>
<td>000C1H/010C1H</td>
<td>11111111B</td>
<td>SPOR detection voltage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When reset occurs: VDD &lt; 2.11 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When reset is released: VDD ≥ 2.16 V</td>
</tr>
<tr>
<td>000C2H/010C2H</td>
<td>11111011B</td>
<td>HS mode, High-speed on-chip oscillator clock: 1.25MHz</td>
</tr>
<tr>
<td>000C3H/010C3H</td>
<td>10000101B</td>
<td>On-chip debugging enabled</td>
</tr>
</tbody>
</table>

5.3 Constants

Table 5.2 lists the constants that are used in this sample program.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED_OPEN_ON</td>
<td>02H</td>
<td>P0 register setting value (to turn on LED (OPEN))</td>
</tr>
<tr>
<td>LED_CLOSE_ON</td>
<td>01H</td>
<td>P0 register setting value (to turn on LED (CLOSE))</td>
</tr>
<tr>
<td>LED_OFF</td>
<td>FCH</td>
<td>P0 register setting value (to turn off LED)</td>
</tr>
<tr>
<td>IR_RECEIVER_ON</td>
<td>04H</td>
<td>P0 register setting value (to start power supply to the infrared receiver)</td>
</tr>
<tr>
<td>IR_RECEIVER_OFF</td>
<td>FBH</td>
<td>P0 register setting value (to stop power supply to the infrared receiver)</td>
</tr>
<tr>
<td>MOTOR_SIGNAL1_LOW</td>
<td>DFH</td>
<td>P0 register setting value (to output low-level motor control signal 1)</td>
</tr>
<tr>
<td>MOTOR_SIGNAL1_HIGH</td>
<td>20H</td>
<td>P0 register setting value (to output high-level motor control signal 1)</td>
</tr>
<tr>
<td>MOTOR_STOP</td>
<td>CFH</td>
<td>P0 register setting value (to output low-level motor control signals 1 and 2)</td>
</tr>
<tr>
<td>STANDBY</td>
<td>0</td>
<td>Operation status setting value (standby mode)</td>
</tr>
<tr>
<td>AUTOMATIC</td>
<td>1</td>
<td>Operation status setting value (automatic mode)</td>
</tr>
<tr>
<td>MANUAL</td>
<td>2</td>
<td>Operation status setting value (manual mode)</td>
</tr>
<tr>
<td>DETECTED</td>
<td>0</td>
<td>Value used for determining infrared detection</td>
</tr>
<tr>
<td>START</td>
<td>0</td>
<td>Infrared detection process status</td>
</tr>
<tr>
<td>END</td>
<td>1</td>
<td>Infrared detection process status</td>
</tr>
<tr>
<td>g_pwm_duty_h[20]</td>
<td>00H, 00H, 00H, 00H, 00H, 00H, 00H, 13H, 21H, 32H, 42H, 52H, 61H, 6FH, 7EH, 8EH, 9EH, ADH, 00H, 00H, 00H</td>
<td>TDR01H register setting value Sets PWM output width of motor control signal 2.</td>
</tr>
<tr>
<td>g_pwm_duty_l[20]</td>
<td>00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 00H, 87H, 33H, C7H, 67H, 07H, A7H, 53H, F3H, 93H, 33H, D3H, 00H, 00H, 00H</td>
<td>TDR01L register setting value Sets PWM output width of motor control signal 2.</td>
</tr>
</tbody>
</table>
## 5.4 Variables

Table 5.3 lists the variables.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint8_t</td>
<td>g_operation_status</td>
<td>Operation status</td>
<td>main(), r_main_led_light(), r_intc0_interrupt(), r_intc2_interrupt()</td>
</tr>
<tr>
<td>uint8_t</td>
<td>g_open_flag</td>
<td>Open/close flag</td>
<td>main(), R_MAIN_UserInit(), r_main_motor_control(), r_intc0_interrupt(), r_intc2_interrupt()</td>
</tr>
<tr>
<td>uint8_t</td>
<td>g_ir_flag</td>
<td>Infrared detection flag</td>
<td>main(), r_main_infrared_detection()</td>
</tr>
<tr>
<td>uint8_t</td>
<td>g_motor_control_flag</td>
<td>Motor control flag</td>
<td>r_main_motor_control()</td>
</tr>
<tr>
<td>uint8_t</td>
<td>g_200ms_counter</td>
<td>Number of occurrences of 12-bit interval timer interrupt</td>
<td>main(), r_it_interrupt(), r_intc0_interrupt(), r_intc2_interrupt()</td>
</tr>
</tbody>
</table>
5.5 Functions

Table 5.4 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>R_MAIN_Userinit</td>
<td>Main initial setting</td>
</tr>
<tr>
<td>R_INTC0_Start</td>
<td>External interrupt enable function of INTP0</td>
</tr>
<tr>
<td>R_INTC2_Start</td>
<td>External interrupt enable function of INTP2</td>
</tr>
<tr>
<td>R_IT_Start</td>
<td>12-bit interval timer operation start function</td>
</tr>
<tr>
<td>R_TAU0_Channel0_Start</td>
<td>Timer array unit channel0 operation start function</td>
</tr>
<tr>
<td>R_TAU0_Channel0_Stop</td>
<td>Timer array unit channel0 operation stop function</td>
</tr>
<tr>
<td>R_TAU0_Channel2_Start</td>
<td>Timer array unit channel2 operation start function</td>
</tr>
<tr>
<td>R_TAU0_Channel2_Stop</td>
<td>Timer array unit channel2 operation stop function</td>
</tr>
<tr>
<td>R_TAU0_Channel3_Start</td>
<td>Timer array unit channel3 operation start function</td>
</tr>
<tr>
<td>R_TAU0_Channel3_Stop</td>
<td>Timer array unit channel3 operation stop function</td>
</tr>
<tr>
<td>r_it_interrupt</td>
<td>Interval timer signal detection interrupt function</td>
</tr>
<tr>
<td>r_intc0_interrupt</td>
<td>External interrupt processing of INTP0</td>
</tr>
<tr>
<td>r_intc2_interrupt</td>
<td>External interrupt processing of INTP2</td>
</tr>
<tr>
<td>r_main_motor_control</td>
<td>Motor control function</td>
</tr>
<tr>
<td>r_main_infrared_detection</td>
<td>Infrared detection processing</td>
</tr>
<tr>
<td>r_main_led_light</td>
<td>LED blinking processing</td>
</tr>
</tbody>
</table>
5.6 Function Specifications

This part describes function specifications of the sample code.

**[Function name] main**

<table>
<thead>
<tr>
<th>Outline</th>
<th>Main processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_cgc.h, r_cg_port.h, r_cg_tau.h, r_cg_it.h, r_cg_intp.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>—</td>
</tr>
<tr>
<td>Description</td>
<td>After executing the main user initialization function, this function shifts the system to STOP mode, performs the infrared detection process approximately every 200 ms. If infrared light has been detected, controls the motor to open the lid and turns on/off the LED. If no infrared light has been detected for five seconds, controls the motor to close the lid.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return value</td>
<td>None</td>
</tr>
<tr>
<td>Remarks</td>
<td>None</td>
</tr>
</tbody>
</table>

**[Function name] R_MAIN_UserInit**

<table>
<thead>
<tr>
<th>Outline</th>
<th>Main initial setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_cgc.h, r_cg_port.h, r_cg_tau.h, r_cg_it.h, r_cg_intp.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>static void R_MAIN_UserInit(void);</td>
</tr>
<tr>
<td>Description</td>
<td>This function enables the interrupts by using the EI instruction. The function then controls the motor to close the lid and turns on/off the LED.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return value</td>
<td>None</td>
</tr>
<tr>
<td>Remarks</td>
<td>None</td>
</tr>
</tbody>
</table>

**[Function name] R_INTC0_Start**

<table>
<thead>
<tr>
<th>Outline</th>
<th>External interrupt enable function of INTP0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_INTC0_Start(void);</td>
</tr>
<tr>
<td>Description</td>
<td>This function is used for setting to enable operation of the external interrupt of INTP0.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return value</td>
<td>None</td>
</tr>
<tr>
<td>Remarks</td>
<td>None</td>
</tr>
</tbody>
</table>

**[Function name] R_INTC2_Start**

<table>
<thead>
<tr>
<th>Outline</th>
<th>External interrupt enable function of INTP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_INTC2_Start(void);</td>
</tr>
<tr>
<td>Description</td>
<td>This function is used for setting to enable operation of the external interrupt of INTP2.</td>
</tr>
<tr>
<td>Arguments</td>
<td>None</td>
</tr>
<tr>
<td>Return value</td>
<td>None</td>
</tr>
<tr>
<td>Remarks</td>
<td>None</td>
</tr>
</tbody>
</table>
**[Function name] R_IT_Start**

Outline 12-bit interval timer operation start function

Header \ r_cg_macrodriver.h, r_cg_it.h, r_cg_userdefine.h

Declaration void R_IT_Start(void);

Description This function is used for setting to enable activation of the 12-bit interval timer.

Arguments None

Return value None

Remarks None

---

**[Function name] R_TAU0_Channel0_Start**

Outline Timer array unit channel0 operation start function

Header \ r_cg_macrodriver.h, r_cg_tau.h, r_cg_userdefine.h

Declaration void R_TAU0_Channel0_Start(void);

Description This function is used for setting to enable count operation of channel 0 of the timer array unit.

Arguments None

Return value None

Remarks None

---

**[Function name] R_TAU0_Channel0_Stop**

Outline Timer array unit channel0 operation stop function

Header \ r_cg_macrodriver.h, r_cg_tau.h, r_cg_userdefine.h

Declaration void R_TAU0_Channel0_Stop(void);

Description This function is used for setting to disable count operation of channel 0 of the timer array unit.

Arguments None

Return value None

Remarks None

---

**[Function name] R_TAU0_Channel2_Start**

Outline Timer array unit channel2 operation start function

Header \ r_cg_macrodriver.h, r_cg_tau.h, r_cg_userdefine.h

Declaration void R_TAU0_Channel2_Start(void);

Description This function is used for setting to enable count operation of channel 2 of the timer array unit.

Arguments None

Return value None

Remarks None

---

**[Function name] R_TAU0_Channel2_Stop**

Outline Timer array unit channel2 operation stop function

Header \ r_cg_macrodriver.h, r_cg_tau.h, r_cg_userdefine.h

Declaration void R_TAU0_Channel2_Stop(void);

Description This function is used for setting to disable count operation of channel 2 of the timer array unit.

Arguments None

Return value None

Remarks None
[Function name] R_TAU0_Channel3_Start
Outline Timer array unit channel3 operation start function
Header r_cg_macrodriver.h, r_cg_tau.h, r_cg_userdefine.h
Declaration void R_TAU0_Channel3_Start(void);
Description This function is used for setting to enable count operation of channel 3 of the timer array unit.
Arguments None
Return value None
Remarks None

[Function name] R_TAU0_Channel3_Stop
Outline Timer array unit channel3 operation stop function
Header r_cg_macrodriver.h, r_cg_tau.h, r_cg_userdefine.h
Declaration void R_TAU0_Channel3_Stop(void);
Description This function is used for setting to disable count operation of channel 3 of the timer array unit.
Arguments None
Return value None
Remarks None

[Function name] r_it_interrupt
Outline Interval timer signal detection interrupt function
Header r_cg_macrodriver.h, r_cg_it.h, r_cg_userdefine.h
Declaration #pragma interrupt r_it_interrupt(vect=INTIT)
Description This function increments the 200-ms counter variable.
Arguments None
Return value None
Remarks None

[Function name] r_intc0_interrupt
Outline External interrupt processing of INTP0
Header r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration #pragma interrupt r_intc0_interrupt(vect=INTP0)
Description This function controls the motor to close the lid and turns on/off the LED.
Arguments None
Return value None
Remarks None

[Function name] r_intc2_interrupt
Outline External interrupt processing of INTP2
Header r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h
Declaration #pragma interrupt r_intc2_interrupt(vect=INTP2)
Description This function controls the motor to open the lid and turns on/off the LED.
Arguments None
Return value None
Remarks None
[Function name] r_main_motor_control

Outline Motor control function
Header r_cg_macrodriver.h, r_cg_cgc.h, r_cg_port.h, r_cg_tau.h, r_cg_it.h, r_cg_intp.h, r_cg_userdefine.h
Declaration void r_main_motor_control(void);
Description This function controls the motor to open and close the lid.
Arguments None
Return value None
Remarks None

[Function name] r_main_infrared_detection

Outline Infrared detection processing
Header r_cg_macrodriver.h, r_cg_cgc.h, r_cg_port.h, r_cg_tau.h, r_cg_it.h, r_cg_intp.h, r_cg_userdefine.h
Declaration static void r_main_infrared_detection(void);
Description This function performs the infrared detection process.
Arguments None
Return value None
Remarks None

[Function name] r_main_led_light

Outline LED blinking processing
Header r_cg_macrodriver.h, r_cg_cgc.h, r_cg_port.h, r_cg_tau.h, r_cg_it.h, r_cg_intp.h, r_cg_userdefine.h
Declaration static void r_main_led_light(void);
Description This function performs the LED blinking process.
Arguments None
Return value None
Remarks None
5.7 Flowcharts

Figure 5.1 shows an overall flow of the sample code.

![Overall Flow Diagram](image)

**5.7.1 Initialization Function**

Figure 5.2 shows the flowchart for the initialization function.

![Initialization Function Diagram](image)
5.7.2 System Initial Setting

Figure 5.3 shows the flowchart for the system initial setting.

![Flowchart of System Initial Setting]

- **R_Systeminit()**
- **Peripheral I/O redirection**
  - Function disable
  - PIOR0 register ← 00H
- **Input/output setting**
  - R_PORT_Create()
- **CPU clock initialization setting**
  - R_CGC_Create()
- **Timer array unit initialization setting**
  - R_TAU0_Create()
- **External interrupt initialization setting**
  - R_INTC_Create()
- **12-bit interval timer initialization setting**
  - R_IT_Create()
- **return**

Figure 5.3 System Initial Setting
5.7.3 Ports Initial Setting

Figure 5.4 shows the flowchart for the ports initial setting.

```
R_PORT_Create()

Set the port register
P0 register ←00H

Set the analog input alternate-function pins
PM0 register ←98H

Set the port mode register
PMC0 register ←91H

return
```

![Figure 5.4 Port Initial Setting](image)

Note: Refer to the initialization flowchart in the RL78/G10 Initialization CC-RL (R01AN2668E) Application Note for details on how to set unused ports.

Caution: When designing circuits, always make sure unused ports are properly processed and all electrical characteristics are met. Also make sure each unused input-only port is connected to VDD or VSS through a resister.
5.7.4 CPU Initial Setting

Figure 5.5 shows the flowchart for the CPU initial setting.

- **R_CGC_Create()**
- **Set high-speed system clock**
  - CMC register ← 00H: High-speed system clock pin operation mode: input port
- **Set clock operation control**
  - MSTOP bit ← 1: X1 oscillator stopped.
- **Select CPU/peripheral hardware (fCLK)**
  - MCM0 bit ← 0: Select main on-chip oscillator clock (fOCO) as main system clock (fCLK)
  - OSMC register ← 10H: Select Low-speed on-chip oscillator clock as operating clock of 12-bit interval timer.
  - HIOSTOP bit ← 0: High-speed on-chip oscillator operating
- **return**

Figure 5.5 CPU Initial Setting
5.7.5 Timer Array Unit Initial Setting

Figure 5.6 and Figure 5.7 shows the flowchart for timer array unit initial setting.

```
R_TAU0_Create()

Supply clock to the timer array unit
TAU0EN bit ←1

Selection of operation clock
CK00 = f_CLK
CK01 = f_CLK
TPS0 register ←00H

Deactivate timer array unit 0
TH0 register ←0AH
TT0 register ←0FH

Disable TAU interrupts
TMMK00 bit ←1
TMIF00 bit ←0
TMMK01 bit ←1
TMIF01 bit ←0
TMMK01H bit ←1
TMIF01H bit ←0
TMMK02 bit ←1
TMIF02 bit ←0
TMMK03 bit ←1
TMIF03 bit ←0
TMMK03H bit ←1
TMIF03H bit ←0

Clear TAU interrupt request flag
TMMK00 bit ←1
TMIF00 bit ←0
TMMK01 bit ←1
TMIF01 bit ←0
TMMK01H bit ←1
TMIF01H bit ←0
TMMK02 bit ←1
TMIF02 bit ←0
TMMK03 bit ←1
TMIF03 bit ←0
TMMK03H bit ←1
TMIF03H bit ←0

Set interrupt priority level
Channel 0 : level3
Channel 1 : level3
Channel 2 : level3
Channel 3 : level3
TMPR100 bit ←1
TMPR000 bit ←1
TMPR101 bit ←1
TMPR001 bit ←1
TMPR102 bit ←1
TMPR002 bit ←1
TMPR103 bit ←1
TMPR003 bit ←1

Initial setting of channel 0
- Operation clock: CK00
- Operation mode: Interval timer mode (PWM output (master))
- Simultaneous channel operation function
- Software trigger start
- Set channel 0 output to 0
- Output of channel 0 is disabled

A

Figure 5.6 Timer Array Unit Initial Setting (1/2)
```
Initial setting of channel 1
- Operation clock: CK00
- Operation mode: Interval timer mode
  (PWM output (slave))
- Simultaneous channel operation function
- Software trigger start
- Set channel 1 output to 0
- Channel 1 is negative logic output
- Output of channel 1 is enabled

TMR01H register ← 04H
TMR01L register ← 09H
TDR01H register ← 57H
TDR01L register ← E4H
TOM01 bit ← 1
TOL01 bit ← 1
TO01 bit ← 0
TOE01 bit ← 1

Initial setting of channel 2
- Operation clock: CK00
- Operation mode: Interval timer mode
- Independent channel operation function
- Software trigger start
- Set channel 2 output to 0
- Output of channel 2 is disabled

TMR02H register ← 00H
TMR02L register ← 00H
TDR02H register ← 18H
TDR02L register ← 69H
TOM02 bit ← 0
TOL02 bit ← 0
TO02 bit ← 0
TOE02 bit ← 0

Initial setting of channel 3
- Operation clock: CK00
- Operation mode: Interval timer mode
- Independent channel operation function
- Software trigger start
- Set channel 3 output to 0
- Channel 3 is negative logic output
- Output of channel 3 is enabled

TMR03H register ← 00H
TMR03L register ← 00H
TDR03H register ← 00H
TDR03L register ← 0FH
TOM03 bit ← 0
TOL03 bit ← 1
TO03 bit ← 0
TOE03 bit ← 1

Set P04 to output port

Set P07 to output port

return

Figure 5.7 Timer Array Unit Initial Setting (2/2)
Clock supply to timer array unit started
- Peripheral enable register 0 (PER0)

Clock supply to timer array unit

Symbol: PER0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMKAEN</td>
<td>CMPEN</td>
<td>ADCEN</td>
<td>IICAOEN</td>
<td>0</td>
<td>SAU0EN</td>
<td>0</td>
<td>TAU0EN</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>x</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Bit 0

<table>
<thead>
<tr>
<th>TAU0EN</th>
<th>Control of timer array unit input clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Stops supply of input clock.</td>
</tr>
<tr>
<td>1</td>
<td>Supplies input clock.</td>
</tr>
</tbody>
</table>

Operation clock setting
- Timer clock select register 0 (TPS0)

Selection of operation clock (CK00)

Symbol: TPS0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRS013</td>
<td>PRS012</td>
<td>PRS011</td>
<td>PRS010</td>
<td>PRS003</td>
<td>PRS002</td>
<td>PRS001</td>
<td>PRS000</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 3-0

<table>
<thead>
<tr>
<th>PRS003</th>
<th>PRS002</th>
<th>PRS001</th>
<th>PRS000</th>
<th>Operation Clock (CK00) Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>fCLK 1.25 MHz</td>
<td>fCLK 2.5 MHz</td>
<td>fCLK 5 MHz</td>
<td>fCLK 10 MHz</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>fCLK/2 625 kHz</td>
<td>1.25 MHz 2.5 MHz</td>
<td>2.5 MHz 5 MHz</td>
<td>5 MHz 10 MHz</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>fCLK/2 313 kHz</td>
<td>625 kHz 1.25 MHz</td>
<td>2.5 MHz 5 MHz</td>
<td>5 MHz 10 MHz</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>fCLK/2 156 kHz</td>
<td>625 kHz 1.25 MHz</td>
<td>2.5 MHz 5 MHz</td>
<td>5 MHz 10 MHz</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>fCLK/2 78.1 kHz</td>
<td>156 kHz 313 kHz</td>
<td>625 kHz 1.25 MHz</td>
<td>2.5 MHz 5 MHz</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>fCLK/2 39.1 kHz</td>
<td>78.1 kHz 156 kHz</td>
<td>313 kHz 625 kHz</td>
<td>1.25 MHz 2.5 MHz</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>fCLK/2 19.5 kHz</td>
<td>78.1 kHz 156 kHz</td>
<td>313 kHz 625 kHz</td>
<td>1.25 MHz 2.5 MHz</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>fCLK/2 9.76 kHz</td>
<td>78.1 kHz 156 kHz</td>
<td>313 kHz 625 kHz</td>
<td>1.25 MHz 2.5 MHz</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>fCLK/2^4 4.88 kHz</td>
<td>9.76 kHz 19.5 kHz</td>
<td>39.1 kHz 78.1 kHz</td>
<td>39.1 kHz 78.1 kHz</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>fCLK/2^4 2.44 kHz</td>
<td>4.88 kHz 9.76 kHz</td>
<td>19.5 kHz 39.1 kHz</td>
<td>39.1 kHz 78.1 kHz</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>fCLK/2^4 1.22 kHz</td>
<td>2.44 kHz 4.88 kHz</td>
<td>9.76 kHz 19.5 kHz</td>
<td>39.1 kHz 78.1 kHz</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>fCLK/2^4 610 Hz</td>
<td>1.22 kHz 2.44 kHz</td>
<td>4.88 kHz 9.77 kHz</td>
<td>19.5 kHz 39.1 kHz</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>fCLK/2^14 305 Hz</td>
<td>610 Hz 1.22 kHz</td>
<td>2.44 kHz 4.88 kHz</td>
<td>19.5 kHz 39.1 kHz</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>fCLK/2^14 153 Hz</td>
<td>305 Hz 610 Hz</td>
<td>1.22 kHz 2.44 kHz</td>
<td>19.5 kHz 39.1 kHz</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>fCLK/2^14 76.3 Hz</td>
<td>153 Hz 305 Hz</td>
<td>610 Hz 1.22 kHz</td>
<td>19.5 kHz 39.1 kHz</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>fCLK/2^14 38.1 Hz</td>
<td>76.3 Hz 153 Hz</td>
<td>305 Hz 610 Hz</td>
<td>19.5 kHz 39.1 kHz</td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Channel stop control
- Timer channel stop register 0 (TT0)
  Stop the counting operation of each channel

Symbol: TT0

<table>
<thead>
<tr>
<th>Bit 3-0</th>
<th>Operation stop trigger of channel n (n = 0 to 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No trigger operation</td>
</tr>
<tr>
<td>1</td>
<td>TE0n is cleared to 0, and counting operation is stopped.</td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Setting up the channel 0 operation mode

- Timer mode register 00 (TMR00H, TMR00L)
- Select an operation clock ($f_{MCK}$).
- Select a count clock.
- Select a start trigger and capture trigger.
- Select a valid edge for timer input.

Set up the operation mode.

Symbol: TMR00H

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKS001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CCS00</td>
<td>0</td>
<td>STS002</td>
<td>STS001</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 7

<table>
<thead>
<tr>
<th>CKS001</th>
<th>Selection of operation clock ($f_{MCK}$) of channel 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation clock CK00 set by timer clock select register 0 (TPS0)</td>
</tr>
<tr>
<td>0</td>
<td>Operation clock CK01 set by timer clock select register 0 (TPS0)</td>
</tr>
</tbody>
</table>

Bit 4

<table>
<thead>
<tr>
<th>CCS00</th>
<th>Selection of count clock ($f_{TCLK}$) of channel 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation clock ($f_{MCK}$) specified by the CKS001 bit</td>
</tr>
<tr>
<td>1</td>
<td>Valid edge of input signal input from the TI00 pin</td>
</tr>
</tbody>
</table>

Bit 2-0

<table>
<thead>
<tr>
<th>STS002</th>
<th>STS001</th>
<th>STS000</th>
<th>Setting of start trigger or capture trigger of channel 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Only software trigger start is valid (other trigger sources are unselected).</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Valid edge of the TI00 pin input is used as the start trigger and capture trigger.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Both the edges of the TI00 pin input are used as a start trigger and a capture trigger.</td>
</tr>
</tbody>
</table>
| 1      | 0      | 0      | When the channel is used as a slave channel with the one-shot pulse output, PWM output function, or multiple PWM output function: 
The interrupt request signal of the master channel (INTTM00) is used as the start trigger. |
| 1      | 1      | 0      | When the channel is used as a slave channel in two-channel input with one-shot pulse output function: 
The interrupt request signal of the master channel (INTTM00) is used as the start trigger. A valid edge of the TI03 pin input of the slave channel is used as the end trigger |

Other than above Setting prohibited

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Symbol: TMR00L

<table>
<thead>
<tr>
<th>Bit 3-0</th>
<th>Operation mode of channel 0</th>
<th>Related function</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1/0</td>
<td>Interval timer mode</td>
<td>Interval timer / Square wave output / Divider function / PWM output (master)</td>
<td>Counting down</td>
</tr>
<tr>
<td>0 1 0 1/0</td>
<td>Capture mode</td>
<td>Input pulse interval measurement</td>
<td>Counting up</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>Event counter mode</td>
<td>External event counter</td>
<td>Counting down</td>
</tr>
<tr>
<td>1 0 0 1/0</td>
<td>One-count mode</td>
<td>Delay counter / One-shot pulse output / PWM output (slave)</td>
<td>Counting down</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>Capture &amp; one-count mode</td>
<td>Measurement of high-/low-level width of input signal</td>
<td>Counting up</td>
</tr>
<tr>
<td>Other than above</td>
<td>Setting prohibited</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The MD000 bit operation varies depending on the operation mode (see the table below)

<table>
<thead>
<tr>
<th>Operation mode (Value set by the MD003 to MD001 bits) (see table above)</th>
<th>MD000</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Interval timer mode (0, 0, 0)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started (timer output does not change, either).</td>
</tr>
<tr>
<td>• Capture mode (0, 1, 0)</td>
<td>1</td>
<td>Timer interrupt is generated when counting is started (timer output also changes).</td>
</tr>
<tr>
<td>• Event counter mode (0, 1, 1)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started (timer output does not change, either).</td>
</tr>
<tr>
<td>• One-count mode (1, 0, 0)</td>
<td>0</td>
<td>Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Start trigger is valid during counting operation. At that time, interrupt is also generated.</td>
</tr>
<tr>
<td>• Capture &amp; one-count mode (1, 1, 0)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Other than above</td>
<td>Setting prohibited</td>
<td></td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Setting up the channel 1 operation mode

- Timer mode register 01 (TMR01H, TMR01L)
  Select an operation clock (fMCK).
  Select a count clock.
  Select the 16/8-bit timer.
  Select a start trigger and capture trigger.
  Select a valid edge for timer input.
  Set up the operation mode.

Symbol: TMR01H

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKS011</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CCS01</td>
<td>SPLIT01</td>
<td>STS012</td>
<td>STS011</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 7

**CKS011** Channel 1 operation clock (fMCK) selection

| 0 | Operation clock CK00 set by timer clock select register 0 (TPS0) |
| 1 | Operation clock CK01 set by timer clock select register 0 (TPS0) |

Bit 4

**CCS01** Selection of count clock (fTCLK) of channel 1

| 0 | Operation clock (fMCK) specified with the CKS011 bit |
| 1 | Valid edge of the input signal from the TI01 pin |

Bit 3

**SPLIT01** Selection of 8 or 16-bit timer operation for channel 1

| 0 | Operates as 16-bit timer |
| 1 | Operates as 8-bit timer |

Bit 2-0

**STS002** **STS001** **STS000** Setting of start trigger or capture trigger of channel 1

| 0 | 0 | 0 | Only software trigger start is valid (other trigger sources are unselected). |
| 0 | 0 | 1 | Valid edge of the TI01 pin input is used as both the start trigger and capture trigger. |
| 0 | 1 | 0 | Both the edges of the TI01 pin input are used as a start trigger and a capture trigger. |
| 1 | 0 | 0 | Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function). |
| 1 | 1 | 0 | When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM0n) is used as the start trigger. A valid edge of the TI03 pin input of the slave channel is used as the end trigger |

Other than above Setting prohibited

Note: Refer to the RL78/G10 User's Manual (Hardware version) for details on how to set registers.
Symbol: TMR01L

<table>
<thead>
<tr>
<th>Bit 3-0</th>
<th>Operation mode of channel 0</th>
<th>Related function</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1/0</td>
<td>Interval timer mode</td>
<td>Interval timer / Square wave output / Divider function / PWM output (master)</td>
<td>Counting down</td>
</tr>
<tr>
<td>0 1 0 1/0</td>
<td>Capture mode</td>
<td>Input pulse interval measurement</td>
<td>Counting up</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>Event counter mode</td>
<td>External event counter</td>
<td>Counting down</td>
</tr>
<tr>
<td>1 0 0 1/0</td>
<td>One-count mode</td>
<td>Delay counter / One-shot pulse output / PWM output (slave)</td>
<td>Counting down</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>Capture &amp; one-count mode</td>
<td>Measurement of high-/low-level width of input signal</td>
<td>Counting up</td>
</tr>
</tbody>
</table>

The MD000 bit operation varies depending on the operation mode (see the table below).

<table>
<thead>
<tr>
<th>Operation mode (Value set by the MD003 to MD001 bits) (see table above)</th>
<th>MD000</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Interval timer mode (0, 0, 0)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started. Timer output does not change, either.</td>
</tr>
<tr>
<td>• Capture mode (0, 1, 0)</td>
<td>1</td>
<td>Timer interrupt is generated when counting is started. Timer output also changes.</td>
</tr>
<tr>
<td>• Event counter mode (0, 1, 1)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started. Timer output does not change, either.</td>
</tr>
<tr>
<td>• One-count mode (1, 0, 0)</td>
<td>0</td>
<td>Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Start trigger is valid during counting operation. At that time, interrupt is also generated.</td>
</tr>
<tr>
<td>• Capture &amp; one-count mode (1, 1, 0)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started. Timer output does not change, either. Start trigger is invalid during counting operation. At that time interrupt is not generated, either.</td>
</tr>
</tbody>
</table>

Other than above Setting prohibited

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Setting up the channel 2 operation mode

- Timer mode register 02 (TMR02H, TMR02L)
  Select an operation clock (fMCK).
  Select a count clock.
  Select a software trigger start.
  Set up the operation mode.

Symbol: TMR02H

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>Selection of operation clock (fMCK) of channel 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKS021</td>
<td>Operation clock CK00 set by timer clock select register 0 (TPS0)</td>
</tr>
<tr>
<td>0</td>
<td>Operation clock CK01 set by timer clock select register 0 (TPS0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 4</th>
<th>Selection of count clock (fTCLK) of channel 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCS02</td>
<td>Operation clock (fMCK) specified by the CKS021 bit</td>
</tr>
<tr>
<td>0</td>
<td>Valid edge of input signal input from the TI02 pin</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 3</th>
<th>Selection of independent channel operation/simultaneous channel operation (slave/master) of channel 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASTER0</td>
<td>Operates as the slave channel in the independent channel operation function or the simultaneous channel operation function.</td>
</tr>
<tr>
<td>0</td>
<td>Operates as the master channel in the simultaneous channel operation function.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 2-0</th>
<th>Setting of start trigger or capture trigger of channel 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>STS002</td>
<td>STS001</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Symbol: TMR02L

<table>
<thead>
<tr>
<th>CIS021</th>
<th>CIS020</th>
<th>0</th>
<th>0</th>
<th>MD023</th>
<th>MD022</th>
<th>MD021</th>
<th>MD020</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 7, 6

<table>
<thead>
<tr>
<th>CIS021</th>
<th>CIS020</th>
<th>Selection of TI02 pin input valid edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Falling edge</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Rising edge</td>
</tr>
</tbody>
</table>
| 1      | 0      | Both edges (when low-level width is measured)  
Start trigger: Falling edge, Capture trigger: Rising edge |
| 1      | 1      | Both edges (when high-level width is measured)  
Start trigger: Rising edge, Capture trigger: Falling edge |

Bit 3-0

<table>
<thead>
<tr>
<th>MD0 23</th>
<th>MD0 22</th>
<th>MD0 21</th>
<th>MD0 20</th>
<th>Operation mode of channel 2</th>
<th>Related function</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/0</td>
<td>Interval timer mode</td>
<td>Interval timer / Square wave output / Divider function / PWM output (master)</td>
<td>Counting down</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/0</td>
<td>Capture mode</td>
<td>Input pulse interval measurement</td>
<td>Counting up</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Event counter mode</td>
<td>External event counter</td>
<td>Counting down</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1/0</td>
<td>One-count mode</td>
<td>Delay counter / One-shot pulse output / PWM output (slave)</td>
<td>Counting down</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Capture &amp; one-count mode</td>
<td>Measurement of high-/low-level width of input signal</td>
<td>Counting up</td>
</tr>
<tr>
<td>Other than above</td>
<td>Setting prohibited</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The MD020 bit operation varies depending on the operation mode (see the table below)

<table>
<thead>
<tr>
<th>Operation mode (Value set by the MD023 to MD021 bits) (see table above)</th>
<th>MD020</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Interval timer mode (0, 0, 0)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started (timer output does not change, either).</td>
</tr>
<tr>
<td>• Capture mode (0, 1, 0)</td>
<td>1</td>
<td>Timer interrupt is generated when counting is started (timer output also changes).</td>
</tr>
<tr>
<td>• Event counter mode (0, 1, 1)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started (timer output does not change, either).</td>
</tr>
<tr>
<td>• One-count mode (1, 0, 0)</td>
<td>0</td>
<td>Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.</td>
</tr>
<tr>
<td>• Capture &amp; one-count mode (1, 1, 0)</td>
<td>1</td>
<td>Start trigger is valid during counting operation. At that time, interrupt is also generated.</td>
</tr>
<tr>
<td>Other than above</td>
<td>Setting prohibited</td>
<td></td>
</tr>
</tbody>
</table>
Setting up the channel 3 operation mode
- Timer mode register 03 (TMR03H, TMR03L)
  Select an operation clock (f_{MCK}).
  Select a count clock.
  Select a software trigger start.
  Set up the operation mode.

Symbol: TMR03H

<table>
<thead>
<tr>
<th>Bit 7</th>
<th>CKS031</th>
<th>Selection of operation clock (f_{MCK}) of channel 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Operation clock CK00 set by timer clock select register 0 (TPS0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Operation clock CK01 set by timer clock select register 0 (TPS0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 4</th>
<th>CCS03</th>
<th>Selection of count clock (f_{TCLK}) of channel 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Operation clock (f_{MCK}) specified by the CKS031 bit</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Valid edge of input signal input from the TI03 pin</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 3</th>
<th>SPLIT03</th>
<th>Selection of 8 or 16-bit timer operation for channels 1 and 3 (n = 1, 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Operates as 16-bit timer.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Operates as 8-bit timer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 2-0</th>
<th>STS032</th>
<th>STS031</th>
<th>STS030</th>
<th>Setting of start trigger or capture trigger of channel 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Only software trigger start is valid (other trigger sources are unselected).</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Valid edge of the TI03 pin input is used as the start trigger and capture trigger.</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Both the edges of the TI03 pin input are used as a start trigger and a capture trigger.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>When the channel is used as a slave channel with the one-shot pulse output, PWM output function, or multiple PWM output function: The interrupt request signal of the master channel (INTTM03) is used as the start trigger.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM03) is used as the start trigger. A valid edge of the TI03 pin input of the slave channel is used as the end trigger</td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Bit 7, 6

<table>
<thead>
<tr>
<th>CIS031</th>
<th>CIS030</th>
<th>Selection of TIO3 pin input valid edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Falling edge</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Rising edge</td>
</tr>
</tbody>
</table>
| 1      | 0      | Both edges (when low-level width is measured)  
Start trigger: Falling edge, Capture trigger: Rising edge |
| 1      | 1      | Both edges (when high-level width is measured)  
Start trigger: Rising edge, Capture trigger: Falling edge |

Bit 3-0

<table>
<thead>
<tr>
<th>MD033</th>
<th>MD032</th>
<th>MD031</th>
<th>MD030</th>
<th>Operation mode of channel 3</th>
<th>Related function</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/0</td>
<td>Interval timer mode</td>
<td>Interval timer / Square wave output / Divider function / PWM output (master)</td>
<td>Counting down</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/0</td>
<td>Capture mode</td>
<td>Input pulse interval measurement</td>
<td>Counting up</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Event counter mode</td>
<td>External event counter</td>
<td>Counting down</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1/0</td>
<td>One-count mode</td>
<td>Delay counter / One-shot pulse output / PWM output (slave)</td>
<td>Counting down</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Capture &amp; one-count mode</td>
<td>Measurement of high-/low-level width of input signal</td>
<td>Counting up</td>
</tr>
</tbody>
</table>

Other than above Setting prohibited

The MD030 bit operation varies depending on the operation mode (see the table below)

<table>
<thead>
<tr>
<th>Operation mode (Value set by the MD033 to MD031 bits) (see table above)</th>
<th>MD030</th>
<th>TCR counting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Interval timer mode (0, 0, 0)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started (timer output does not change, either).</td>
</tr>
<tr>
<td>• Capture mode (0, 1, 0)</td>
<td>1</td>
<td>Timer interrupt is generated when counting is started (timer output also changes).</td>
</tr>
<tr>
<td>• Event counter mode (0, 1, 1)</td>
<td>0</td>
<td>Timer interrupt is not generated when counting is started (timer output does not change, either).</td>
</tr>
</tbody>
</table>
| • One-count mode (1, 0, 0)                                             | 0     | Start trigger is invalid during counting operation.  
At that time, interrupt is not generated, either. |
| • Capture & one-count mode (1, 1, 0)                                   | 1     | Start trigger is valid during counting operation.  
At that time, interrupt is also generated. |
| Other than above Setting prohibited                                    |       |                        |

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Setting up the timer output mode

- Timer output mode register 0 (TOM0)
  Set up the timer output mode for each channel.

Symbol: TOM0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TOM03</td>
<td>TOM02</td>
<td>TOM01</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 1

<table>
<thead>
<tr>
<th>TOM01</th>
<th>Control of timer output mode of channel 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Used as the independent channel operation function (to produce toggle output by the interrupt request signal (INTTM01))</td>
</tr>
<tr>
<td>1</td>
<td>Slave channel output mode (output is set by the interrupt request signal (INTTM00, INTTM02) of the master channel, and reset by the timer interrupt request signal (INTTM01) of the slave channel)</td>
</tr>
</tbody>
</table>

Configuring the output level for the timer output pin

- Timer output level register 0 (TOL0)
  Configure the output level for the timer output pin for each channel.

Symbol: TOL0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TOL03</td>
<td>TOL02</td>
<td>TOL01</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 3, 1

<table>
<thead>
<tr>
<th>TOL0n</th>
<th>Control of timer output level of channel n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Positive logic output (active-high)</td>
</tr>
<tr>
<td>1</td>
<td>Positive logic output (active-high)</td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
Configuring the output value for the timer output pin
- Timer output register 0 (TO0)
  Configure the output value for the timer output pin for each channel.

Symbol: TO0

<table>
<thead>
<tr>
<th>Bit 3-0</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO0n</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TO03</td>
<td>TO02</td>
<td>TO01</td>
<td>TO00</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 3-0

<table>
<thead>
<tr>
<th>TO0n</th>
<th>Timer output of channel n</th>
</tr>
</thead>
</table>
| 0       | Timer output value is “0”.
| 1       | Timer output value is “1”.

Enabling the timer output
- Timer output enable register 0 (TOE0)
  Enable/disable the timer output for each channel.

Symbol: TOE0

<table>
<thead>
<tr>
<th>Bit 3, 1</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOE0n</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TOE03</td>
<td>TOE02</td>
<td>TOE01</td>
<td>TOE00</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Bit 3, 1

<table>
<thead>
<tr>
<th>TOE0n</th>
<th>Timer output enable/disable of channel n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Disable output of timer.</td>
</tr>
<tr>
<td></td>
<td>Without reflecting on TO0n bit timer operation, to fixed the output.</td>
</tr>
<tr>
<td></td>
<td>Writing to the TO0n bit is enabled and the level set in the TO0n bit is output from the TO0n pin.</td>
</tr>
<tr>
<td>1</td>
<td>Enable output of timer.</td>
</tr>
<tr>
<td></td>
<td>Reflected in the TO0n bit timer operation, to generate the output waveform.</td>
</tr>
<tr>
<td></td>
<td>Writing to the TO0n bit is disabled (writing is ignored).</td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
5.7.6 External Interrupt Setting

Figure 5.8 shows the flowchart for setting the external interrupt.

```
R_INTC_Create()

Disable INTPrn interrupt, clear the INTPrn interrupt request flag

Set the interrupt priority level

Set the valid edge of the external interrupt

Set P41 as input port

return

MK0L register
PMK1-PMK0 bit ← 11B : Interrupt servicing disabled
MK0H register
PMK3-PMK2 bit ← 11B : Interrupt servicing disabled
IF0L register
PIF1-PIF0 bit ← 00B : No interrupt request signal is generated
IF0H register
PIF3-PIF2 bit ← 00B : No interrupt request signal is generated

PR10L register
PPR10 bit ← 1 : Specifying level 3 (low priority)
PR00L register
PPR00 bit ← 1
PR10H register
PPR12 bit ← 1 : Specifying level 3 (low priority)
PR00H register
PPR02 bit ← 1

EGN0 register← 05H : Set the valid edge of the INTP0 and INTP2 pins to the falling edge

PM4.1 bit ← 1
```
Set edge detection of INTP0 pin and INTP2 pin
- External interrupt rising edge enable register 0 (EGP0)
- External interrupt falling edge enable register 0 (EGN0)

Set valid edge of INTP0 pin and INTP2 pin.

Symbol: EGP0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>EGP3</td>
<td>EGP2</td>
<td>EGP1</td>
<td>EGP0</td>
</tr>
</tbody>
</table>

Symbol: EGN0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>EGN3</td>
<td>EGN2</td>
<td>EGN1</td>
<td>EGN0</td>
</tr>
</tbody>
</table>

Set valid edge of INTP0 pin and INTP2 pin.

<table>
<thead>
<tr>
<th>EGPn</th>
<th>EGNn</th>
<th>INTPn pin valid edge selection (n = 0 to 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Edge detection disabled</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Falling edge</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Rising edge</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Both rising and falling edges</td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
5.7.7  12-bit Interval Timer Initial Setting

Figure 5.9 shows the flowchart for the 12-bit interval timer initial setting.

```
R_IT_Create()

Supply clock to the 12-bit interval timer
  TMKAEN bit ← 1

Stop 12-bit interval timer operation
  RINTE bit ← 0

Disable 12-bit interval timer interrupt
  ITMK bit ← 1
  ITIF bit ← 0

Clear the 12-bit interval timer interrupt request flag

Set interrupt priority level: level 3
  ITPR1 bit ← 1
  ITPR0 bit ← 1

Set the 12-bit interval timer to about 200ms
  ITMCH register ← 0BH
  ITMCL register ← B7H

return
```

Figure 5.9  12-bit Interval Timer Initial Setting
12-bit interval timer clock supply setting
- Peripheral enable register 0 (PER0)
  Enable clock supply to the 12-bit interval timer.

Symbol: PER0

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMKAEN</td>
<td>7</td>
<td>Control of 12-bit interval timer input clock supply</td>
</tr>
<tr>
<td>CMPEN</td>
<td>6</td>
<td>Enables input clock supply.</td>
</tr>
<tr>
<td>ADCEN</td>
<td>5</td>
<td>Enables input clock supply.</td>
</tr>
<tr>
<td>IICA0EN</td>
<td>4</td>
<td>Enables input clock supply.</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>Enables input clock supply.</td>
</tr>
<tr>
<td>SAU0EN</td>
<td>2</td>
<td>Enables input clock supply.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Enables input clock supply.</td>
</tr>
<tr>
<td>TAU0EN</td>
<td>0</td>
<td>Enables input clock supply.</td>
</tr>
</tbody>
</table>

12-bit interval timer operation and to specify the timer compare value setting
- 12-bit interval timer control register (ITMCH, ITMCL)
  Stop 12-bit interval timer count operation.
  Set the 12-bit interval timer compare value.

Symbol: ITMCH

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RINTE</td>
<td>7</td>
<td>12-bit interval timer operation control</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>Count operation stopped (count clear)</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>Count operation started</td>
</tr>
<tr>
<td>ITCMP11-ITCMP8</td>
<td>4-0</td>
<td>Specification of 12-bit interval timer compare value</td>
</tr>
</tbody>
</table>

12-bit interval timer operation and to specify the timer compare value setting
- 12-bit interval timer control register (ITMCH, ITMCL)
  Stop 12-bit interval timer count operation.
  Set the 12-bit interval timer compare value.

Symbol: ITMCL

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITCMP7-ITCMP0</td>
<td>7-0</td>
<td>Specification of 12-bit interval timer compare value</td>
</tr>
<tr>
<td>B7H</td>
<td>7-0</td>
<td>Setting prohibited</td>
</tr>
<tr>
<td>00H</td>
<td>Setting prohibited</td>
<td></td>
</tr>
</tbody>
</table>

Note: Refer to the RL78/G10 User’s Manual (Hardware version) for details on how to set registers.
5.7.8 Main Processing

Figure 5.10 to Figure 5.12 shows the flowchart of the main processing.

![Flowchart of Main Processing](image)

Figure 5.10 Main Processing (1/3)
Clear the 200-ms counter

C

Open/close flag = 1?

YES

LED (OPEN) ON

P0.1 bit ← 1

Motor control function
r_main_motor_control()

g_open_flag ← 0

Clear the Open/close flag

g_200ms_counter ← 00H

Clear the 200-ms counter

Clear the interrupt request of the 12-bit interval timer

ITIF bit ← 0

Infrared detection processing
main_infrared_detection()

Infrared detection flag = 0?

NO (Branch when infrared is detected)

YES

200-ms counter = 25?

YES

Motor control function
r_main_motor_control()

P0.1 bit ← 0

P0.0 bit ← 0

LED OFF

Change the operation status to STANDBY

g_operation_status ← STANDBY

Clear the 200-ms counter

g_200ms_counter ← 00H

Set the Open/close flag

g_open_flag ← 1

Clear the interrupt request of the 12-bit interval timer

ITIF bit ← 0

E

Figure 5.11 Main Processing (2/3)
D

200-ms counter ≥ 20?

YES

LED blinking processing
r_main_led_light()

Clear the 200-ms counter

NO

E

Figure 5.12  Main Processing (3/3)

g_200ms_counter ← 00H
5.7.9 Main Initial Setting

Figure 5.13 shows the flowchart of the main initial setting.

```
R_MAIN_UserInit()

Disable maskable interrupts
IE ← 0

LED (CLOSE) ON
P0.0 bit ← 1

Motor control function
r_main_motor_control()

LED OFF
P0.1 bit ← 0
P0.0 bit ← 0

Set the Open/close flag
g_open_flag ← 1

External interrupt enable function of INTP0
R_INTC0_Start()

External interrupt enable function of INTP2
R_INTC2_Start()

12-bit interval timer operation start function
R_IT_Start()

return
```

Figure 5.13 Main Initial Setting
5.7.10 Motor Control Function

Figure 5.14 and Figure 5.15 shows the flowchart of the main initial setting.

```
r_main_motor_control()

Open/close flag = 1?
  YES
    Set the motor signal 1 to low level output
    Set the PWM output width of motor signal 2
    Clear the motor control flag
    Disable 12-bit interval timer interrupt
    Timer array unit channel 0 operation start function
      R_TAU0_Channel0_Start()
    motor control flag = 1?
      YES
        Timer array unit channel 0 operation stop function
          R_TAU0_Channel0_Stop()
        Enable 12-bit interval timer interrupt
        LED OFF
      return
    HALT mode
      Clear the timer array unit channel 0 interrupt request flag
      Increment the interrupt request occurrence count counter
  NO
    g_motor_control_flag ← 0

Figure 5.14 Motor Control Function (1/2)
```
Open/close flag = 1?

Interrupt request occurrence count counter = 20?

Set the motor control flag

Interrupt request occurrence count counter = 51?

Set the motor control flag

Interrupt request occurrence count counter = 41?

Set the PWM output width of motor signal 2

Figure 5.15 Motor Control Function (2/2)
5.7.11 Infrared Detection Processing

Figure 5.16 and Figure 5.17 shows the flowchart of the infrared detection processing.

![Flowchart of Infrared Detection Processing](image-url)
Infrared was detected 5 times?

- **YES**
  - Set the interval time of channel 2 of timer array unit to 670us
  - Timer array unit channel 2 operation start function
    - R_TAU0_Channel2_Start()
  - Timer array unit channel 3 operation start function
    - R_TAU0_Channel3_Start()

- **NO**
  - HALT mode
  - Clear the timer array unit channel 2 interrupt request flag
    - Infrared was not detected?
      - **YES**
        - Set the status to END
        - Clear the infrared detection flag
      - **NO**

- **counter = 4?**
  - **NO**
    - Increment the counter
    - Timer array unit channel 3 operation stop function
      - R_TAU0_Channel3_Stop()
    - Timer array unit channel 2 operation stop function
      - R_TAU0_Channel2_Stop()
    - Set the interval time of channel 2 of timer array unit to 5ms

- **YES**
  - Set the status to END
    - status ← END
  - Set the infrared detection flag

Figure 5.17  Infrared Detection Processing (2/2)
5.7.12 LED Blinking Processing

Figure 5.18 shows the flowchart of the infrared detection processing.

![Flowchart of LED Blinking Processing]

Figure 5.18 LED Blinking Processing
5.7.13 External Interrupt Processing of INTP0

Figure 5.19 shows the flowchart of the external interrupt processing of intp0.

![Flowchart of INTP0]

5.7.14 External Interrupt Processing of INTP2

Figure 5.20 shows the flowchart of the external interrupt processing of intp2.

![Flowchart of INTP2]
5.7.15 12-bit Interval Timer Operation Start

Figure 5.21 shows the flowchart of the 12-bit interval timer operation start.

![Flowchart of 12-bit Interval Timer Operation Start]

```
R_IT_Start()

Enable 12-bit interval timer interrupt
ITIF bit ← 0
ITMK bit ← 0

Start 12-bit interval timer count
RINTE bit ← 1

return
```

Figure 5.21 12-bit Interval Timer Operation Start
5.7.16 Timer Array Unit Channel0 Operation Start Function

Figure 5.22 shows the flowchart of the timer array unit channel0 operation start function.

```
R_TAU0_Channel0_Start()

Timer array unit channel 0 interrupt enabled

TMIF00 bit ← 0
TMMK00 bit ← 0

Timer array unit channel 1 interrupt disabled

TMIF01 bit ← 0
TMMK01 bit ← 1

Enable timer output of channel 1 of timer array unit

TOE01 bit ← 1

Enable operation of channel0 and channel 1 of timer array unit

TS01 bit ← 1
TS00 bit ← 1

return
```

Figure 5.22 Timer Array Unit Channel0 Operation Start Function
5.7.17 Timer Array Unit Channel0 Operation Stop Function

Figure 5.23 shows the flowchart of the timer array unit channel0 operation stop function.

```
R_TAU0_Channel0_Stop()

Disable operation of channel0 and channel1 of timer array unit

Disable timer output of channel1 of timer array unit

Set the timer output value of channel1 of timer array unit to 0

Timer array unit channel0 interrupt disabled

Timer array unit channel1 interrupt disabled

return

TT01 bit ← 1
TT00 bit ← 1
TOE01 bit ← 0
TO01 bit ← 0
TMMK00 bit ← 1
TMIF00 bit ← 0
TMMK01 bit ← 1
TMIF01 bit ← 0
```

Figure 5.23 Timer Array Unit Channel0 Operation Stop Function
5.7.18 Timer Array Unit Channel2 Operation Start Function

Figure 5.24 shows the flowchart of the timer array unit channel2 operation start function.

![Flowchart of Timer Array Unit Channel2 Operation Start Function]

R_TAU0_Channel2_Start()

Timer array unit channel 2 interrupt enabled

Enable operation of channel 2 of timer array unit

return

TMIF02 bit ← 0
TMMK02 bit ← 0
TS02 bit ← 1

Figure 5.24 Timer Array Unit Channel2 Operation Start Function

5.7.19 Timer Array Unit Channel2 Operation Stop Function

Figure 5.25 shows the flowchart of the timer array unit channel2 operation stop function.

![Flowchart of Timer Array Unit Channel2 Operation Stop Function]

R_TAU0_Channel2_Stop()

Disable operation of channel 2 of timer array unit

TT02 bit ← 1

Timer array unit channel 2 interrupt disabled

TMMK02 bit ← 1
TMIF02 bit ← 0

return

Figure 5.25 Timer Array Unit Channel2 Operation Stop Function
5.7.20 Timer Array Unit Channel3 Operation Start Function

Figure 5.26 shows the flowchart of the timer array unit channel3 operation start function.

```
R_TAU0_Channel 3_Start()

Timer array unit channel 3 interrupt disabled

TMIF03 bit ← 0
TMMK03 bit ← 1

Enable timer output of channel 3
of timer array unit

TOE03 bit ← 1

Enable operation of channel 3
of timer array unit

TS03 bit ← 1

Set P07 to low level output

return
```

Figure 5.26 Timer Array Unit Channel3 Operation Start Function
5.7.21 Timer Array Unit Channel3 Operation Stop Function

Figure 5.27 shows the flowchart of the timer array unit channel3 operation stop function.

```
R_TAU0_Channel3_Stop()

Set P07 to high level output

Disable operation of channel 3 of timer array unit

Disable timer output of channel 3 of timer array unit

Timer array unit channel 3 interrupt disabled

return

TT03 bit ← 1
TOE03 bit ← 0
TMMK03 bit ← 1
TMIF03 bit ← 0
```

Figure 5.27 Timer Array Unit Channel3 Operation Stop Function
5.7.22 Interval Timer Signal Detection Interrupt Function

Figure 5.28 shows the flowchart of the interval timer signal detection interrupt function.

```
r_it_interrupt()

Increment the 200-ms counter
g_200ms_counter ← g_200ms_counter + 1

return
```

Figure 5.28 Interval Timer Signal Detection Interrupt Function
5.7.23 External Interrupt Processing of INTP0

Figure 5.29 and Figure 5.30 shows the flowchart of the external interrupt processing of INTP0.

![Flowchart of External Interrupt Processing of INTP0](image-url)
Set the 200-ms counter to 2  
\[ g_{200ms\_counter} \leftarrow 2 \]

Clear the INTP0 external interrupt request flag  
\[ PIF0 \leftarrow 0 \]

Clear the INTP2 external interrupt request flag  
\[ PIF2 \leftarrow 0 \]

3 times execution?  
NO

HALT mode

Clear the 12-bit internal timer interrupt request flag  
\[ ITIF \leftarrow 0 \]

Figure 5.30 External Interrupt Processing of INTP0 (2/2)
5.7.24 External Interrupt Processing of INTP2

Figure 5.31 shows the flowchart of the external interrupt processing of INTP0.

![Flowchart of External Interrupt Processing of INTP0](image-url)
6. **Sample Code**

   The sample code is available on the Renesas Electronics Website.

7. **Documents for Reference**

   RL78/G10 User's Manual: Hardware (R01UH0384E)
   RL78 Family User's Manual: Software (R01US0015E)

   (The latest versions of the documents are available on the Renesas Electronics Website.)

   Technical Updates/Technical Brochures

   (The latest versions of the documents are available on the Renesas Electronics Website.)

**Website and Support**

Renesas Electronics website
   [http://www.renesas.com](http://www.renesas.com)

Inquiries
   [http://www.renesas.com/contact/](http://www.renesas.com/contact/)
## Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Mar. 9, 2018</td>
<td>—</td>
<td>First edition issued</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. 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 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 a product with a different part number, confirm that the change will not lead to problems.
   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.
Notice

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

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other disputes 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, drawing, chart, program, algorithm, 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 not alter, modify, copy, or otherwise misappropriate 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, copy or otherwise misappropriation of Renesas Electronics products.

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

   - "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment, and industrial robots etc.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

6. When using the 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 minimum ratings, operating power supply voltage range, heat radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond such specified ranges.

7. 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. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of 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 by yourself as warranty for your product/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate the safety of the final products or systems manufactured by you.

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

9. 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 not use Renesas Electronics products for any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.

10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.

11. This document shall not be reprinted, 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.

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

(Note 2) "Renesas Electronics products" means any product developed or manufactured by or for Renesas Electronics.

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

Renesas Electronics India Pvt. Ltd.
No. 777, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-57270090, Fax: +91-80-57267717
Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-6141

© 2017 Renesas Electronics Corporation. All rights reserved.
Colophon 6.0