To our customers,

---

**Old Company Name in Catalogs and Other Documents**

On April 1\(^{st}\), 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1\(^{st}\), 2010
Renesas Electronics Corporation

---

Issued by: Renesas Electronics Corporation ([http://www.renesas.com](http://www.renesas.com))

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.

2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.

4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.

5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.

6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

- “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.
- “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
- “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.

8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

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

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

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

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.
H8/300H Tiny Series
Ohm Meter Implementation Example

Introduction
In this example an ohm meter is implemented using an inverting op-amp. A reference resistor and the A/D converter built into the H8/36014 are used to measure the resistance value.

Target Device
H8/300H Tiny Series H8/36014

Contents
1. Specifications ............................................................................................................ 2
2. Functions Used ........................................................................................................... 6
3. Principles of Operation ............................................................................................ 9
4. Description of Software .......................................................................................... 11
5. Flowcharts .............................................................................................................. 16
6. Program Listing ........................................................................................................ 21
1. Specifications

(1) The hardware configuration used to measure resistance values is shown in figure 1.

(2) In this sample task the ohm meter comprises the reference resistor R1, an inverting op-amp, and the A/D converter built into the H8/36014.

(3) The output voltage from the op-amp is input to analog input pin 0 (the AN0 pin) of the A/D converter, which performs A/D conversion.

(4) The resistance value is obtained from the A/D conversion result by calculation and displayed as a decimal number on a 7-segment LED connected to the I/O ports.

(5) In this sample task the operating voltage (Vcc) and analog power supply voltage (AVcc) of the H8/36014 are 5 V, and the OSC clock frequency is 20 MHz, supplied as an external clock by a crystal oscillator.
(6) In this sample task the R1 resistance value has been determined so as to allow accurate measurement of resistance values up to 10 kΩ. The design procedure is described below.

(a) C-MOS rail-to-rail (some manufacturers use the term “full swing”) operational amplifier is a type of op-amp used as an inverting amplifier. It can produce an output amplitude up to the power supply voltage. Due to the relationship of the op-amp and inverting amplifier circuitry, the following relationship exists.

\[ \frac{E_s}{R_1} = -\frac{E_x}{R_x} \]

(Es: reference voltage, R1: reference resistance, Ex: op-amp output voltage, Rx: resistance being measured)

(b) Based on this relationship, the following equation can be used to calculate the resistance being measured.

\[ R_x = -\frac{R_1}{E_s} \times E_x \]

(c) To simplify the circuit configuration a common power supply voltage is used in this sample task for the op-amp and the H8/36014. Therefore, bias voltage Eb is applied to the non-inverted input pin (+) so that a signal is output from the negative side. Taking bias voltage Eb into account, the equation for calculating Rx changes to the following.

\[ R_x = -\frac{R_1}{(E_s - E_b)} \times (E_x - E_b) \]

(d) To obtain a measuring range up to 10 kΩ a value of 10 kΩ is assigned for reference resistor R1, 5 V for reference voltage Es, and 2.5 V for bias voltage Eb, resulting in the following.

\[ R_x = -\frac{10000}{(5 - 2.5)} \times (E_x - 2.5) \]

\[ = 10000 - 4000 \times E_x \]

Ex: op-amp output voltage

(e) The measuring range can be changed easily by changing reference resistor R1.

Example: Increasing R1 10 times or 100 times also increases the Rx measuring range 10 times or 100 times, respectively.

(7) The operations in this sample task are as follows.

(a) Connect the resistors to be measured (using probes, or the like) to Rx as shown in figure 1.

(b) When, for example, 10 kΩ (unknown) resistors are connected, the op-amp output will be 0 V due to the relationship between the inverting amplifier and the bias voltage. This output voltage is input to analog input pin 0 (the AN0 pin) of the A/D converter.

(c) Using the equation shown in step (d) above, the resistance being measured (Rx) is calculated as 10 kΩ.

(d) The resistance value calculated from the measurement results from the A/D converter is displayed using the 7-segment LEDs. A value of 10 kΩ is displayed as the decimal number 10.00.
Figure 2 7-Segment LED Control Method
(8) In this sample task the A/D conversion result is displayed using the 7-segment LEDs as a decimal number (00.00–99.99). The LED display for the A/D conversion result is shown in figure 3.

A/D conversion result (10-bit data): A/D data register A (ADDRA: 2 bytes)

<table>
<thead>
<tr>
<th>AD9</th>
<th>AD8</th>
<th>AD7</th>
<th>AD6</th>
<th>AD5</th>
<th>AD4</th>
<th>AD3</th>
<th>AD2</th>
<th>AD1</th>
<th>AD0</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>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Shift the contents of ADDRA 6 bits to the right and perform logical AND with H'03FF

0 0 0 0 0 0 AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0

Multiply by coefficient x

Add correction value y and store it in variable R

Actually, store the value in variable R after multiplying by 100 so that value is displayed to two decimal places

Coefficient x (float, 4 bytes)

= –(RH – RL) / (IH – IL)

Correction value y (float, 4 bytes)

= RL – IH * x

RH: Measured resistance value (upper limit)
RL: Measured resistance value (lower limit)
IH: AD input value (upper limit)
IL: AD input value (lower limit)

Example:
ADDRA = b'0100000000000000

((ADDRA >> 6 & 0x03ff) * x + y) * 100.0 → R (= 495)

LED1 display data

LED2 display data

LED3 display data

LED4 display data

Figure 3  LED Display for A/D Conversion Result
2. Functions Used

(1) A block diagram of the H8/36014 functions used in this sample task is shown in figure 4, and the assignment of functions is shown in table 1.

<table>
<thead>
<tr>
<th>Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer W</td>
<td>The free-running function of timer W is used to perform A/D conversion of the input to analog input pin 0 (AN0). The A/D conversion cycle is measured using the timer W overflow cycle of 32.768 ms.</td>
</tr>
<tr>
<td>Timer V</td>
<td>The free-running function of timer V is used to control switching of the 7-segment LEDs. Dynamic display is implemented by illuminating the four 7-segment LEDs sequentially each timer V overflow cycle of 2.048 ms.</td>
</tr>
<tr>
<td>A/D converter</td>
<td>The A/D converter performs A/D conversion of the output voltage of the inverting op-amp, and the resistance value is calculated from the detected voltage value.</td>
</tr>
<tr>
<td>Port 2</td>
<td>Switching of the four 7-segment LEDs is performed based on the output from port 2 output pins P20 and P21. Output pins P20 and P21 are connected to the input pins of the 2-to-4-line decoder.</td>
</tr>
<tr>
<td>Port 5</td>
<td>The display of the 7-segment LEDs is based on the output from port 5 output pins P50 to P57. The 10-bit data obtained by A/D conversion of the input to AN0 is converted to 3-digit hexadecimal data and output to the LEDs.</td>
</tr>
</tbody>
</table>
(2) A connection diagram for the 7-segment LEDs used is shown in figure 5. As shown in figure 5, a high-level signal output from port 5 causes the corresponding LED segment to light. Table 2 shows the correspondence between port 5 output and 7-segment LED display data.

![Figure 5 7-Segment LED Connection Diagram and Internal Wiring Diagram](image-url)
### Table 2  Correspondence Between Port 5 Output and 7-Segment LED Display Data

<table>
<thead>
<tr>
<th>LED Display</th>
<th>Port 5 Output Data</th>
<th>LED Display</th>
<th>Port 5 Output Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P57</td>
<td>P56</td>
<td>P55</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
3. Principles of Operation

(1) The principles of operation whereby A/D conversion of the input to the AN0 pin is performed using timer W are illustrated in figure 6. As shown in figure 6, in this sample task measurement of the A/D conversion cycle (0.49152 seconds) using the timer W overflow flag and determination of the end of A/D conversion in the tmrw routine are accomplished without using A/D converter interrupts.

![Diagram](image-url)

Note: In this application note the OVF flag of timer W is indicated as OVFW and the OVIE flag of timer W is indicated as OVIEW.

Figure 6 Operating Principle of Remote Control Reception Using Timer W
(2) The operation principles of 7-segment LED display control are shown below. Displaying “3210” using LED4 to LED1 is illustrated in figure 7. As shown in figure 7, dynamic 7-segment LED display is implemented by illuminating LED4 to LED1 sequentially each timer V overflow cycle.

Figure 7  Operating Principle of 7-Segment LED Display Control
4. Description of Software

4.1 Modules

Table 3 shows the modules used in this sample task.

Table 3 Modules

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main routine</td>
<td>main</td>
<td>Performs initial settings, enables interrupts</td>
</tr>
<tr>
<td>Timer W interrupt handling</td>
<td>tmrw</td>
<td>Clears interrupt flag, converts A/D conversion data to LED display data and stores it in RAM</td>
</tr>
<tr>
<td>Timer V interrupt handling</td>
<td>tmrv</td>
<td>Clears interrupt flag, controls LED display data output and LED display switching</td>
</tr>
</tbody>
</table>

4.2 Arguments

No arguments are used in this sample task.

4.3 Internal Registers Used

The internal registers used in this sample task are shown in table 4.

Table 4 Internal Registers Used

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Address</th>
<th>Set Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCRV0</td>
<td>Timer control register V0</td>
<td>H'FFA0</td>
<td>H'03</td>
</tr>
<tr>
<td></td>
<td>Selects the input clock signals of TCNTV, specifies the clearing conditions of TCNTV, and controls each interrupt request.</td>
<td></td>
<td>(initial setting)</td>
</tr>
<tr>
<td>CMIEB</td>
<td>Compare match interrupt enable B</td>
<td>Bit 7</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>When this bit is cleared to 0, interrupt requests indicated by the CMFB bit in TCSRV are disabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMIEA</td>
<td>Compare match interrupt enable A</td>
<td>Bit 6</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>When this bit is set to 0, interrupt requests indicated by the CMFA bit in TCSRV are disabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVIE</td>
<td>Timer overflow interrupt enable</td>
<td>Bit 5</td>
<td>0/1</td>
</tr>
<tr>
<td></td>
<td>When this bit is cleared to 0, interrupt requests indicated by the OVF bit in TCSRV are disabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>When this bit is set to 1, interrupt requests indicated by the OVF bit in TCSRV are enabled.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Address</td>
<td>Set Value</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>TCRV0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCLR1</td>
<td>Counter clear 1 and 0</td>
<td>Bit 4</td>
<td>0</td>
</tr>
<tr>
<td>CCLR0</td>
<td>These bits specify the clearing conditions of TCNTV. Clearing is disabled when CCLR1 = 0 and CCLR0 = 0.</td>
<td>Bit 3</td>
<td>0</td>
</tr>
<tr>
<td>CKS2</td>
<td>Clock select 2 to 0</td>
<td>Bit 2</td>
<td>0</td>
</tr>
<tr>
<td>CKS1</td>
<td>These bits select the clock signals input to TCNTV and the counting condition in combination with ICKS0 in TCRV1. When CKS2 = 0, CKS1 = 1, CKS0 = 1, and ICKS0 = 1, TCNTV counts on internal clock φ/128, falling edge.</td>
<td>Bit 1</td>
<td>1</td>
</tr>
<tr>
<td>CKS0</td>
<td></td>
<td>Bit 0</td>
<td>1</td>
</tr>
</tbody>
</table>

**TCSRV**

Timer control/status register V
Indicates the status flag and controls output using compare match.

| CMFB | Compare match flag B | Bit 7 | 0 |
| CMFA | Compare match flag A | Bit 6 | 0 |
| OVF  | Timer overflow flag  | Bit 5 | 0 |
| OS3  | Output select 3 and 2 | Bit 3 | 0 |
| OS2  | These bits set the output level of the TMOV pin by compare match B. No change when OS3 = 0 and OS2 = 0. | Bit 2 | 0 |
| OS1  | Output select 3 and 2 | Bit 1 | 0 |
| OS0  | These bits set the output level of the TMOV pin by compare match A. No change when OS1 = 0 and OS0 = 0. | Bit 0 | 0 |

**TCRV1**

Timer control register V1
Selects the edge at the TRGV pin, enables TRGV input, and selects the clock input to TCNTV.

| TVEG1 | TRGV input edge select 1 and 0 | Bit 4 | 0 |
| TVEG0 | These bits select the TRGV input edge. TRGV trigger input is disabled when TREG1 = 0 and TREG0 = 0. | Bit 3 | 0 |
| TRGE  | TRGV input enable | Bit 2 | 0 |

Enables or disables TCNTV counting-up by the input of the TRGV pin.
When TREG = 0, starting counting-up TCNTV by the input of the TRGV pin, and halting counting-up TCNTV when TCNTV is cleared by a compare match, are disabled.
<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Address</th>
<th>Set Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCRV1 ICKS0</td>
<td>Internal clock select 0</td>
<td>Bit 0</td>
<td>1</td>
</tr>
<tr>
<td>TCRW</td>
<td>Timer control register W</td>
<td>H'FF81</td>
<td>H'30</td>
</tr>
<tr>
<td>TMRW</td>
<td>Timer mode register W</td>
<td>H'FF80</td>
<td>H'C8</td>
</tr>
<tr>
<td>CTS</td>
<td>Counter start</td>
<td>Bit 7</td>
<td>1</td>
</tr>
<tr>
<td>TIERW</td>
<td>Timer interrupt enable register W</td>
<td>H'FF82</td>
<td>H'00 (initial setting)</td>
</tr>
<tr>
<td>OVIE</td>
<td>Timer overflow interrupt enable</td>
<td>Bit 7</td>
<td>0/1</td>
</tr>
<tr>
<td>TSRW</td>
<td>Timer status register W</td>
<td>H'FF83</td>
<td>H'00</td>
</tr>
<tr>
<td>OVF</td>
<td>Timer overflow</td>
<td>Bit 7</td>
<td>0</td>
</tr>
<tr>
<td>TCNT</td>
<td>Timer counter</td>
<td>H'FF86</td>
<td>H'00</td>
</tr>
<tr>
<td>ADCSR</td>
<td>A/D control/status register</td>
<td>H'FFB8</td>
<td>H'00 (initial setting)</td>
</tr>
<tr>
<td>ADF</td>
<td>A/D end flag</td>
<td>Bit 7</td>
<td>0</td>
</tr>
<tr>
<td>ADIE</td>
<td>A/D interrupt enable</td>
<td>Bit 6</td>
<td>0</td>
</tr>
</tbody>
</table>

This bit selects the clock signals input to TCNTV and the counting condition in combination with CKS2 to CKS0 in TCRV0. When CKS2 = 0, CKS1 = 1, CKS0 = 1, and ICKS0 = 1, TCNTV counts on internal clock $\phi/128$, falling edge.

When CKS2 = 0, CKS1 = 1, and CKS0 = 1, the TCNT input clock is set as 1/8 the system clock.

When OVIE = 0, interrupt requests by the OVF flag are disabled.

When OVIE = 1, interrupt requests by the OVF flag are enabled.

When OVF = 0 when TCNT has not overflowed.

OVF = 1 when TCNT overflows.

Set to 1 when A/D conversion ends in single mode.

Cleared to 0 when 0 is written after reading ADF = 1.

When this bit is cleared to 0, A/D conversion end interrupt requests are disabled.
<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Address</th>
<th>Set Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADCSR</td>
<td>ADST</td>
<td>A/D start</td>
<td>Bit 5</td>
</tr>
<tr>
<td></td>
<td>SCAN</td>
<td>Scan mode</td>
<td>Bit 4</td>
</tr>
<tr>
<td></td>
<td>CKS</td>
<td>Clock select</td>
<td>Bit 3</td>
</tr>
<tr>
<td></td>
<td>CH2</td>
<td>Channel select</td>
<td>Bit 2</td>
</tr>
<tr>
<td></td>
<td>CH1</td>
<td></td>
<td>Bit 1</td>
</tr>
<tr>
<td></td>
<td>CH0</td>
<td></td>
<td>Bit 0</td>
</tr>
<tr>
<td>ADCR</td>
<td></td>
<td>A/D control register</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRGE</td>
<td>Trigger enable</td>
<td>Bit 7</td>
</tr>
<tr>
<td>ADDR</td>
<td></td>
<td>A/D data register C</td>
<td></td>
</tr>
<tr>
<td>PCR2</td>
<td></td>
<td>Port control register 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PDR2</td>
<td>Port data register 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PMR5</td>
<td>Port mode register 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POF57</td>
<td>P57 pin function switch</td>
<td>Bit 7</td>
</tr>
<tr>
<td></td>
<td>POF56</td>
<td>P56 pin function switch</td>
<td>Bit 6</td>
</tr>
<tr>
<td></td>
<td>WKP5</td>
<td>P55/WKP5/ADTRG pin function switch</td>
<td>Bit 5</td>
</tr>
<tr>
<td></td>
<td>WKP4</td>
<td>P54/WKP4 pin function switch</td>
<td>Bit 4</td>
</tr>
</tbody>
</table>
### Register Descriptions

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
<th>Address</th>
<th>Set Value</th>
</tr>
</thead>
</table>
| PMR5 WKP3 | P53/WKP3 pin function switch
   When this bit is cleared to 0, P53 functions as a general I/O port. | Bit 3 | 0 |
| WKP2 | P52/WKP2 pin function switch
   When this bit is cleared to 0, P52 functions as a general I/O port. | Bit 2 | 0 |
| WKP1 | P51/WKP1 pin function switch
   When this bit is cleared to 0, P51 functions as a general I/O port. | Bit 1 | 0 |
| WKP0 | P50/WKP0 pin function switch
   When this bit is cleared to 0, P50 functions as a general I/O port. | Bit 0 | 0 |
| PUCR5 | Port pull-up control register 5
   Controls the pull-up MOS in bit units of the port 5 pins set as the input ports.
   When PUCR2 = H'00, pull-up MOS is off for pins P57 to P50. | H'FFD1 | H'00 |
| PDR5 | Port data register 5
   General I/O port data register of port 5. | H'FFD8 | H'00 |
| PCR5 | Port control register 5
   Selects input or output status in bit units for pins to be used as general I/O ports of port 5.
   When PCR5 = H'FF, pins P57 to P50 function as general output pins. | H'FFE8 | H'FF |

### RAM Usage

The RAM usage in this sample task is shown in table 5.

#### Table 5 RAM Usage

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>Address</th>
<th>Used in</th>
</tr>
</thead>
<tbody>
<tr>
<td>dig_0</td>
<td>Stores LED1 display data (1 byte)</td>
<td>H'FB8A</td>
<td>main, tmrw</td>
</tr>
<tr>
<td>dig_1</td>
<td>Stores LED2 display data (1 byte)</td>
<td>H'FB8B</td>
<td>main, tmrw</td>
</tr>
<tr>
<td>dig_3</td>
<td>Stores LED3 display data (1 byte)</td>
<td>H'FB8C</td>
<td>main, tmrw</td>
</tr>
<tr>
<td>dig_4</td>
<td>Stores LED4 display data (1 byte)</td>
<td>H'FB8D</td>
<td>main, tmrw</td>
</tr>
<tr>
<td>cnt</td>
<td>8-bit counter for display switching of LED1 to LED4 (1 byte)</td>
<td>H'FB8E</td>
<td>main, tmrw</td>
</tr>
<tr>
<td>counter_sub</td>
<td>8-bit counter for adjusting the A/D acquisition interval (1 byte)</td>
<td>H'FB8F</td>
<td>main, tmrw</td>
</tr>
<tr>
<td>R</td>
<td>For displaying the resistance value (2 bytes)</td>
<td>H'FB80</td>
<td>tmrw</td>
</tr>
<tr>
<td>x</td>
<td>Coefficient (4 bytes)</td>
<td>H'FB82</td>
<td>main, tmrw</td>
</tr>
<tr>
<td>y</td>
<td>Correction value (4 bytes)</td>
<td>H'FB86</td>
<td>main, tmrw</td>
</tr>
</tbody>
</table>
5. Flowcharts

5.1 Main Routine (main)

```
main

* Set CCR interrupt mask bit (I) to 1 to mask interrupt requests

CCR I-bit ← 1

dig_0 ← H'40

dig_1 ← H'40

dig_2 ← H'40

dig_3 ← H'40

cnt ← H'00

Initial settings for RAM area to be used
dig_0 = H'40 = LED1 initial display data " _ "
dig_1 = H'40 = LED2 initial display data " _ "
dig_2 = H'40 = LED3 initial display data " _ "
dig_3 = H'40 = LED4 initial display data " _ "
cnt  = H'00 = LED enable counter initial value = 0

Initial settings for port 2
P20/SCK3: P20 output pin (output data = 0)
P21/RXD:  P21 output pin (output data = 0)
P22/TXD:  P22 input pin

Initial settings for port 5
P50/WKP0:  P50 output pin (output data = 0)
P51/WKP1:  P51 output pin (output data = 0)
P52/WKP2:  P52 output pin (output data = 0)
P53/WKP3:  P53 output pin (output data = 0)
P54/WKP4:  P54 output pin (output data = 0)
P55/WKP5/ADTRG: P55 output pin (output data = 0)
P56:  P56 output pin (output data = 0)
P57:  P57 output pin (output data = 0)

Note: * In this task example task pointer settings are performed by INIT.SCR (assembly language).
```
Initial settings for timer W
TCNT input clock: Internal clock $\phi/8$
Clear timer overflow (OVF) to 0
Start timer counter count-up operation
Disable interrupt requests using OVF flag in TSRW
Initialize 8-bit counter_sub to 15

Initial settings for timer V
Operating mode: Free-running operation
TCNTV input clock: Internal clock $\phi/128$
Counting condition: Falling edge
Initial settings for A/D converter
Scan mode: Single mode
Analog input channel: AN0
A/D conversion time: 134 states (max.)

Clear timer V overflow flag (OVF) to 0
Clear timer W overflow flag (OVFW) to 0
Enable timer overflow interrupts for timer V
Enable timer overflow interrupts for timer W
Calculate coefficient
Calculate correction value
Clear CCR interrupt mask bit (I) to 0

Note: In this application note the OVF flag of timer W is indicated as OVFW and the OVIE flag of timer W is indicated as OVIEW.
5.2 Timer W Interrupt Handling Routine (tmrw)

Are timer overflow interrupts enabled for timer W?

Clear timer W overflow flag (OVFW) to 0

Decrement counter_sub

Is counter_sub value 0?

Initialize 8-bit counter_sub to 15

Set A/D start (ADST) to 1 to begin A/D conversion of AN0

Is A/D conversion finished?

Calculate resistance value (Multiply by 100 to obtain a value with 2 decimal places)

Reference first digit after decimal point of resistance value from dsp_data and store it in dig_0

Reference first digit of resistance value from dsp_data and store it in dig_1

Reference second digit of resistance value from dsp_data and store it in dig_2

Add decimal point (H’80)

Reference third digit of resistance value from dsp_data and store it in dig_3

Notes: In this application note the OVF flag of timer W is indicated as OVFW and the OVIE flag of timer W is indicated as OVIEW.

1. \( x_0 = R \mod 10 \)
2. \( x_1 = (R \mod 100) / 10 \)
3. \( x_2 = (R \mod 1,000) / 100 \)
4. \( x_3 = R / 1,000 \)
5.3 Timer V Interrupt Handling Routine (tmrv)

```
ptr ← &dig_0
ptr ← ptr + cnt
PDR5 ← *ptr
PDR2 ← cnt
cnt ← cnt + 1
```

- Store dig_0 address in ptr
- At cnt value to dig_0 address and store result in ptr
- Store contents of address indicated by ptr in PDR5
- Store cnt value in PDR2
- Increment cnt

```
OVF = 1?

Yes

OVF ← 0
PDR5 ← *ptr
PDR2 ← cnt
```

- Timer overflow interrupt request for timer V?
- Clear timer V overflow flag (OVF) to 0
- Store contents of address indicated by ptr in PDR5

```
cnt >= 4?

No
Yes

cnt ← 0
```

- cnt >= 4?
- Initialize cnt
6. Program Listing

Note: In this application note the OVF flag of timer W is indicated as OVFW and the OVIE flag of timer W is indicated as OVIEW.

INIT.SRC (program listing)

```assembly
.export _INIT
.import _main

.section P, CODE
_INIT:
    mov.w #h'ff80,r7
    ldc.b #b'10000000, ccr
    jmp @_main

.end
```

/* H8/300H tiny Series -H8/36014- Application note */
/* Application */
/* Resistance value measurement example */

#include <machine.h>

/* Symbol definition */
struct BIT {
    unsigned char b7:1; /* bit 7 */
    unsigned char b6:1; /* bit 6 */
    unsigned char b5:1; /* bit 5 */
    unsigned char b4:1; /* bit 4 */
    unsigned char b3:1; /* bit 3 */
    unsigned char b2:1; /* bit 2 */
    unsigned char b1:1; /* bit 1 */
    unsigned char b0:1; /* bit 0 */
};

#define PDR2 *(volatile unsigned char *)0xFFD5 /* Port data register 2 */
#define PCR2 *(volatile unsigned char *)0xFFE5 /* Port control register 2 */
#define PMR5 *(volatile unsigned char *)0xFFE1 /* Port mode register 5 */
#define PUCR5 *(volatile unsigned char *)0xFFD1 /* Port pull-up control register 5 */
#define PDR5 *(volatile unsigned char *)0xFFD8 /* Port data register 5 */
#define PCR5 *(volatile unsigned char *)0xFFE8 /* Port control register 5 */
#define TMRW *(volatile unsigned char *)0xFFF80 /* Timer mode register W */
#define TCRW *(volatile unsigned char *)0xFFF81 /* Timer control register W */
#define TCRW_BIT (*(struct BIT *)0xFFF81) /* Timer Control Register W */
#define TIERW *(volatile unsigned char *)0xFFF82 /* Timer interrupt enable register W */
#define TIERW_BIT (*(struct BIT *)0xFFF82) /* Timer Interrupt Enable Register */
#define OVIEW TIERW_BIT.b7 /* Timer Overflow Interrupt Enable W */
#define TSRW *(volatile unsigned char *)0xFFF83 /* Timer status register W */
#define TSRW_BIT (*(struct BIT *)0xFFF83) /* Timer Status Register W */
#define OVFW TSRW_BIT.b7 /* Timer Overflow W */
#define TCRV0 *(volatile unsigned char *)0xFFA0 /* Timer control register V0 */
#define OVIE TCRV0_BIT.b5 /* Timer overflow interrupt enable */
#define TCSRV *(volatile unsigned char *)0xFFA1 /* Timer control/status register V */
#define OVF TCSRV_BIT.b5 /* Timer overflow flag */
#define TCRV1 *(volatile unsigned char *)0xFFA5 /* Timer control register V1 */
#define ADCSR *(volatile unsigned char *)0xFFB8 /* A/D control/status register */
#define ADCSR_BIT (*(struct BIT *)0xFFB8)
#define ADST ADCSR_BIT.b5 /* A/D start */
#define ADCR *(volatile unsigned char *)0xFFB9 /* A/D control register */
#define ADDRA *(volatile unsigned int *)0xFFB0 /* A/D data register A */
#define ADDRAI *(volatile unsigned int *)0xFFB1 /* A/D data register A */
#define ADDRC *(volatile unsigned int *)0xFFB2 /* A/D data register A */
#define ADDRCI *(volatile unsigned int *)0xFFB3 /* A/D data register A */
#define IH 513.0 /* INPUT HIGH */
#define IL 28.0 /* INPUT LOW */
#define RH 9.347 /* Resistance HIGH */
#define RL 0.002 /* Resistance LOW */

#pragma interrupt (tmrw)
#pragma interrupt (tmrv)

/* Function */
extern void INIT(void); /* Stack pointer set */
void main(void); /* main routine */
void tmrw(void); /* Timer W interrupt routine */
void tmrv(void); /* Timer V interrupt routine */

/* Data table */
const unsigned char dsp_data[16] =
{
  0x3f, /* LED display data = "0" */
  0x06, /* LED display data = "1" */
  0x5b, /* LED display data = "2" */
  0x4f, /* LED display data = "3" */
  0x66, /* LED display data = "4" */
  0x6d, /* LED display data = "5" */
  0x7d, /* LED display data = "6" */
  0x27, /* LED display data = "7" */
  0x7f, /* LED display data = "8" */
  0x6f, /* LED display data = "9" */
  0x77, /* LED display data = "A" */
  0x7c, /* LED display data = "B" */
  0x39, /* LED display data = "C" */
  0x5e, /* LED display data = "D" */
  0x79, /* LED display data = "E" */
  0x71 /* LED display data = "F" */
};
/ * RAM define */
unsigned char dig_0;  /* Dig-0 LED display data store */
unsigned char dig_1;  /* Dig-1 LED display data store */
unsigned char dig_2;  /* Dig-2 LED display data store */
unsigned char dig_3;  /* Dig-3 LED display data store */
unsigned char cnt;    /* LED enable counter */
unsigned char counter_sub; /* sub-counter */
int R;                /* resistance(for display) */
float x;              /* coefficient */
float y;              /* correction */

/* Vector address */
#pragma section V1    /* Vector section set */
void (*const VEC_TBL1[])(void) = {
    INIT /* H'0000 Reset vector */
};
#pragma section V2    /* Vector section set */
void (*const VEC_TBL2[])(void) = {
    tmrw /* H'002a Timer W interrupt vector */
};
#pragma section V3    /* Vector section set */
void (*const VEC_TBL3[])(void) = {
    tmrv /* H'002c Timer V interrupt vector */
};
#pragma section /* P */

/**********************************************************************************
/* Main program */
**********************************************************************************/
void main(void)
{
    set_imask_ccr(1);         /* CCR I-bit = 1 */

dig_0 = 0x40;              /* Used RAM area initialize */
dig_1 = 0x40;              /* Used RAM area initialize */
dig_2 = 0x40;              /* Used RAM area initialize */
dig_3 = 0x40;              /* Used RAM area initialize */
cnt = 0x00;                /* Used RAM area initialize */
PCR2 = 0x03;               /* Port 2 initialize */
PMR5 = 0x00;               /* Port 5 initialize */
PUCR5 = 0x00;
PDR5 = 0x00;
PCR5 = 0xff;

TCRW = 0x30;               /* Timer W initialize */
TSRW = 0x00;               /* Clock Select */
TMRW = 0xC8;               /* Clear OVF */
TIERW = 0x00;              /* Timer Counter Count Start */
counter_sub = 15;          /* Initialize 8bit Counter_sub */
TCRV0 = 0x03; /* Timer V initialize */
TCRV1 = 0xe3; /* Internal clock select */
TCSR V = 0x10; /* Clear OVF to 0 */

ADCSR = 0x00; /* A/D converter initialize */
ADCR  = 0x7e;

OVF = 0; /* Clear OVF to 0 */
OVFW = 0; /* Clear OVF to 0 */
OVIE = 1; /* Timer V OVF interrupt enable */
OVIEW = 1; /* Timer W OVF interrupt enable */

x = -(RH - RL) / (IH - IL); /* Get coefficient */
y = RL - IH * x; /* Get correction */

set_imask_ccr(0); /* CCR I-bit = 0 */

while(1);
}

/*******************************************************************************/
/* Timer W Interrupt */
/*******************************************************************************/
void tmrw(void)
{
  if ( OVFW == 1 ) {
    OVFW = 0; /* Clear OVF */
    counter_sub--; /* Decrement 8bit Counter */
    if ( counter_sub == 0x00 ) { /* 8bit Counter != H'00 */
      counter_sub = 15; /* Initialize 8bit Counter_sub */
      ADST = 1; /* A/D converter start */
      while(ADST == 1); /* A/D converter end ? */
      R = ((ADDRA >> 6 & 0x03ff) * x + y) * 100.0;
      dig_0 = dsp_data[R % 10]; /* Dig-0 LED display data set */
      dig_1 = dsp_data[(R % 100)/10]; /* Dig-1 LED display data set */
      dig_2 = dsp_data[(R % 1000)/100] | 0x80; /* Dig-2 LED display data set */
      dig_3 = dsp_data[R / 1000]; /* Dig-3 LED display data set */
    }
  }
}

/*******************************************************************************/
/* Timer V Interrupt */
/*******************************************************************************/
void tmrv(void)
{
  unsigned char *ptr; /* Pointer set */
  ptr = &dig_0; /* LED display data store address set */

  while(OVF == 1){ /* OVF = 1 ? */
    OVF = 0; /* Clear OVF to 0 */
    ptr += cnt; /* LED display data read */
    PDR5 = *ptr; /* LED display data output */
  }
}
PDR2 = cnt;       /* LED enable data output */
cnt++;            /* "cnt" increment */
if (cnt >= 4){    /* 4 times end ? */
    cnt = 0;     /* "cnt" initialize */
}
## Revision Record

<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>Dec.20.04</td>
<td></td>
<td>26</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>


Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.

2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.

3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.

The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.

Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).

4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.

5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.

6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.

7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.

8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.