Introduction

In this application note, a method for automatically adjusting the brightness of an LED ceiling light is explained. In this application system, a phototransistor is used to measure the illuminance, and the timer KB0 of the RL78/G11 is used to control the amount of light from the LED.

Target Device

RL78/G11

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

1. Specifications ................................................................. 4
   1.1 Phototransistor ........................................................ 4

2. Operation Check Conditions .................................................. 5

3. Related Application Notes ..................................................... 5

4. Description of the Hardware ................................................... 6
   4.1 Hardware Configuration Example ................................. 6
   4.2 List of Pins to be Used .............................................. 6

5. Description of the Software .................................................... 7
   5.1 Operation Outline ...................................................... 7
   5.2 List of Option Byte Settings ......................................... 9
   5.3 List of Constants ....................................................... 9
   5.4 List of Variables ....................................................... 9
   5.5 List of Functions ....................................................... 10
   5.6 Function Specifications ............................................. 10
   5.7 Flowcharts ............................................................... 14
      5.7.1 Overall Flow ....................................................... 14
      5.7.2 Initialization Function ........................................... 14
      5.7.3 System Function ................................................ 15
      5.7.4 CPU Clock Setup ............................................... 16
      5.7.5 16-bit timer KB0 initial Setting .............................. 17
      5.7.6 D/A Converter Initial Setting ................................. 18
      5.7.7 Programmable gain amplifier (PGA) Initial Setting ... 18
      5.7.8 Comparator 0 Initial Setting .................................... 19
      5.7.9 Timer Array Unit Initial Setting ............................... 20
      5.7.10 12-bit Interval Timer Initial Setting ....................... 21
      5.7.11 A/D Converter Initial Setting ................................. 22
      5.7.12 Initial Setting of External Interrupt ....................... 23
      5.7.13 Main Processing ............................................... 24
      5.7.14 Main Initialization Setting .................................... 25
      5.7.15 D/A converter Operation Startup ......................... 25
      5.7.16 Programmable gain amplifier Operation Startup .... 25
      5.7.17 Comparator 0 Operation Startup ............................ 26
      5.7.18 INTP1 External Interrupt Enabling Function ........... 26
      5.7.19 Timer Array Unit Operation Startup ..................... 27
      5.7.20 16-bit timer KB0 Operation Startup ...................... 27
      5.7.21 INTP1 External Interrupt Function ....................... 28
      5.7.22 16-bit timer KB0 Operation Stop .......................... 29
      5.7.23 Timer Array Unit Interrupt Processing .................. 30
      5.7.24 A/D Converter Operation Startup ....................... 31
      5.7.25 A/D Conversion Result Acquisition ...................... 31
      5.7.26 A/D Converter Operation Stop ............................ 32
6. Sample Code ............................................................................................................. 33
7. Documents for Reference ......................................................................................... 33
1. Specifications

In this application system, a phototransistor is used to measure the illuminance, and the timer KB0 of the RL78/G11 is used to hold the amount of light from the LED constant.

Figure 1.1 shows the schematic system configuration.

![Figure 1.1 Schematic System Configuration](image)

When the ceiling light power is turned on, after initial setting of the RL78/G11, the LED is lit by the PWM output from the 16-bit timer KB0 (hereafter "timer KB0"). A programmable gain amplifier (hereafter "PGA") and an A/D converter are used to measure the current flowing to the LED. At the same time, the brightness is measured within the range detectable by a phototransistor mounted in the LED ceiling light.

In this application system, the brightness is maintained at 300 lx, which is optimal for a conference room. The duty ratio of the PWM output of the timer KB0 is controlled to adjust the current flowing to the LED such that the brightness is 300 lx (feedback control).

In this application system, simple PWM output control is used. In actuality, PI control or the like should be used according to the circuit and the device specifications. For information on PI control, refer to "LED Control Using RL78/I1A (R01AN1087) ".

1.1 Phototransistor

This application system uses a phototransistor. When actually preparing the circuit, the circuit should be designed so that the electrical characteristics of the phototransistor are satisfied.
2. Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed in the table below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller used</td>
<td>RL78/G11(R5F1058A)</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>High-speed on-chip oscillator (HOCO) clock: 24MHz</td>
</tr>
<tr>
<td></td>
<td>CPU/peripheral hardware clock: 24MHz / 48MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5.0 V (Operation is possible over a voltage range of 2.7 V to 5.5 V.)</td>
</tr>
<tr>
<td></td>
<td>LVD operation (V_{LVD}): Reset mode which uses 2.81 V (2.76 V to 2.87 V)</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>CS+ V7.00.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>(CS+)</td>
<td></td>
</tr>
<tr>
<td>C compiler (CS+)</td>
<td>CC-RL V1.07.00 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>e2 studio V7.1.0 from Renesas Electronics Corp.</td>
</tr>
<tr>
<td>(e2 studio)</td>
<td></td>
</tr>
<tr>
<td>C compiler (e2 studio)</td>
<td>CC-RL V1.07.00 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/G13 Initialization CC-RL (R01AN2575) Application Note
RL78/G11 LED Control by Using 16-Bit Timer KB0 CC-RL (R01AN4109EJ)
LED Control Using RL78/I1A (R01AN1087EJ)
4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used for this application note.

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

3. VDD must be held at not lower than the reset release voltage (VLVD) that is specified as LVD.

4.2 List of Pins to be Used

Table 4.1 lists the pins to be used and their functions.

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P31/TKBO0</td>
<td>Output</td>
<td>Timer KBO0 output port</td>
</tr>
<tr>
<td>P33/ANI18</td>
<td>Input</td>
<td>Phototransistor analog input port</td>
</tr>
<tr>
<td>P22/PGAI</td>
<td>Input</td>
<td>Analog input port for LED feedback resistor</td>
</tr>
<tr>
<td>P122/INTP1</td>
<td>Input</td>
<td>Switch input port</td>
</tr>
</tbody>
</table>
5. Description of the Software

5.1 Operation Outline

In this application note, when power to the ceiling light is turned on, RL78/G11 initial settings (functions set: timer KB0, A/D converter, D/A converter, PGA, comparator, timer array unit, external interrupt) are executed.

Details of the sample code initial settings are as follows.

1. Timer KB0 initial settings
   - The timer KB0 is set to standalone mode.
   - As the active level of the timer output TKBO0, "high level" is set, and as the default level, "low level" is set.
   - The PWM period is set to 4 μs, and the duty ratio is set to 0%.
   - A TKBO0 forced output stop function 1 is set, and low level fixed output is selected. Comparator 0 is selected as the trigger for the function 1. Type 1 is selected as the operating mode.

2. A/D converter initial settings
   - A/D voltage comparator operation is enabled.
   - Resolution is set to 10 bits.
   - VDD is selected for VREF(+), and VSS is selected for VREF(-).
   - Software trigger mode, one-shot conversion mode, and select mode are set.
   - ANI18 is set as an analog input pin, and the conversion start channel is set to ANI18.
   - Conversion time is set to standard 1 mode and 95/\(f_{\text{CLK}}\) (3.9583 μs).
   - Generation of an interrupt signal (INTAD) upon ADLL ≤ ADCR ≤ ADUL is selected, with the upper limit set to ADUL = 255 and the lower limit to ADLL = 0.

3. D/A converter initial settings
   - The D/A converter 0 conversion value is set to 163.
   - When the LED overcurrent is 400 mA, PGA is set to 8x, and the current detection resistance is 1 Ω, the reference voltage for overcurrent detection is
     \[ 400 \text{ mA} \times 8 \times 1 \Omega = 3.2 \text{ V} \]
   To generate the overcurrent detection reference voltage from the D/A converter analog output,
   \[ \text{analog output voltage } V_{\text{ANOi}} = \frac{V_{\text{DD}} \times (DACSi)}{256} \]
   \[ V_{\text{ANOi}} = 3.2 \text{ V}, \ V_{\text{DD}} = 5 \text{ V}, \ \text{and so } DACSi = 163 \]

4. Programmable gain amplifier (PGA) initial settings
   - PGAGND is selected as the PGA GND.
   - The PGA amplification factor is set to 8.
5. Comparator initial settings
   - The comparator speed is set to high-speed mode.
   - Standard mode is selected.
   - The input signal of the + pin of the comparator 0 is set to the output of the programmable gain amplifier.
   - The input signal of the - pin of the comparator 0 is set to the output of channel 0 of the internal D/A converter.
   - Interrupt request by comparator 0 single-edge detection is selected.
   - Interrupt request at the comparator 0 rising edge is selected.
   - A comparator 0 filter is set with sampling at f_{CLK} (f_{CLK} = f_{IH} = 24 MHz).
   - Comparator 0 interrupts are enabled, and priority level is set to 3.

6. Timer array unit 0 (TAU0) initial settings
   - Channel 0 is set to the interval timer.
   - The interval time is set to 300 μs.
   - TAU0 channel 0 count end interrupts are enabled, and the priority level is set to 3.

7. 12-bit Interval Timer initial settings
   - The interval time is set to 50 ms.

8. External interrupts initial settings
   - The P122/INTP1 pin is used.
   - The valid edge of the INTP1 pin is set to the falling edge.

After completion of initial settings, the RL78/G11 makes a transition to HALT mode. The RL78/G11 returns from HALT mode with each interrupt factor. Operation at the time of occurrence of different interrupt factors is indicated below.

- TAU0 channel 0 count end interrupt
  The LED current is acquired using the PGA and the A/D converter. The illuminance of the phototransistor is acquired using the A/D converter. The acquired values are then used in simple feedback control. The active level width of the PWM output of the timer KB0 is changed such that the brightness in the environs of the LED lighting fixture is 300 lx.

- INTP1 interrupt
  Switch on/off states are detected. When the switch is detected to be turned off, output from the timer KB0 is stopped. When the switch is detected to be turned on, output from the timer KB0 is started.

When an overcurrent is detected during LED current measurement, the forced output stop function of the timer KB0 is used to stop PWM output. In this application note, the forced output stop function is not canceled.
5.2 List of Option Byte Settings

Table 5.1 summarizes the settings of the option bytes.

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000C0H/010C0H</td>
<td>01101110B</td>
<td>Disables the watchdog timer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Stops counting after the release from the reset state.)</td>
</tr>
<tr>
<td>000C1H/010C1H</td>
<td>01111111B</td>
<td>LVD reset mode, 2.81 V (2.76 V to 2.87 V)</td>
</tr>
<tr>
<td>000C2H/010C2H</td>
<td>11110000B</td>
<td>HS mode, ( f_{\text{IH}} ): 24MHz, ( f_{\text{HOCO}} ): 48MHz</td>
</tr>
<tr>
<td>000C3H/010C3H</td>
<td>10000100B</td>
<td>Enables the on-chip debugger.</td>
</tr>
</tbody>
</table>

5.3 List of Constants

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

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>0</td>
<td>Flag setting value (OFF)</td>
</tr>
<tr>
<td>ON</td>
<td>1</td>
<td>Flag setting value (ON)</td>
</tr>
<tr>
<td>TARGET_LUX</td>
<td>0x012C</td>
<td>Lighting target value</td>
</tr>
<tr>
<td>TARGET_LED</td>
<td>0x0047</td>
<td>LED current target value</td>
</tr>
</tbody>
</table>

5.4 List of Variables

Table 5.3 lists the global variables that are used in the sample program.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint16_t</td>
<td>g_result_buffer</td>
<td>Stores conversion results of A/D converter.</td>
<td>main, ( r_{\text{tau0}} \text{-channel0_interrupt} )</td>
</tr>
<tr>
<td>uint16_t</td>
<td>g_pwm_duty</td>
<td>Stores duty value of timer KB0.</td>
<td>main, ( r_{\text{tau0}} \text{-channel0_interrupt} )</td>
</tr>
<tr>
<td>uint8_t</td>
<td>g_switch_flag</td>
<td>Stores switch status.</td>
<td>main, ( r_{\text{tau0}} \text{-channel0_interrupt} )</td>
</tr>
<tr>
<td>unsigned short int</td>
<td>g_fb_led_value</td>
<td>Stores LED current conversion result.</td>
<td>main, ( r_{\text{tau0}} \text{-channel0_interrupt} )</td>
</tr>
<tr>
<td>unsigned short int</td>
<td>g_fb_led_value_old</td>
<td>Stores previous LED current conversion result.</td>
<td>main, ( r_{\text{tau0}} \text{-channel0_interrupt} )</td>
</tr>
<tr>
<td>unsigned short int</td>
<td>g_get_lux</td>
<td>Stores illuminance acquisition result.</td>
<td>main, ( r_{\text{tau0}} \text{-channel0_interrupt} )</td>
</tr>
</tbody>
</table>
5.5 List of 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 function</td>
</tr>
<tr>
<td>R_MAIN_UserInit</td>
<td>Main user initialization function</td>
</tr>
<tr>
<td>R_DAC0_Start</td>
<td>DAC0 operation start</td>
</tr>
<tr>
<td>R_PGA_Start</td>
<td>PGA operation start</td>
</tr>
<tr>
<td>R_COMP0_Start</td>
<td>Comparator 0 operation start</td>
</tr>
<tr>
<td>R_INTC1_Start</td>
<td>Enables INTP1 external interrupt</td>
</tr>
<tr>
<td>R_TAU0_Channel0_Start</td>
<td>Timer array unit channel 0 start processing</td>
</tr>
<tr>
<td>R_TMR_KB0_Start</td>
<td>16-bit timer KB0 operation start</td>
</tr>
<tr>
<td>R_TMR_KB0_Stop</td>
<td>16-bit timer KB0 operation stop</td>
</tr>
<tr>
<td>r_tau0_channel0_interrupt</td>
<td>Timer array unit channel 0 interrupt processing</td>
</tr>
<tr>
<td>R_ADC_Start</td>
<td>A/D Converter operation start</td>
</tr>
<tr>
<td>R_ADC_Get_Result</td>
<td>Get A/D conversion results.</td>
</tr>
<tr>
<td>R_ADC_Stop</td>
<td>A/D Converter operation stop</td>
</tr>
<tr>
<td>r_intc1_interrupt</td>
<td>Processes INTP1 external interrupt</td>
</tr>
<tr>
<td>R_IT_Start</td>
<td>12-bit interval timer operation start</td>
</tr>
<tr>
<td>R_IT_Stop</td>
<td>12-bit interval timer operation stop</td>
</tr>
</tbody>
</table>

5.6 Function Specifications

This section describes the specifications for the functions that are used in this sample program.

[Function Name] main

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Main function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrondriver.h, r_cg_cgc.h, r_cg_tau.h, r_cg_it.h, r_cg_tmkb.h, r_cg_adc.h, r_cg_dac.h, r_cg_pga.h, r_cg_comp.h, r_cg_intp.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>-</td>
</tr>
<tr>
<td>Explanation</td>
<td>After execution of the main user initialization function, enables interrupts, makes a transition to HALT mode, and waits for the occurrence of an interrupt.</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>
### R_MAIN_UserInit

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Synopsis</th>
<th>Header</th>
<th>Declaration</th>
<th>Explanation</th>
<th>Arguments</th>
<th>Return value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_MAIN_UserInit</td>
<td>Main user initialization function</td>
<td>r_cg_macrodriver.h, r_cg_cgc.h, r_cg_tau.h, r_cg_it.h, r_cg_tmkb.h, r_cg_adc.h, r_cg_dac.h, r_cg_pga.h, r_cg_comp.h, r_cg_intp.h, r_cg_userdefine.h</td>
<td>static void R_MAIN_UserInit(void);</td>
<td>Enables interrupts by the EI instruction.</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

### R_DAC0_Start

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Synopsis</th>
<th>Header</th>
<th>Declaration</th>
<th>Explanation</th>
<th>Arguments</th>
<th>Return value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_DAC0_Start</td>
<td>Starts the D/A converter operation.</td>
<td>r_cg_macrodriver.h, r_cg_dac.h, r_cg_userdefine.h</td>
<td>void R_DAC0_Start(void);</td>
<td>Starts the D/A converter operation.</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

### R_PGA_Start

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Synopsis</th>
<th>Header</th>
<th>Declaration</th>
<th>Explanation</th>
<th>Arguments</th>
<th>Return value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_PGA_Start</td>
<td>Starts the PGA operation.</td>
<td>r_cg_macrodriver.h, r_cg_comp.h, r_cg_userdefine.h</td>
<td>void R_PGA_Start(void);</td>
<td>Starts the PGA operation.</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

### R_COMP0_Start

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Synopsis</th>
<th>Header</th>
<th>Declaration</th>
<th>Explanation</th>
<th>Arguments</th>
<th>Return value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_COMP0_Start</td>
<td>Starts the Comparator 0 operation.</td>
<td>r_cg_macrodriver.h, r_cg_comp.h, r_cg_userdefine.h</td>
<td>void R_COMP0_Start(void);</td>
<td>Starts the Comparator 0 operation.</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

### R_INTC1_Start

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Synopsis</th>
<th>Header</th>
<th>Declaration</th>
<th>Explanation</th>
<th>Arguments</th>
<th>Return value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_INTC1_Start</td>
<td>Enables INTP1 external interrupt.</td>
<td>r_cg_macrodriver.h, r_cg_intp.h, r_cg_userdefine.h</td>
<td>void R_INTC1_Start(void);</td>
<td>Enables INTP1 external interrupt.</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
### [Function Name] R_TAU0_Channel0_Start

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Timer array unit channel 0 starts count operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_tau.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_TAU0_Channel0_Start(void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Timer array unit channel 0 starts count operation.</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_TMR_KB0_Start

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Starts the 16-bit timer KB0 operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_tmkb.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_TMR_KB0_Start(void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Starts the 16-bit timer KB0 operation.</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_TMR_KB0_Stop

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Stops the 16-bit timer KB0 operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_tmkb.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_TMR_KB0_Stop(void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Stops the 16-bit timer KB0 operation.</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_tau0_channel0_interrupt

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Timer array unit channel 0 interrupt processing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_tmkb.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void __near r_tau0_channel0_interrupt(void)</td>
</tr>
<tr>
<td>Explanation</td>
<td>Timer array unit channel 0 interrupt processing.</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_ADC_Start

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Starts A/D conversion processing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_adc.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_ADC_Start (void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Starts A/D conversion processing.</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_ADC_Get_Result

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Get A/D conversion results.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_adc.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_ADC_Get_Result(void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Get A/D conversion results.</td>
</tr>
<tr>
<td>Arguments</td>
<td>uint16_t * const buffer</td>
</tr>
<tr>
<td>Return value</td>
<td>None</td>
</tr>
<tr>
<td>Remarks</td>
<td>None</td>
</tr>
</tbody>
</table>

### [Function Name] R_ADC_Stop

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Stops A/D conversion processing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_adc.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_ADC_Stop (void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Stops A/D conversion processing.</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_intc1_interrupt

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Processes INTP1 external interrupt.</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>static void __near r_intc1_interrupt(void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Processes INTP1 external interrupt.</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

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Starts 12-bit interval timer operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_it.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_IT_Start(void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Starts 12-bit interval timer operation.</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_Stop

<table>
<thead>
<tr>
<th>Synopsis</th>
<th>Stops 12-bit interval timer operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>r_cg_macrodriver.h, r_cg_it.h, r_cg_userdefine.h</td>
</tr>
<tr>
<td>Declaration</td>
<td>void R_IT_Stop(void);</td>
</tr>
<tr>
<td>Explanation</td>
<td>Stops 12-bit interval timer operation.</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>
5.7 Flowcharts

5.7.1 Overall Flow

Figure 5.1 shows the overall flow of the sample program described in this application note.

![Overall Flow Flowchart](image)

**Figure 5.1 Overall Flow**

5.7.2 Initialization Function

Figure 5.2 shows the flowchart for the initialization function.

![Initialization Function Flowchart](image)

**Figure 5.2 Initialization Function**
5.7.3 System Function

Figure 5.3 shows the flowchart for the system function.

```plaintext
R_Systeminit()

Disuse peripheral I/O redirection function

Set up CPU clock
R_CGC_Create()

Set up 16-bit Timer KB0
R_TMR_KB_Create()

Set up D/A converter
R_DAC0_Create()

Set up PGA
R_PGA_Create()

Set up Comparator
R_COMP_Create()

Set up Timer Array Unit
R_TAU0_Create()

Set up 12-bit interval timer
R_IT_Create()

Set up External interrupt
R_INTC_Create()

return

PIOR0 register ← 00H
PIOR1 register ← 00H
PIOR2 register ← 00H
PIOR3 register ← 00H
```

Figure 5.3 System Function
5.7.4 CPU Clock Setup

Figure 5.4 shows the flowchart for setting up the CPU clock.

![Flowchart](image)

**Figure 5.4 CPU Clock Setup**

Caution: For details on the procedure for setting up the CPU clock (R_CGC_Create()), refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575E).
5.7.5 16-bit timer KB0 initial Setting

Figure 5.5 shows the flowchart for the initial setting of the 16-bit timer KB0.

```
R_TMR_KB0_Create()

16-bit timer KB0 reset
16-bit timer KB0 reset release

Enable clock supply to 16-bit timer KB0

Operation clock selection

16-bit timer KB0 interrupt processing disabled
Clear 16-bit timer KB0 interrupt request signal

Interrupt prioritization of INTTMKB0

16-bit timer KB0 output control setting

16-bit timer KB0 compare value setting

16-bit timer KB0 compare value setting

Forced output stop function setting
- Forced output stop function 1: Low level output
- Forced output stop function 2: Low level output
- Forced output stop function 1 Trigger: Comparator 0
- Forced output stop function 2 Trigger: None

TKBO0 pin setting

return
```

Figure 5.5 16-bit timer KB0 Initial Setting
5.7.6 D/A Converter Initial Setting
Figure 5.6 shows the flowchart for the initial setting of the D/A Converter.

![Figure 5.6 D/A Converter Initial Setting](image)

5.7.7 Programmable gain amplifier (PGA) Initial Setting
Figure 5.7 shows the flowchart for the initial setting of the Programmable gain amplifier (PGA).

![Figure 5.7 Programmable gain amplifier (PGA) Initial Setting](image)
Comparator 0 Initial Setting

Figure 5.8 shows the flowchart for the initial setting of the comparator 0.

R_COMP_Create()

Comparator Reset
Comparator reset release

Clock supply to Comparator

Clock supply to PGA

Comparator 0, Comparator 1
Operation stop

Comparator 0 interrupt processing disabled
Comparator 0 interrupt request signal clear

Comparator 1 interrupt processing disabled
Comparator 1 interrupt request signal clear

Comparator input signal selection

Comparator mode setting

Comparator speed selection

Comparator Filter control setting

Comparator 0 interrupt request setting

Interrupt priority setting of INTCMP0

return

CMPRES bit ← 1
CMPRES bit ← 0

CMPEN bit ← 1

PGA0EN bit ← 1

C0ENB bit ← 0
C1ENB bit ← 0

CMPMK0 bit ← 1
CMPIF0 bit ← 0

CMPMK1 bit ← 1
CMPIF1 bit ← 0

COMP0SEL1,COMP0SEL0 ← 10B : Select output of programmable gain amplifier
C0REFSEL1,C0REFSEL0 ← 10B : Select output of channel 0 of the on-chip D/A converter

COMPMDR register ← 00H
C0MON bit ← 0 : standard mode
C0VRF bit ← 0 : The reference voltage for comparator 0 is the voltage specified in COMPISEL.C0REFSEL.

SPDMD bit ← 1 : Comparator high-speed mode

COMPFIR register ← 01H
C0EDG bit ← 0 : Interrupt request by comparator 1 one-edge detection
C0EPO bit ← 0 : Interrupt request at comparator 1 rising edge
C0FCK1 bit ← 0 : sampling at fCLK
C0FCK0 bit ← 1

C0IE bit ← 1 : Comparator 0 interrupt request enabled

CMPPR00 bit ← 1
CMPPR10 bit ← 1

Figure 5.8 Comparator 0 Initial Setting
5.7.9 Timer Array Unit Initial Setting

Figure 5.9 shows the flowchart for the initial setting of the timer Array Unit.

```
R_TAU0_Create()

Timer Array unit reset
Timer Array unit reset release

Clock supply to Timer Array unit

Operation setting of TAU 0
Set operation clock of TAU 0
Operation clock 0 (CK00): 24 MH

TAU 0 operation stop

Disable TAU interrupt
Clear TAU interrupt request signal

Interrupt priority setting of INTI00

Initialize TAU0 channel 0
- Operation clock: CK00
- Operation mode: Interval timer mode
- Functional feature: Independent channel operation function
- Start: Software trigger start
- Interval timer: 300 us
- Initial channel 0 output value: 0
- Timer output is disabled

return
```

TAU0RES bit ← 1
TAU0RES bit ← 0

TAU0EN bit ← 1

TPS0 register ← 0000H

TT0 register ← 0A0FH

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

TMPR100 bit ← 1 : Level 3
TMPR000 bit ← 1

TMR00 register ← 0000H
TDR00 register ← 1C1FH
TO0 register ← 0000H
TOE0 register ← 0000H

Figure 5.9 Timer Array Unit Initial Setting
5.7.10 12-bit Interval Timer Initial Setting

Figure 5.10 shows the flowchart for the initial setting of the 12-bit Interval Timer.

![Flowchart of 12-bit Interval Timer Initial Setting]

- **R_IT_Create()**: Initialize 12-bit interval timer
  - Operation clock: \( f_{IL} \) (15kHz)
  - Interval timer: 50ms

- **Clock supply to 12-bit interval timer**
  - ITMC register ← 00H

- **12-bit interval timer operation stop**
  - TMKAEN bit ← 1

- **Disable 12-bit interval timer interrupt**
  - TMKAMK bit ← 1

- **Clear 12-bit interval timer interrupt request signal**
  - TMKAIF bit ← 0

- **Interrupt priority setting of INTIT (Level 3)**
  - TMKAPR1 bit ← 1
  - TMKAPR0 bit ← 1

- **12-bit interval Timer reset**
  - TMKRES bit ← 1
  - TMKRES bit ← 0

- **12-bit interval Timer reset release**
  - TPS3 = 00H
  - ITMC register ← 02EDH

**Figure 5.10** 12-bit Interval Timer Initial Setting
Figure 5.11 shows the flowchart for the initial setting of the A/D Converter.

```plaintext
R_ADC_Create()

A/D converter reset
A/D converter reset release

Clock supply to A/D converter

Stop operation of A/D converter

Disable A/D conversion end interrupt

Clear A/D conversion end interrupt request flag

Interrupt priority setting of INTAD

ANI3, ANI18 set to analog input

Setting of A/D converter mode register 0
- Mode: Normal 1
- Conversion Time: 3.96μs
- Select mode

Setting of A/D converter mode register 1
- Software trigger mode
- One-shot conversion mode

Setting of A/D converter mode register 2
- Reference voltage
  - +side: Supplied from VDD
  - -side: Supplied from VSS
- Checking the upper limit and lower limit conversion result
- 10-bit resolution

Setting comparison result upper limit / lower limit value

Specify analog input channel
- Set the analog input channel to ANI18

Start A/D voltage comparator operation

return

A/D converter reset
A/D converter reset release

Clock supply to A/D converter

Stop operation of A/D converter

Disable A/D conversion end interrupt

Clear A/D conversion end interrupt request flag

Interrupt priority setting of INTAD

ANI3, ANI18 set to analog input

Setting of A/D converter mode register 0
- Mode: Normal 1
- Conversion Time: 3.96μs
- Select mode

Setting of A/D converter mode register 1
- Software trigger mode
- One-shot conversion mode

Setting of A/D converter mode register 2
- Reference voltage
  - +side: Supplied from VDD
  - -side: Supplied from VSS
- Checking the upper limit and lower limit conversion result
- 10-bit resolution

Setting comparison result upper limit / lower limit value

Specify analog input channel
- Set the analog input channel to ANI18

Start A/D voltage comparator operation

return

A/D converter reset
A/D converter reset release

Clock supply to A/D converter

Stop operation of A/D converter

Disable A/D conversion end interrupt

Clear A/D conversion end interrupt request flag

Interrupt priority setting of INTAD

ANI3, ANI18 set to analog input

Setting of A/D converter mode register 0
- Mode: Normal 1
- Conversion Time: 3.96μs
- Select mode

Setting of A/D converter mode register 1
- Software trigger mode
- One-shot conversion mode

Setting of A/D converter mode register 2
- Reference voltage
  - +side: Supplied from VDD
  - -side: Supplied from VSS
- Checking the upper limit and lower limit conversion result
- 10-bit resolution

Setting comparison result upper limit / lower limit value

Specify analog input channel
- Set the analog input channel to ANI18

Start A/D voltage comparator operation

return

A/D converter reset
A/D converter reset release

Clock supply to A/D converter

Stop operation of A/D converter

Disable A/D conversion end interrupt

Clear A/D conversion end interrupt request flag

Interrupt priority setting of INTAD

ANI3, ANI18 set to analog input

Setting of A/D converter mode register 0
- Mode: Normal 1
- Conversion Time: 3.96μs
- Select mode

Setting of A/D converter mode register 1
- Software trigger mode
- One-shot conversion mode

Setting of A/D converter mode register 2
- Reference voltage
  - +side: Supplied from VDD
  - -side: Supplied from VSS
- Checking the upper limit and lower limit conversion result
- 10-bit resolution

Setting comparison result upper limit / lower limit value

Specify analog input channel
- Set the analog input channel to ANI18

Start A/D voltage comparator operation

return
```
5.7.12 Initial Setting of External Interrupt

Figure 5.12 shows the flowchart for the initial setting of the external interrupt.

```
R_INTC_Create()

DISABLE INTP1 interrupt
Clear INTP1 interrupt request flag

Set INTP1 interrupt priority to level 3 (lowest)

Set effective edge for external interrupt

return
```

INTP1
PMK1 bit ← 1
PIF1 bit ← 0

INTP1
PPR11 bit ← 1
PPR01 bit ← 1

EGN0 register ← 02H
EGP1 register ← 00H
Set falling edge for INTP1 pin

Figure 5.12 Initial Setting of External Interrupt
5.7.13 Main Processing

Figure 5.13 shows the flowchart for main processing.

```
main()

Main initial setting
R_MAIN_UserInit()

Initialize global variables

g_switch_flag ← ON

D/A Converter operation Start
R_DAC0_Start()

Programmable gain amplifier operation Start
R_PGA_Start()

Comparator 0 operation Start
R_COMP0_Start()

INTP1 External Interrupt operation start
R_INTC1_Start()

TAU operation start
R_TAU0_Channel0_Start()

Timer KB0 operation start
R_TMR_KB0_Start()

Shift to HALT mode

TAU interrupt

INTP1 interrupt
```

Figure 5.13 Main Processing
5.7.14 Main Initialization Setting
Figure 5.14 shows the flowchart for the main initialization settings.

![Figure 5.14 Main Initialization Setting](image)

5.7.15 D/A converter Operation Startup
Figure 5.15 shows the flowchart for starting the D/A converter operation.

![Figure 5.15 D/A converter Operation Startup](image)

5.7.16 Programmable gain amplifier Operation Startup
Figure 5.16 shows the flowchart for starting the Programmable gain amplifier Operation.

![Figure 5.16 Programmable gain amplifier Operation Startup](image)
### 5.7.17 Comparator 0 Operation Startup

Figure 5.17 shows the flowchart for starting the Comparator 0 Operation.

![Flowchart](image)

R_COMP0_Start()

Start Comparator 0 operation

Comparator 0 stabilization time : 100us

Enable comparator 0 interrupt
Comparator 0 interrupt request flag clear

return

C0ENB bit ← 1

CMPMK0 bit ← 0
CMPIF0 bit ← 0

---

### 5.7.18 INTP1 External Interrupt Enabling Function

Figure 5.18 shows the flowchart for the INTP1 external interrupt enabling function.

![Flowchart](image)

R_INTC1_Start()

Enable external interrupt of INTP1

return

INTP1
PMK1 bit ← 0
PIF1 bit ← 0
5.7.19 Timer Array Unit Operation Startup

Figure 5.19 shows the flowchart for starting the Timer Array Unit Operation.

```
R_TAU0_Channel0_Start()

INTTM00 interrupt enable
INTTM00 interrupt request flag clear

Enable Timer Array Unit channel 0 operation

return

TMMK00 bit ← 0
TMIF00 bit ← 0
TS0 register ← 01H
```

Figure 5.19 Timer Array Unit Operation Startup

5.7.20 16-bit timer KB0 Operation Startup

Figure 5.20 shows the flowchart for starting the 16-bit timer KB0 Operation.

```
R_TMR_KB0_Start()

Enable timer KB0 output

Enable timer KB0 operation

return

TKBIOC01 register ← 01H
TKBCE0 bit ← 1
```

Figure 5.20 16-bit timer KB0 Operation Startup
5.7.21 INTP1 External Interrupt Function

Figure 5.21 shows the flowchart for INTP1 external interrupt function.

![Flowchart of INTP1 External Interrupt Function]

- Start 12-bit interval timer operation
  - R_IT_Start()

- Shift to HALT mode

- Is 12-bit interval timer interrupt generated?
  - Yes
    - Stop 12-bit interval timer operation
      - R_IT_Stop()

  - No
    - Is P122 in low level?
      - Yes
        - Change switch status to OFF
          - g_switch_flag ← OFF
          - Set timer KB0 compare value to 0000H
            - TKBCR01 register ← 0000H
          - Stop timer KB0 operation
            - R_TMR_KB0_Stop()
        - Change switch status to ON
          - g_switch_flag ← ON
          - Start timer KB0 operation
            - R_TMR_KB0_Start()

- No
  - Is switch turned ON or OFF?
    - OFF
    - Change switch status to OFF
      - g_switch_flag ← OFF
    - ON
    - Change switch status to ON
      - g_switch_flag ← ON

- return

**Figure 5.21 INTP1 External Interrupt Function**
5.7.22 16-bit timer KB0 Operation Stop

Figure 5.22 shows the flowchart for stopping the 16-bit timer KB0 Operation.

```
R_TMR_KB0_Stop()

Disable timer KB0 operation

Disable timer KB0 output

TKBCE0 bit ← 0

TKBIOC01 register ← 00H

return
```

Figure 5.22 16-bit timer KB0 Operation Stop
5.7.23 Timer Array Unit Interrupt Processing

Figure 5.23 shows the flowchart for alarm Timer Array Unit interrupt processing.

```
if (r_tau0_channel0_interrupt()) {
    Set analog input channel to ANI18
    Start A/D converter operation
    R_ADC_Start()
}

Shift to HALT mode

A/D conversion end interrupt occurred?

Yes

Store A/D conversion results in RAM
R_ADC_Get_Result()

Obtain illuminance (g_get_lux) from A/D conversion result

Set analog input channel to PGAOUT

Start A/D converter operation
R_ADC_Start()

Shift to HALT mode

A/D conversion end interrupt occurred?

No

Yes

Stop A/D converter operation
R_ADC_Stop()

Eliminate PGA amplification factor (8 times) from A/D conversion result and obtain LED current (g_fb_led_value)

g_fb_led_value ← g_result_buffer / 8

Store timer KB compare register value in RAM

g_pwm_duty ← TKBCR01 register

Provide feedback control

Update timer KB compare register by feedback result

TKBCR01 register ← g_pwm_duty

Rewrite timer KB compare register values simultaneously

TKBTRG0 bit ← 1

return
```

Figure 5.23 Timer Array Unit Interrupt Processing
5.7.24 A/D Converter Operation Startup

Figure 5.24 shows the flowchart for starting the A/D converter operation.

![Flowchart for A/D Converter Operation Startup](image)

5.7.25 A/D Conversion Result Acquisition

Figure 5.25 shows the flowchart for acquiring conversion results of the A/D conversion.

![Flowchart for A/D Conversion Result Acquisition](image)
5.7.26 A/D Converter Operation Stop

Figure 5.26 shows the flowchart for stopping the A/D Converter Operation.

Figure 5.26 A/D Converter Operation Stop
6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

RL78/G11 User's Manual: Hardware (R01UH0637E)
RL78 Family User's Manual: Software (R01US0015E)
(The latest versions of the documents are available on the Renesas Electronics Website.)
Website and Support

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

Inquiries
http://www.renesas.com/contact/

All trademarks and registered trademarks are the property of their respective owners.
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Jan. 28, 2019</td>
<td></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. 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.
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 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 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.

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 depend 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**: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly 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 intended or authorised for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.

6. 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.

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. 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 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. 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.

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 comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parts or transactions.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who 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.

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 directly or indirectly controlled subsidiaries.

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