To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 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

April 1st, 2010
Renesas Electronics Corporation

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

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

H8/300H Tiny Series

Connecting a CdS Cell

Introduction

A CdS cell is connected to the analog input pin, and results displayed on seven-segment LEDs.

Target Device

H8/36014

Contents

1. Specifications

1. Figure 1.1 shows the hardware configuration example of connecting a visible photoconductive cell. As is shown in figure 1.1, a resistor and visible photoconductive cell: CdS cells are connected to the analog input pin 3 (AN3 pin).
2. The signal on the AN3 pin is A/D converted, after which the results of A/D conversion are displayed on the 7-segment LED connected to the I/O port.
3. The 7-segment LED display indicates the 10-bit result of A/D conversion as a hexadecimal value.
4. A/D conversion is at a 0.49152-s interval.

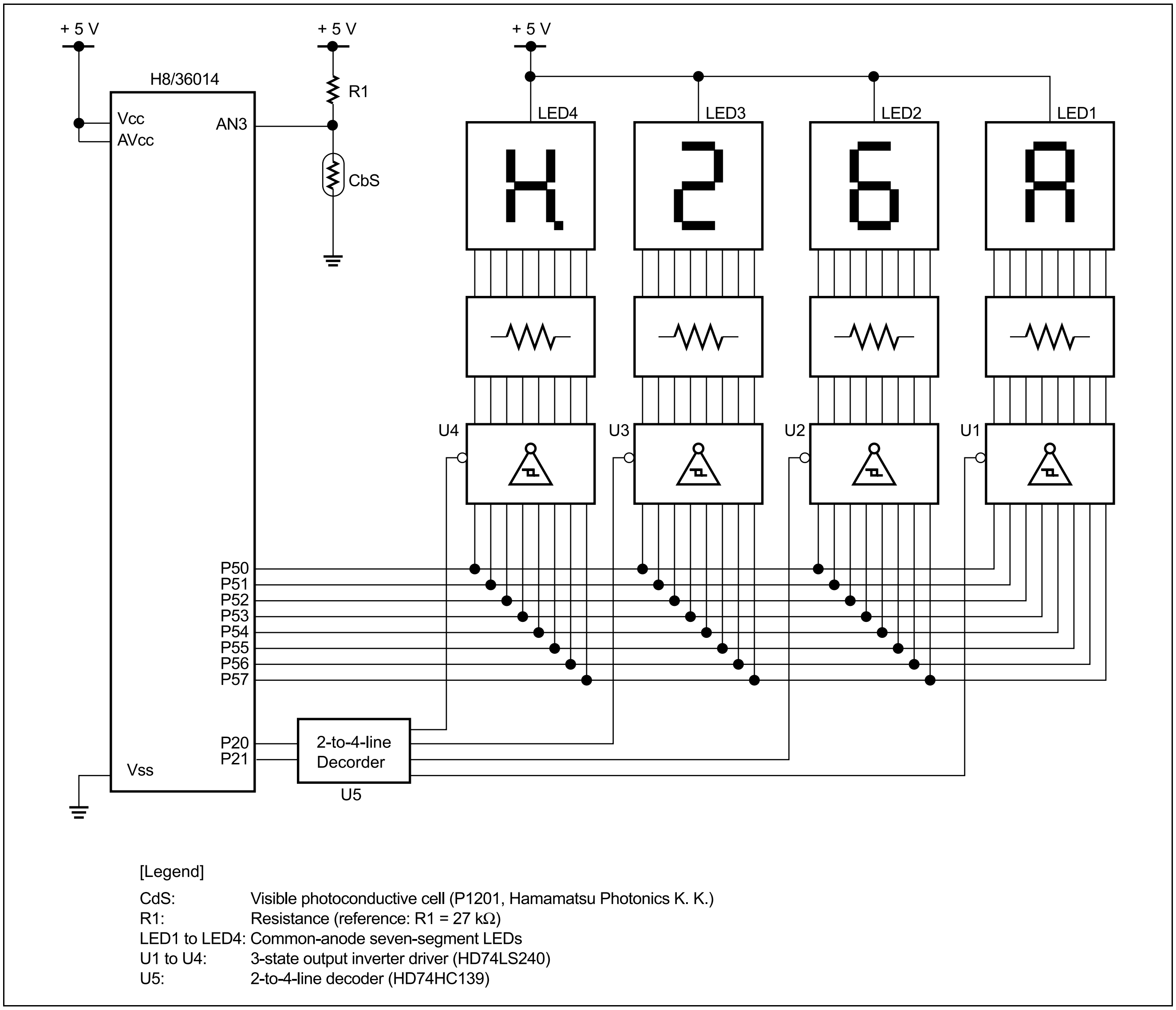


Figure 1.1 Hardware Configuration

5. In this sample task, the H8/36014 operating voltage (Vcc) and analog power supply voltage (AVcc) are 5.0 V, and the OSC clock frequency is 16 MHz.

6. The visible photoconductive cell: CdS cell used in this sample task is made by Hamamatsu Photonics K.K. (model P1201). The specifications of the CdS cell used are as follows.
   A. Table 1.1 lists the absolute maximum ratings and characteristics of the CdS cell (P1201).

Table 1.1 Absolute Maximum Ratings and Characteristics

Maximum Absolute Ratings			Characteristics*[1]						
				Resistance Value*[2]				Response Time *[5]	
				10 lx, 2856K		0 lx*[3]	γ^{100}_{10} *[4]		
Applied voltage (Vdc)	Allowable dissipation (mW)	Ambient temperature (°C)	Maximum Sensitivity Wave Length (nm)	Min. (kΩ)	Max. (kΩ)	Min. (MΩ)	100 to 10 lx	Rise Time (ms)	Fall Time (ms)
100	70	-30 to +80	540	20	60	5.0	0.75	40	30

Notes: 1. Values after exposure left for one to two hours under irradiation light of 100 to 500 lx.
2. A standard tungsten light bulb having a color temperature of 2856K is used as the light source.
3. A value after 10 seconds passed from shielding 10-lx irradiation light.
4. A typical value (fluctuation: ±0.10) of inclination (γ: gamma) characteristics between irradiation light of 10 lx and 100 lx

$$\gamma^{100}_{10} = \left| \frac{\log(R100) - \log(R10)}{\log(E100) - \log(E10)} \right|$$

5. The rise time is the time to reach 63% of saturation conductance (bright) value. The fall time is the time to reach 37% of saturation conductance.

7. Features of the CdS cell are described below.
   A. The visible photoconductive cell P1201 is a semiconductor sensor which utilizes a photoconductive effect where a resistance value decreases when irradiating light. Since this is a non-polarized resistance element approximate to the wavelength sensitivity (relative luminous efficiency) of human eyes, a circuit with a CdS can easily be prepared.
   B. As is shown in figure 1.2, since the resistance of the sensor decreases as the brightness increases, the value displayed on LED also decreases.

8. Operation of this sample task is described below.
   A. The brightness 20 cm away from a candle is approximately 10 lux (l_x).
   B. As is shown in figure 1.2, the resistance value of the visible photoconductive cell P1201 is approximately 35 kΩ when the brightness is 10 lux (l_x).
   C. In the case of this sample task, since a fixed resistance of 27 kΩ is arranged in series, the voltage Vin (V) applied to the analog port AN3 pin is 2.823 (V) found by dividing V_{DD}.

$$\text{Vin} = V_{DD} \times \frac{35\text{ k}}{(27\text{ k} + 35\text{ k})} = 2.823\ (\text{V})$$

   D. The 10-bit data of the A/D conversion result of the voltage Vin of 2.823 (V) in hexadecimal format is "H'242".

$$\frac{2.823\ (\text{V}) \times 1023}{5\ (\text{V})} = 578 = \text{H'242}$$

   E. Accordingly, the LED display at approximately 10 lux (l_x) is "H'242".

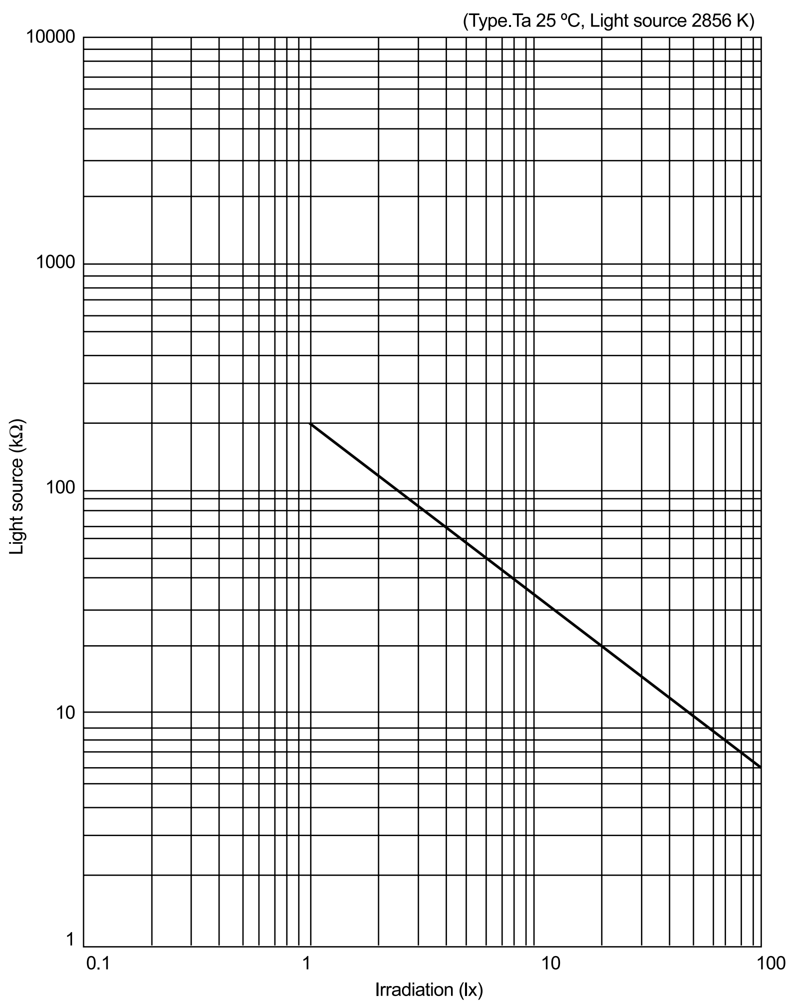
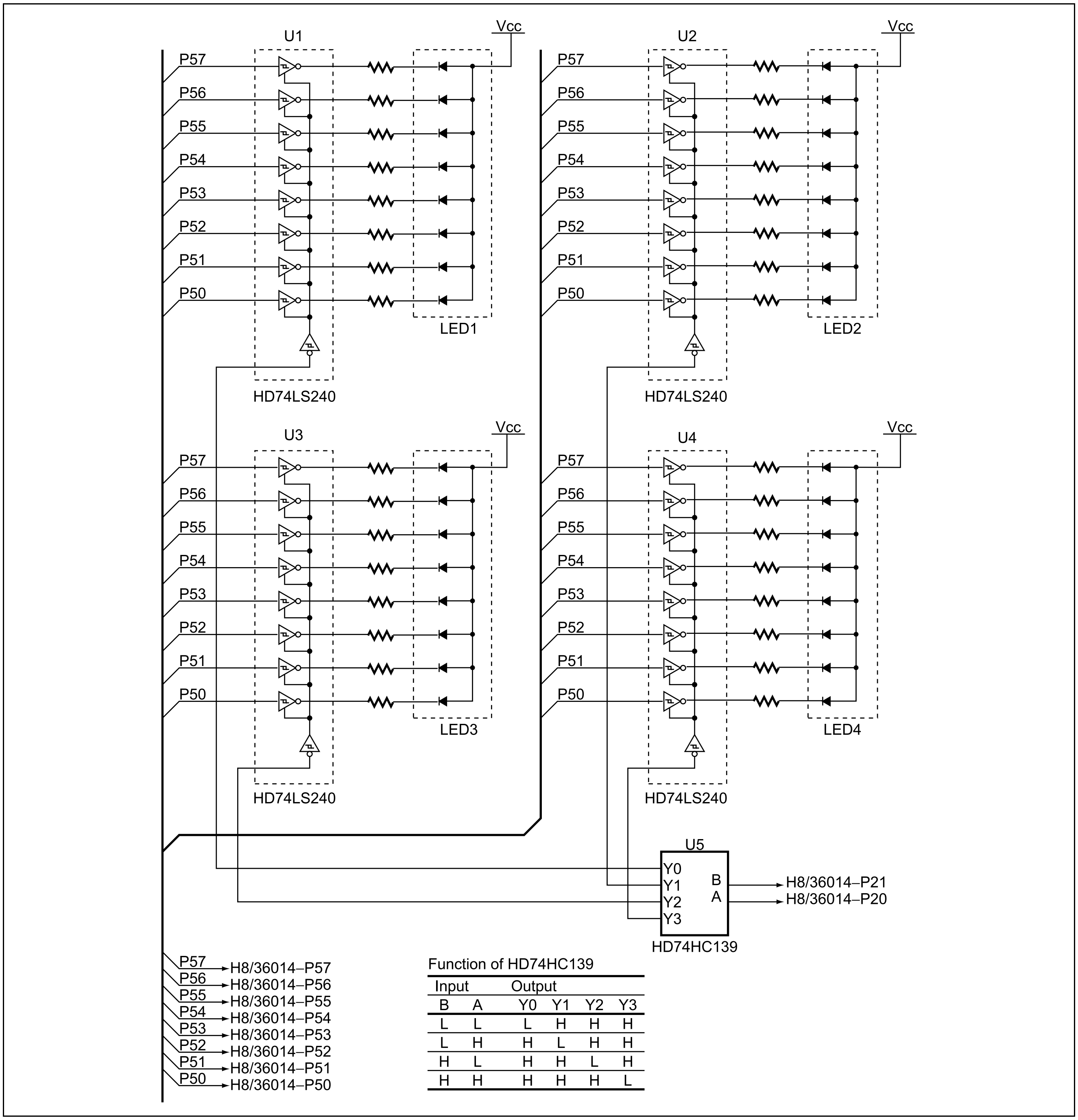


Figure 1.2 Resistance Value vs. Irradiation (Reference)

9. In this sample task, display on the 7-segment LED is handled by inputting port signals output from the microcontroller to the inputs to the tri-state-output inverter drivers (HD74LS240), the outputs from which are in turn connected to the cathodes of the 7-segment LEDs. The port-output pins are used to control the display on each of the four 7-segment LEDs. The enable pins of the tri-state inverter driver control switching between the 7-segment displays. The signals used to switch between the displays are generated by the 2-to-4-line decoder (HD74HC139), which is controlled by two port-pin outputs. Figure 1.3 shows how the 7-segment LEDs are controlled.



Function of HD74HC139

Input		Output			
B	A	Y0	Y1	Y2	Y3
L	L	L	H	H	H
L	H	H	L	H	H
H	L	H	H	L	H
H	H	H	H	H	L

Figure 1.3 7-Segment LED Control

10. In this sample task, the results of A/D conversion are displayed in hexadecimal format (H'3FF to H'000) on three of the 7-segment LEDs. Figure 1.4 shows how this is done.

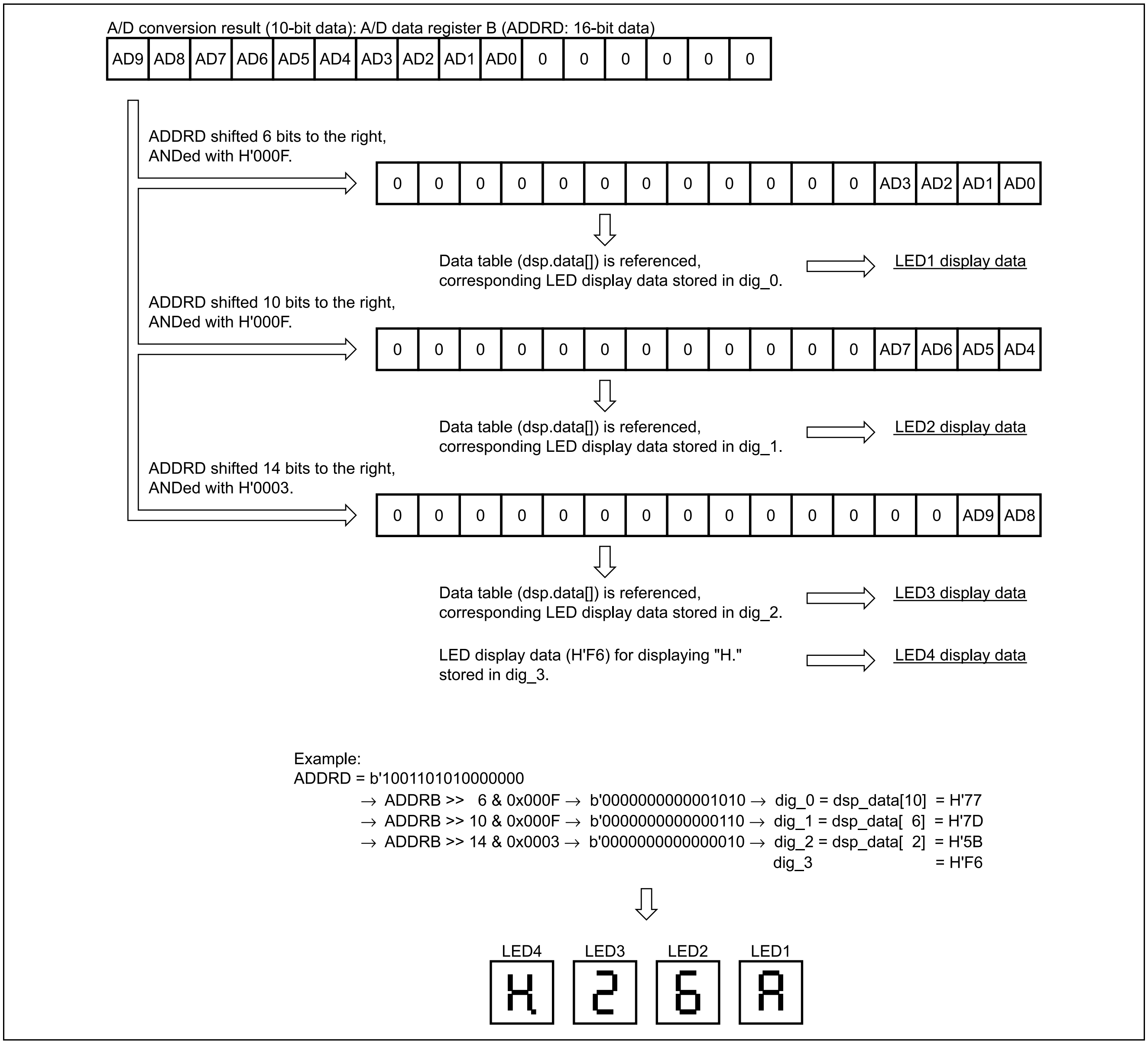


Figure 1.4 How A/D Results are Displayed on the LEDs

2. Description of Function

1. Figure 2.1 shows the functions used in the H8/36014 for this sample task, while table 2.1 indicates the allocation of functions.

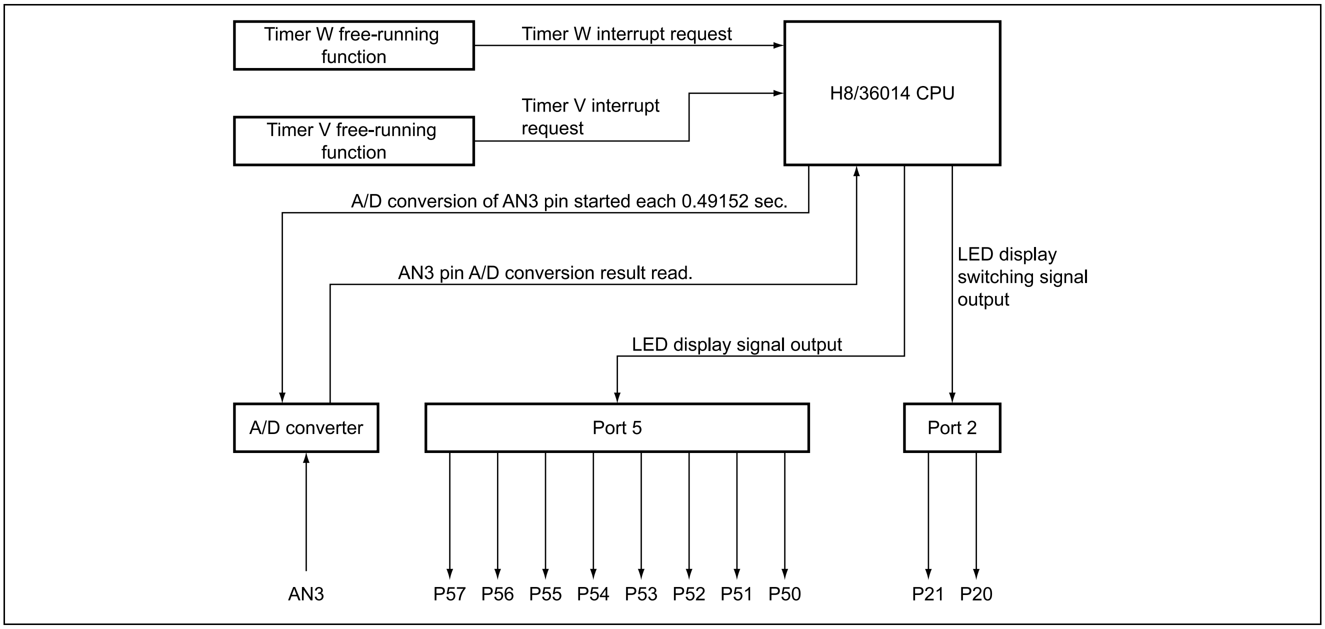


Figure 2.1 Block Diagram of Functions Used

Table 2.1 Function Allocation

Function Used	Function Allocation
Timer W	The timer W free running function is used to perform A/D conversion of the analog input pin 3 (AN3) signal. The A/D conversion period is measured by the timer W overflow period of 32.768 ms.
Timer V	The timer V free running function is used to control switching of the 7-segment LED display. Dynamic lighting is performed by light the four 7-segment LEDs in order at every 2.048 ms Timer V overflow period.
A/D converter	The divided voltage between the AVcc and GND is A/D converted according to the fixed resistance and resistance variation due to the amount of acceptance of the CdS cell connected to the A/D converter analog input pin 3 (AN3).
Port 2	Display on the four 7-segment LEDs is switched by the port 2 output pins P20 and P21. The P20 and P21 output pins are connected to I/O pins of the 2-to-4-line decoder.
Port 5	Results are displayed on the 7-segment LEDs by output from the port 5 output pins P50 to P57. The 10-bit data which is the A/D conversion result at pin AN3 is converted into a three-digit hexadecimal number for output to the LEDs.

2. A diagram of the connection of the seven-segment LEDs used is shown in figure 2.2. As shown in the figure, by outputting "high" from port 5, the corresponding LED segments are lit. The relationship between the port 5 outputs and the LED display data is shown in table 2.2.

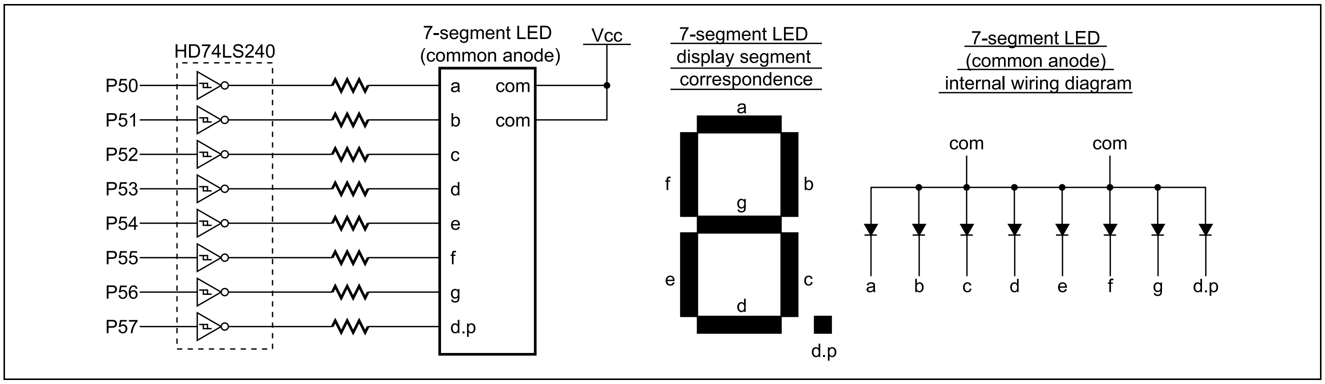
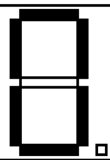
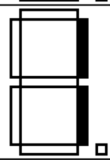
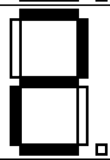
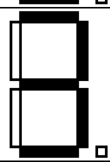
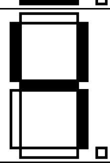
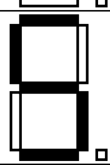
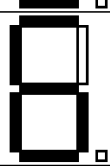
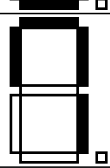
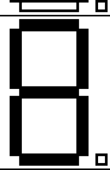
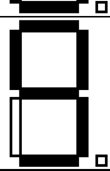
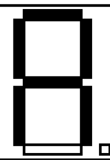
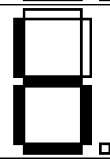
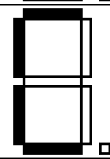
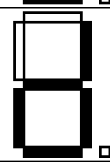
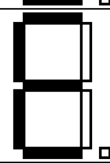
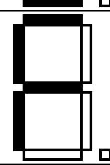
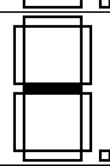
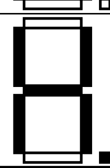


Figure 2.2 7-Segment LED Connection Diagram and Internal Wiring

Table 2.2 Relation between Port 5 Outputs and 7-Segment LED Display Data

LED Display	Port 5 Output Data							
	P57	P56	P55	P54	P53	P52	P51	P50
	0	0	1	1	1	1	1	1
	0	0	0	0	0	1	1	0
	0	1	0	1	1	0	1	1
	0	1	0	0	1	1	1	1
	0	1	1	0	0	1	1	0
	0	1	1	0	1	1	0	1
	0	1	1	1	1	1	0	1
	0	0	1	0	0	1	1	1
	0	1	1	1	1	1	1	1
	0	1	1	0	1	1	1	1
	0	1	1	1	0	1	1	1
	0	1	1	1	1	1	0	0
	0	0	1	1	1	0	0	1
	0	1	0	1	1	1	1	0
	0	1	1	1	1	0	0	1
	0	1	1	1	0	0	0	1
	0	1	0	0	0	0	0	0
	1	1	1	1	0	1	1	0

3. Principles of Operation

1. Figure 3.1 shows the principle of operation when using the timer W to perform A/D conversion of the AN3 pin output. As shown in figure 3.1, in this sample task the A/D conversion period (0.49152 s) is measured using the timer W overflow flag in the tmrw routine, without using the A/D conversion interrupt, to judge the end of A/D conversion.

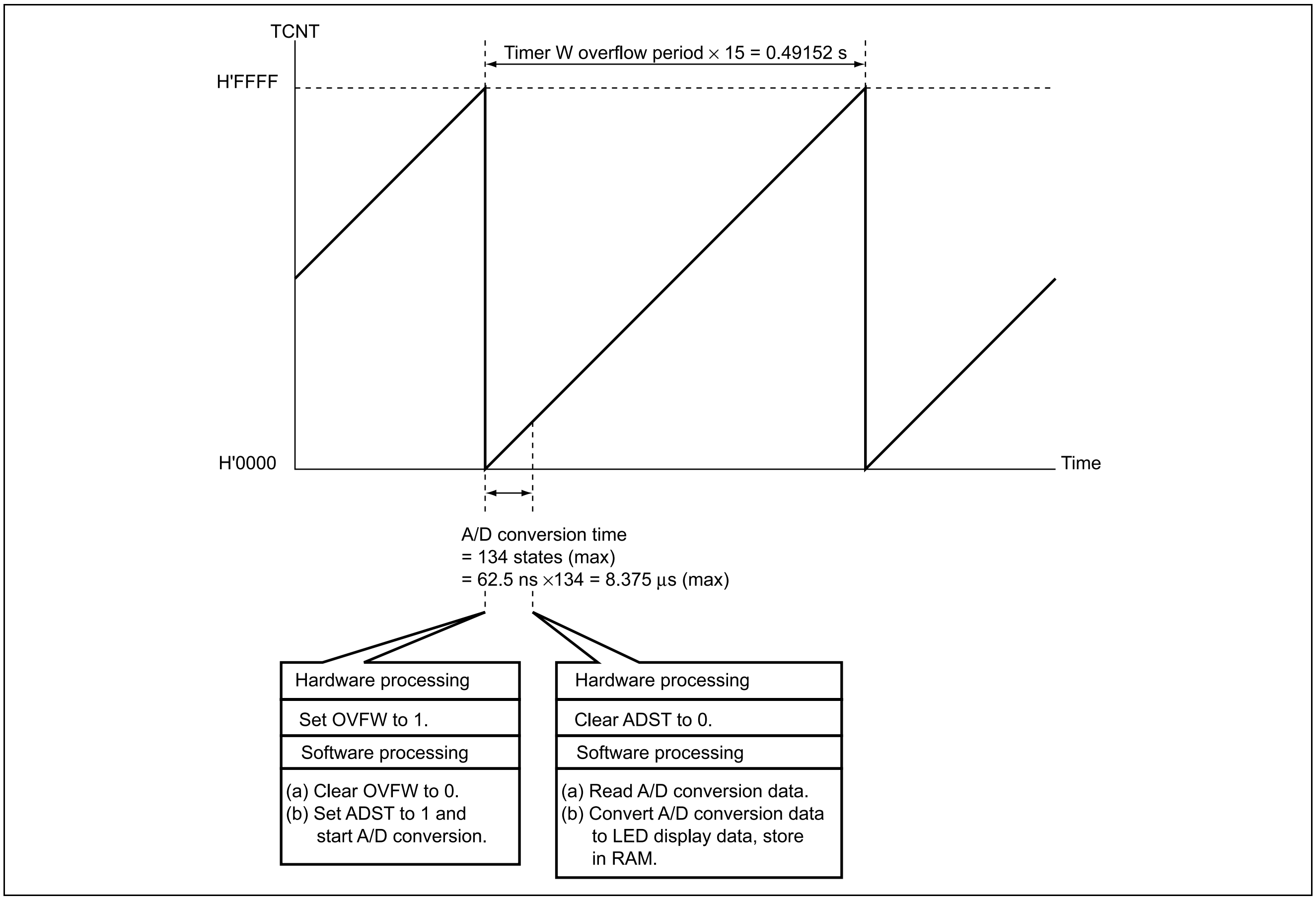


Figure 3.1 Principle of Operation of A/D Conversion of the AN3 Pin Output using Timer W

2. The principle of operation of the seven-segment LED display control is as follows. Figure 3.2 describes the operation for the case in which "3210" is displayed on LED4 through LED1. As shown in figure 3.2, by displaying the data in order on LED1 through LED4 at each timer V overflow period, data is displayed dynamically on the seven-segment LEDs.

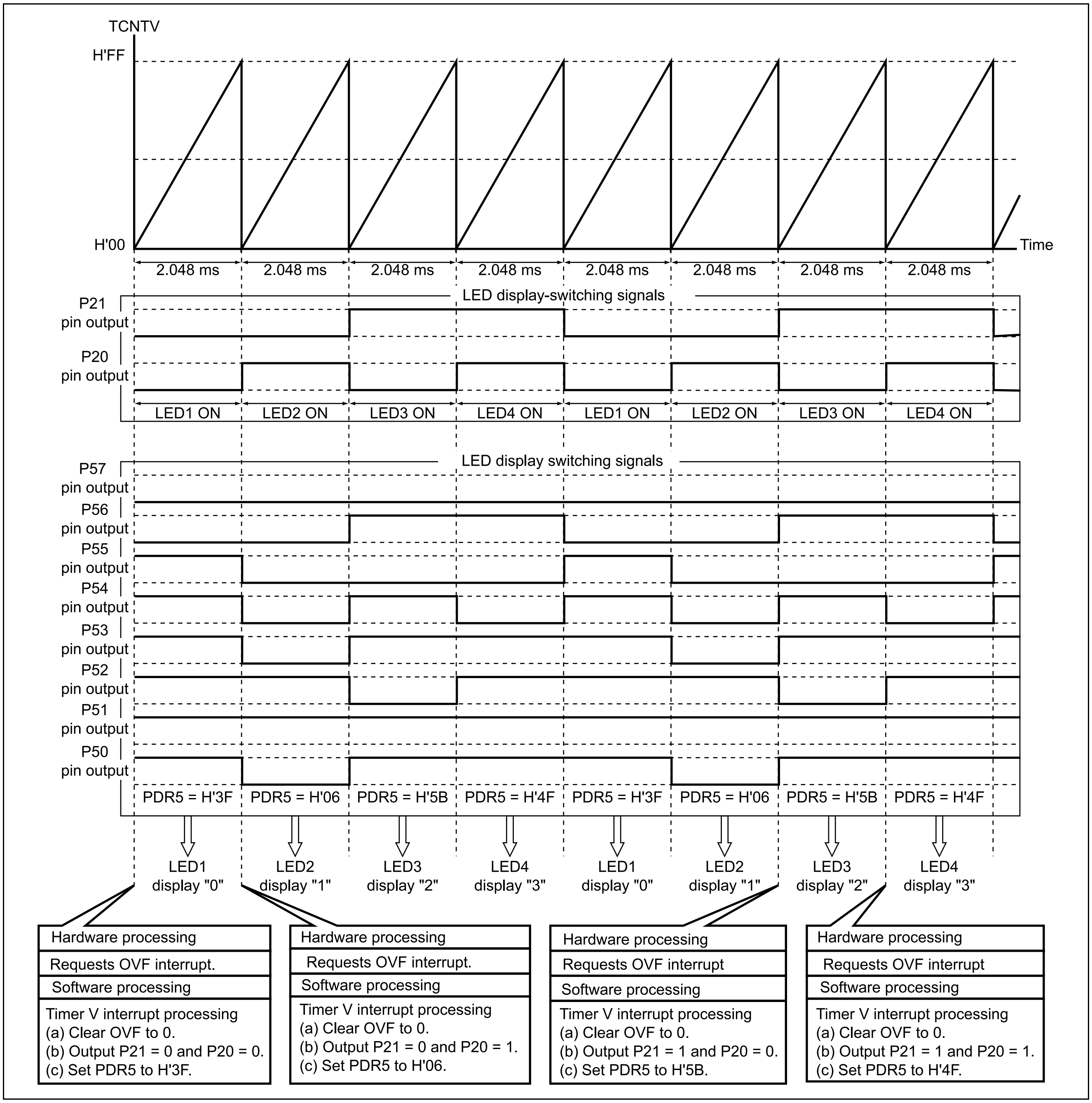


Figure 3.2 Principle of Operation of 7-Segment LED Display Control

4. Description of Software

4.1 Description of Modules

Table 4.1 describes the modules used in this sample task.

Table 4.1 Description of Modules

Module Name	Label Name	Function
Main routine	main	Sets initial values, enables interrupts
Timer W interrupt processing routine	tmrw	Clears interrupt flag, converts A/D conversion data into LED display data, stores in RAM
Timer V interrupt processing routine	tmrv	Clears interrupt flag, controls LED display data output and LED display switching

4.2 Description of arguments

This sample task uses no arguments.

4.3 Description of Internal Registers

The internal registers used in this sample task are described in table 4.2.

Table 4.2 Description of Internal Registers

Register Name		Function	Address	Setting
TCRV0		Timer control register V0: Selects TCNTV input clock, sets TCNTV clear conditions, controls interrupt requests	H'FFA0	H'03 (initial setting)
	CMIEB	Compare-match interrupt enable B: when CMIEB = 0, disables interrupt requests by CMFB in TCSRV	Bit 7	0
	CMIEA	Compare-match interrupt enable A: when CMIEA = 0, disables interrupt requests by CMFA in TCSRV	Bit 6	0
	OVIE	Timer overflow interrupt enable: when OVIE = 0, disables interrupt requests by OVF in TCSRV, when OVIE = 1, enables interrupt requests by OVF in TCSRV	Bit 5	0/1
	CCLR1	Counter clear 1, 0: Sets conditions to clear TCNTV	Bit 4	0
	CCLR0	When CCLR1 = 0 and CCLR0 = 0, TCNTV clearing is disabled	Bit 3	0
	CKS2	Clock select 2 to 0: Combined with ICKS0 in TCRV1, selects clock for input to TCNTV and count conditions	Bit 2	0
	CKS1		Bit 1	1
	CKS0	When CKS2 = 0, CKS1 = 1, CKS0 = 1, and ICKS0 = 1, TCNTV counts on the falling edge of the internal clock ϕ/128	Bit 0	1

Register Name		Function	Address	Setting
TCSRV		Timer control/status register V: Displays status flag, controls output of compare-match	H'FFA1	H'10
	CMFB	Compare-match flag B: Set to 1 when the values of TCNTV and TCORB match	Bit 7	0
	CMFA	Compare-match flag A: Set to 1 when the values of TCNTV and TCORA match	Bit 6	0
	OVF	Timer overflow flag: When the TCNTV value overflows, OVF is set to 1 In the OVF = 1 state, when 0 is written to OVF after reading OVF, OVF is cleared to 0	Bit 5	0
	OS3	Output select 3, 2: Sets the output level of the TMOV pin for the output of compare-match B When set to OS3 = 0 and OS2 = 0, no change occurs	Bit 3	0
	OS2		Bit 2	0
	OS1	Output select 1, 0: Sets the output level of the TMOV pin for the output of compare-match A When set to OS1 = 0 and OS0 = 0, no change occurs	Bit 1	0
	OS0		Bit 0	0
TCRV1		Timer control register V1: Selects TRGV pin edge, TRGV input enable, TCNTV input clock	H'FFA5	H'E3
	TVEG1	TRGV input edge select 1, 0: Selects input edge of TRGV pin When set to TREG1 = 0, TREG0 = 0, disables trigger input from the TRGV pin	Bit 4	0
	TVEG0		Bit 3	0
	TRGE	TRGV input enable: Enables/disables TCNTV count-up by edge input selected using TVEG1 and TVEG0 When set to TREG = 0, disables start of TCNTV count-up due to input to TRGV pin and halt of TCNTV count-up when TCNTV is cleared due to a compare-match	Bit 2	0
	ICKS0	Internal clock select 0: Combined with CKS2 to CKS0 of TCRV0, selects clock for input to TCNTV and count conditions; when set to CKS2 = 0, CKS1 = 1, CKS0 = 1 and ICKS0 = 1, TCNTV counts on the falling edge of the internal clock ϕ/128	Bit 0	1
TMRW		Timer mode register W: Selects general register functions and timer output mode	H'FF80	H'80
	CTS	Counter start: When CTS = 1, indicates TCNT has started counting When CTS = 0, indicates TCNT has stopped counting	Bit 7	1
TCRW		Timer control register W: Selects counter clock Sets counter clear conditions and timer output level	H'FF81	H'30
	CKS2	Clock select: When CKS2 = 0, CKS1 = 1 and CKS0 = 1, sets TCNT input clock to system clock divided by 8	Bit 6	0
	CKS1		Bit 5	1
	CKS0		Bit 4	1
TIERW		Timer interrupt enable register W: controls timer W interrupt requests	H'FF82	H'00 (initial setting)
	OVIE	Timer overflow interrupt enable: when OVIE = 0, disables interrupt requests by OVF When OVIE = 1, enables interrupt requests by OVF	Bit 7	0/1

Register Name		Function	Address	Setting
TSRW		Indicates interrupt request status	H'FF83	H'00
	OVF	Timer overflow: When OVF = 0, indicates TCNT has not overflowed When OVF = 1, indicates that TCNT has overflowed	Bit 7	0
TCNT		Timer counter: 16-bit count-up counter which takes as input the system clock frequency-divided by 8	H'FF86	H'00
ADCSR		A/D control/status register: Consists of A/D converter control bits and conversion status bits	H'FFB8	H'03 (initial setting)
	ADF	A/D end flag: Set to 1 when A/D conversion ends in single mode After reading in the ADF = 1 state, ADF is cleared to 0 by writing 0	Bit 7	0
	ADIE	A/D interrupt enable: When 0, disables A/D conversion end interrupt requests by the ADF	Bit 6	0
	ADST	A/D start: Setting this to 1 starts A/D conversion Automatically cleared to 0 when A/D conversion ends in single mode	Bit 5	0/1
	SCAN	Scan mode: When 0, A/D conversion mode is single mode	Bit 4	0
	CKS	Clock select: When 0, A/D conversion time = 134 states (max)	Bit 3	0
	CH2	Channel select: Selects the analog input channel; when CH2 = 0, CH1 = 1 and CH0 = 1 (SCAN = 0), input channel is AN3	Bit 2	0
	CH1		Bit 1	1
	CH0		Bit 0	1
ADCR		A/D control register: Enables A/D conversion start by an external trigger	H'FFB9	H'7E
	TRGE	Trigger enable: When 0, disables the start of A/D conversion by edge input to the external trigger pin (ADTRG)	Bit 7	0
ADDRD		A/D data register D: Stores the AN3 A/D conversion result	H'FFB6	—
PCR2		Port control register 2: Selects, for each bit, pin I/O to use as the port 2 general I/O port When PCR2 = H'03, pin P22 functions as a general input pin, and pins P21 and P20 function as general output pins	H'FFE5	H'03
PDR2		Port data register 2: General I/O port data register for port 2	H'FFD5	H'F8
PMR5		Port mode register 5: Sets the port 5 pin functions	H'FFE1	H'00
	POF7	P57 pin function switch: When 0, P57 functions as a general I/O port	Bit 7	0
	POF6	P56 pin function switch: When 0, P56 functions as a general I/O port	Bit 6	0
	WKP5	P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ pin function switch: When 0, P55 functions as a general I/O port	Bit 5	0
	WKP4	P54/$\overline{\text{WKP4}}$ pin function switch: When 0, P54 functions as a general I/O port	Bit 4	0
	WKP3	P53/$\overline{\text{WKP3}}$ pin function switch: When 0, P53 functions as a general I/O port	Bit 3	0
	WKP2	P52/$\overline{\text{WKP2}}$ pin function switch: When 0, P52 functions as a general I/O port	Bit 2	0
	WKP1	P51/$\overline{\text{WKP1}}$ pin function switch: When 0, P51 functions as a general I/O port	Bit 1	0
	WKP0	P50/$\overline{\text{WKP0}}$ pin function switch: When 0, P50 functions as a general I/O port	Bit 0	0

Register Name	Function	Address	Setting
PUCR5	Port pull-up control register 5: Controls, for each bit, the pull-up MOS of each port 5 pin set to an input port When PUCR5 = H'00, pull-up MOS for pins P57 to P50 is turned off	H'FFD1	H'00
PDR5	Port data register 5: Port 5 general I/O port data register	H'FFD8	H'00
PCR5	Port control register 5: Selects, for each bit, the pin I/O used as a port 5 general I/O port When PCR5 = H'FF, the pins P57 to P50 function as general output pins	H'FFE8	H'FF

4.4 Description of RAM

Table 4.3 describes the RAM used in this sample task.

Table 4.3 Description of RAM

Label Name	Function	Address	Used in
dig_0	Store LED1 display data (1 byte)	H'FB80	main, tmrw
dig_1	Store LED2 display data (1 byte)	H'FB81	main, tmrw
dig_2	Store LED3 display data (1 byte)	H'FB82	main, tmrw
dig_3	Store LED4 display data (1 byte)	H'FB83	main, tmrw
cnt	8-bit counter to switch LED1 to LED4 display (1 byte)	H'FB84	main, tmrv
counter_sub	8-bit counter to adjust A/D acquisition interval (1 byte)	H'FB85	main, tmrw

4.5 Description of data table

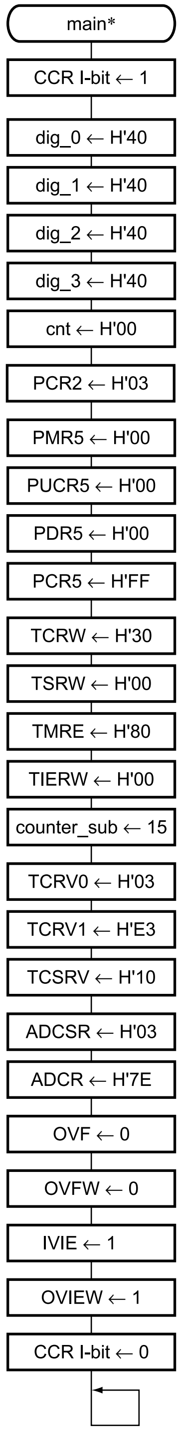
In this sample task, seven-segment LED display data is stored as a one-dimensional data table in ROM. Table 4.4 describes the seven-segment LED display data table (dsp_data[ ]).

Table 4.4 Description of 7-Segment LED Display Data Table (dsp_data[ ])

Array Name	Data	Data Description	Data Size	Address
dsp_data[0]	H'3F	Data output from port 5 to display "0" on an LED	1 byte	H'021C
dsp_data[1]	H'06	Data output from port 5 to display "1" on an LED	1 byte	H'021D
dsp_data[2]	H'5B	Data output from port 5 to display "2" on an LED	1 byte	H'021E
dsp_data[3]	H'4F	Data output from port 5 to display "3" on an LED	1 byte	H'021F
dsp_data[4]	H'66	Data output from port 5 to display "4" on an LED	1 byte	H'0220
dsp_data[5]	H'6D	Data output from port 5 to display "5" on an LED	1 byte	H'0221
dsp_data[6]	H'7D	Data output from port 5 to display "6" on an LED	1 byte	H'0222
dsp_data[7]	H'27	Data output from port 5 to display "7" on an LED	1 byte	H'0223
dsp_data[8]	H'7F	Data output from port 5 to display "8" on an LED	1 byte	H'0224
dsp_data[9]	H'6F	Data output from port 5 to display "9" on an LED	1 byte	H'0225
dsp_data[10]	H'77	Data output from port 5 to display "A" on an LED	1 byte	H'0226
dsp_data[11]	H'7C	Data output from port 5 to display "b" on an LED	1 byte	H'0227
dsp_data[12]	H'39	Data output from port 5 to display "C" on an LED	1 byte	H'0228
dsp_data[13]	H'5E	Data output from port 5 to display "d" on an LED	1 byte	H'0229
dsp_data[14]	H'79	Data output from port 5 to display "E" on an LED	1 byte	H'022A
dsp_data[15]	H'71	Data output from port 5 to display "F" on an LED	1 byte	H'022B

5. Flowchart

1. Main routine (main)



Mask interrupt requests by setting the CCR interrupt mask bit (I) to 1.

Initial settings for RAM area
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: Initial value of LED enable counter = 0

Port 2 initial settings
P20/SCK3: P20 output pin
P21/RXD: P21 output pin
P22/TXD: P22 input pin

Port 5 initial settings
P50/$\overline{WKP0}$: P50 output pin (output data = 0)
P51/$\overline{WKP1}$: P51 output pin (output data = 0)
P52/$\overline{WKP2}$: P52 output pin (output data = 0)
P53/$\overline{WKP3}$: P53 output pin (output data = 0)
P54/$\overline{WKP4}$: P54 output pin (output data = 0)
P55/$\overline{WKP5}$/$\overline{ADTRG}$: P55 output pin (output data = 0)
P56: P56 output pin (output data = 0)
P57: P57 output pin (output data = 0)

Timer W initial settings
TCNT input clock: Internal clock φ divided by 8
Clear timer overflow (OVF) to "0".
Start timer counter counting operation.
Interrupt requests cannot occur due to the TSRW OVF flag.
8-bit counter_sub initialized to 15.

Timer V initial settings
Operation mode: Free-running operation
TCNTV input clock: Internal clock φ/128
Count condition: Falling edge

A/D converter initial settings
Scan mode: Single mode
Analog input channel: AN3
A/D conversion time: 134 states (max)

Clear the timer overflow flag (OVF) to 0.

Clear the timer overflow flag (OVF) to 0.

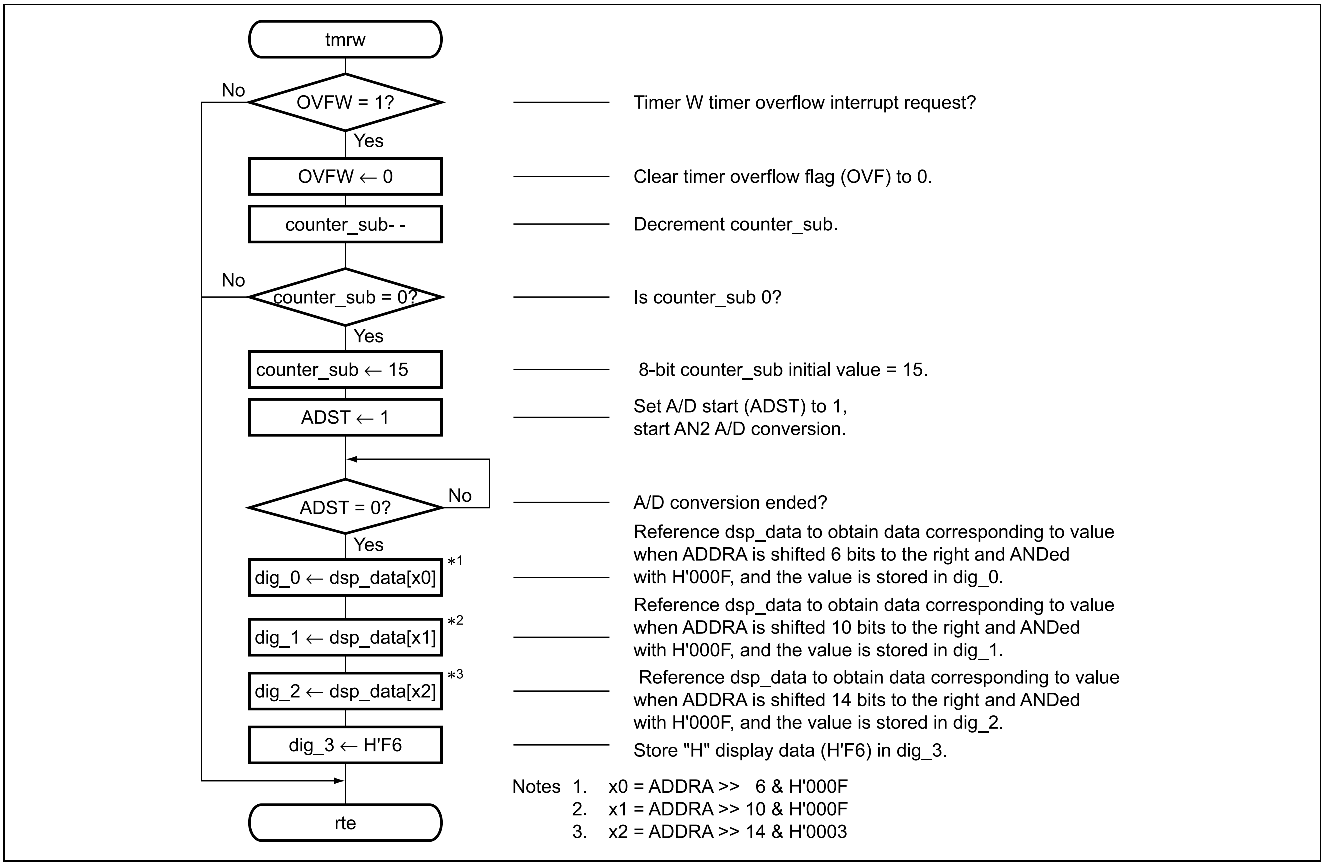
Enable timer V timer overflow interrupt requests.

Enable timer W timer overflow interrupt requests.

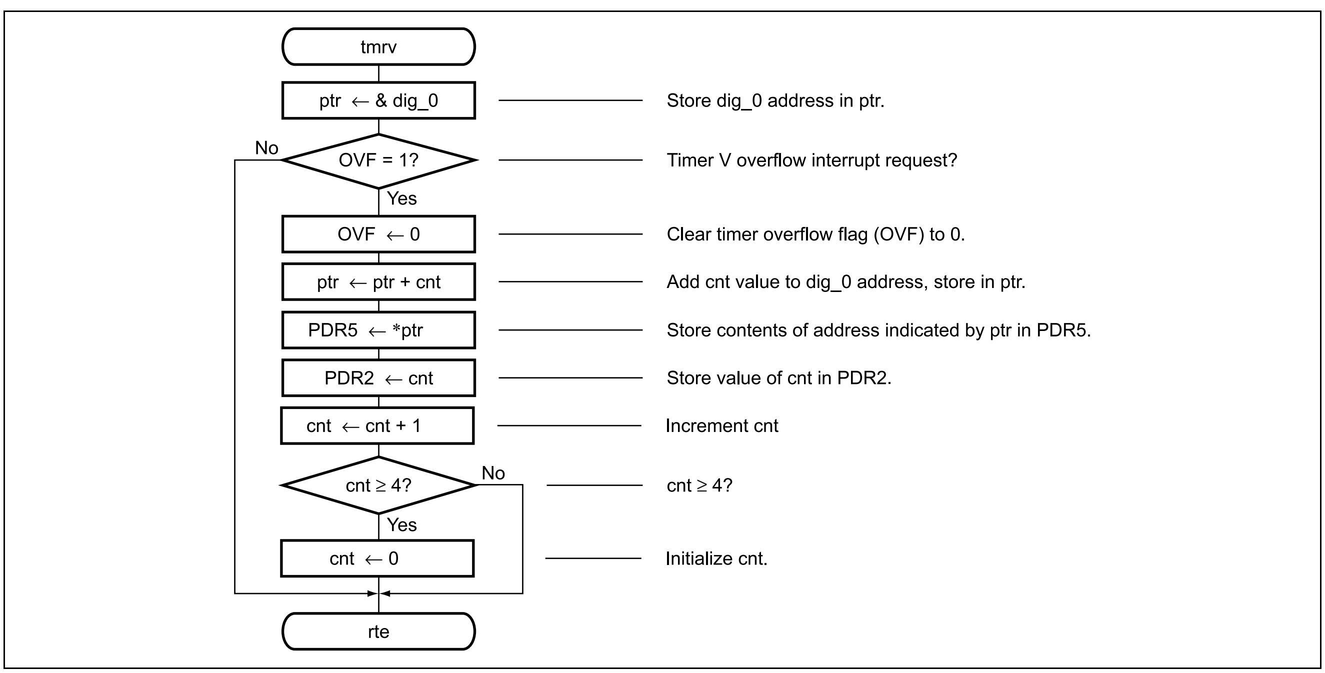
Clear the interrupt mask bit (I) in CCR to 0.

Note: * In this sample task, the stack pointer is specified by INIT.SRC (assembly language).

2. Timer W interrupt processing routine (tmrw)



3. Timer V interrupt processing routine (tmrv)



6. Program Listing

INIT.SRC (program list)

```
     .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 example                                                                              */
/*   Connecting a CdS cell                                                                            */

#include <machine.h>

/*   Symbol defnition                                                                                 */
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 *)0xFF80              /* Timer mode register W             */
#define TCRW        *(volatile unsigned char *)0xFF81              /* Timer control register W          */
#define TCRW_BIT    (*(struct BIT *)0xFF81)                        /* Timer Control Register W          */
#define TIERW       *(volatile unsigned char *)0xFF82              /* Timer interrupt enable register W */
#define TIERW_BIT   (*(struct BIT *)0xFF82)                        /* Timer Interrupt Enable Register   */
#define OVIEW       TIERW_BIT.b7                                   /* Timer Overflow Interrupt Enable W */
#define TSRW        *(volatile unsigned char *)0xFF83              /* Timer status register W           */
#define TSRW_BIT    (*(struct BIT *)0xFF83)                        /* Timer Status Register W           */
#define OVFW        TSRW_BIT.b7                                    /* Timer Over flow W                 */

#define TCRV0       *(volatile unsigned char *)0xFFA0              /* Timer control register V0         */
#define TCRV0_BIT   (*(struct BIT *)0xFFA0)
#define OVIE        TCRV0_BIT.b5                                   /* Timer overflow interrupt enable   */
#define TCSRV       *(volatile unsigned char *)0xFFA1              /* Timer control/status register V   */
#define TCSRV_BIT   (*(struct BIT *)0xFFA1)
#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 ADDRB       *(volatile unsigned int *)0xFFB2               /* A/D data register A             */
#define ADDRC       *(volatile unsigned int *)0xFFB4               /* A/D data register A             */
#define ADDRD       *(volatile unsigned int *)0xFFB6               /* A/D data register A             */

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

/*   Function definition                                                                              */
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;

/*   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;

                                                                                       /* Timer W initialize            */
    TCRW = 0x30;                                                                       /* Clock Select                  */
    TSRW = 0x00;                                                                       /* Clear OVF                     */
    TMRW = 0x80;                                                                       /* Timer Counter Count Start     */
    TIERW = 0x00;                                                                      /* OVF Interrupt Disable         */
    counter_sub = 15;                                                                  /* Initialize 8bit Counter_sub   */


    TCRV0 = 0x03;                                                                      /* Timer V initialize            */
    TCRV1 = 0xe3;                                                                      /* Internal clock select         */
    TCSRV = 0x10;                                                                      /* Clear OVF to 0                */

    ADCSR = 0x03;                                                                      /* 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  */


    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 ?                */
                    dig_0 = dsp_data[ADDRD >>  6 & 0x000f];                 /* Dig-0 LED display data set         */
                    dig_1 = dsp_data[ADDRD >> 10 & 0x000f];                 /* Dig-1 LED display data set         */
                    dig_2 = dsp_data[ADDRD >> 14 & 0x0003];                 /* Dig-2 LED display data set         */
                    dig_3 = 0xf6;                                           /* 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

Rev.	Date	Description	
		Page	Summary
1.00	Sep.29.03	—	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation 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 Corporation 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 Corporation or a third party.
2. Renesas Technology Corporation 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 Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation by various means, including the Renesas Technology Corporation 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 Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation 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 Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation for further details on these materials or the products contained therein.