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.

H8/300H Tiny Series

Connecting a pyroelectric infrared human body detector

Introduction

A pyroelectric infrared human body detector is connected to the analog input pin and the results of A/D conversion are displayed on the 7-segment LEDs.

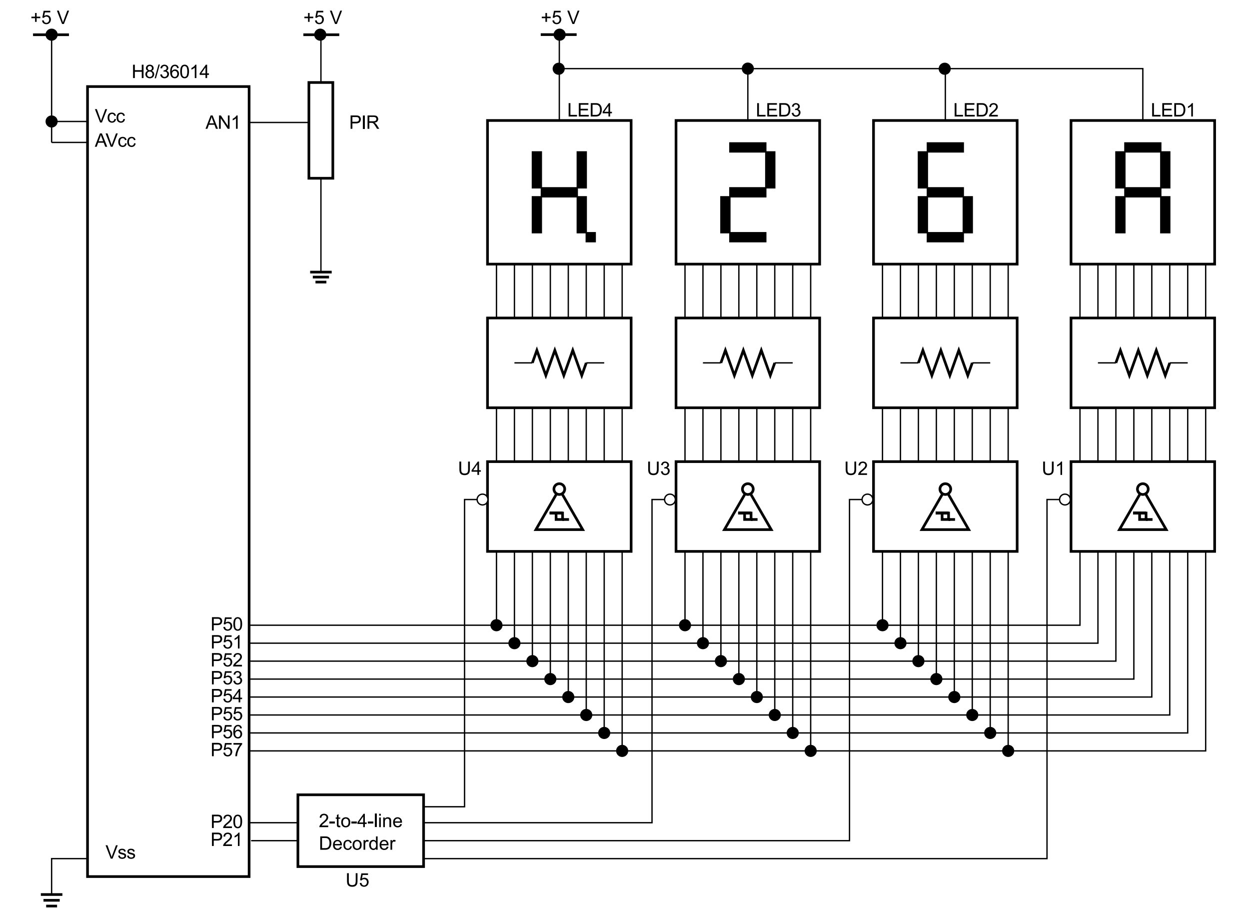
Target Device

H8/36014

Contents

1. Specifications

1. Figure 1.1 shows the hardware configuration for an example of connecting a pyroelectric infrared sensor for human body detection. As is shown in the figure, the sensor is connected to the analog input pin 1 (pin AN1).
2. A/D conversion of the AN1 pin input is performed, and the A/D conversion result is displayed on seven-segment LEDs connected to an I/O port.
3. The seven-segment LED displays shows the 10-bit data of the A/D conversion result in hexadecimal form.
4. The A/D conversion is performed every 0.49152 seconds.



[Legend]

PIR: Pyroelectric infrared human body detector (AMN21112 from Matsushita Electric Works)
LED1 to LED4: Common-anode 7-segment LEDs
U1 to U4: 3-state output inverter driver (HD74LS240)
U5: 2-to-4-line decoder (HD74HC139)

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 pyroelectric infrared human body detector used in this sample task is the pyroelectric MP motion sensor NaPiOn (model: AMN21112) from Matsushita Electric Works. The specifications of this sensor are as follows.
   A. Table 1.1 lists the ratings of the NaPiOn sensor (AMN21112).

Table 1.1 Ratings

Item		Electrical characteristics
Operating voltage	Min	4.5 V DC
	Max	5.5 V DC
Current consumption	Avg	0.17 mA
	Max	0.3 mA
Output current	Max	50μA
Output voltage	Min	0 V
	Avg	2.5 V
	Max	Operating voltage
Average output offset voltage	Min	2.3 V
	Avg	2.5 V
	Max	2.7 V
Steady-state noise	Avg	130 mVp-p
	Max	300 mVp-p
Time required for circuit stabilization after power-on	Max	45 s

   B. Figure 1.2 shows the timing chart of the NaPiOn (AMN21112).

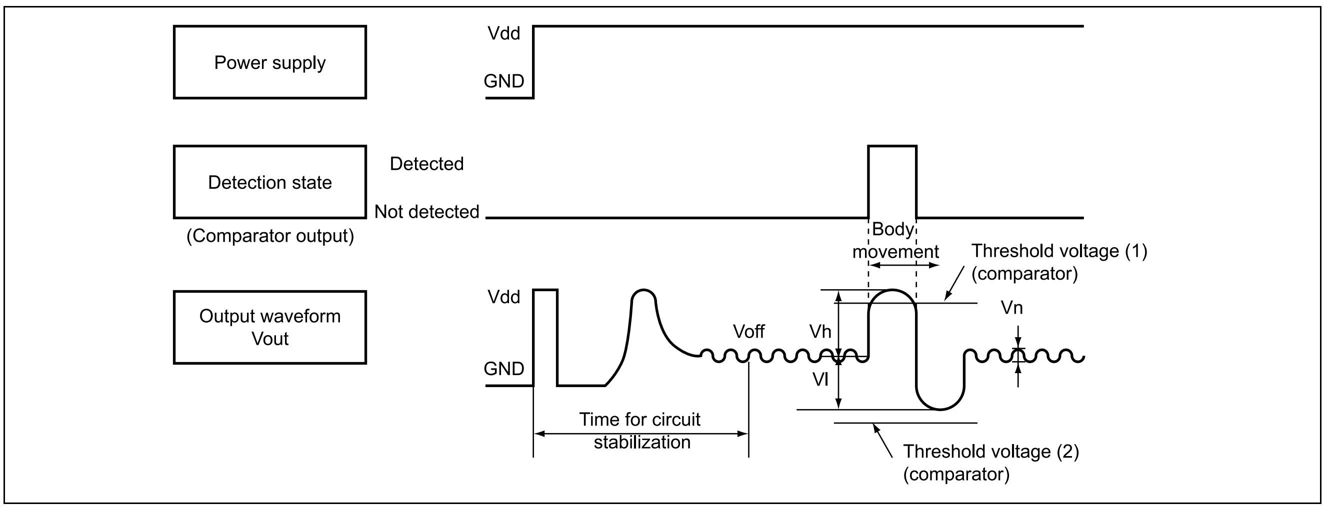


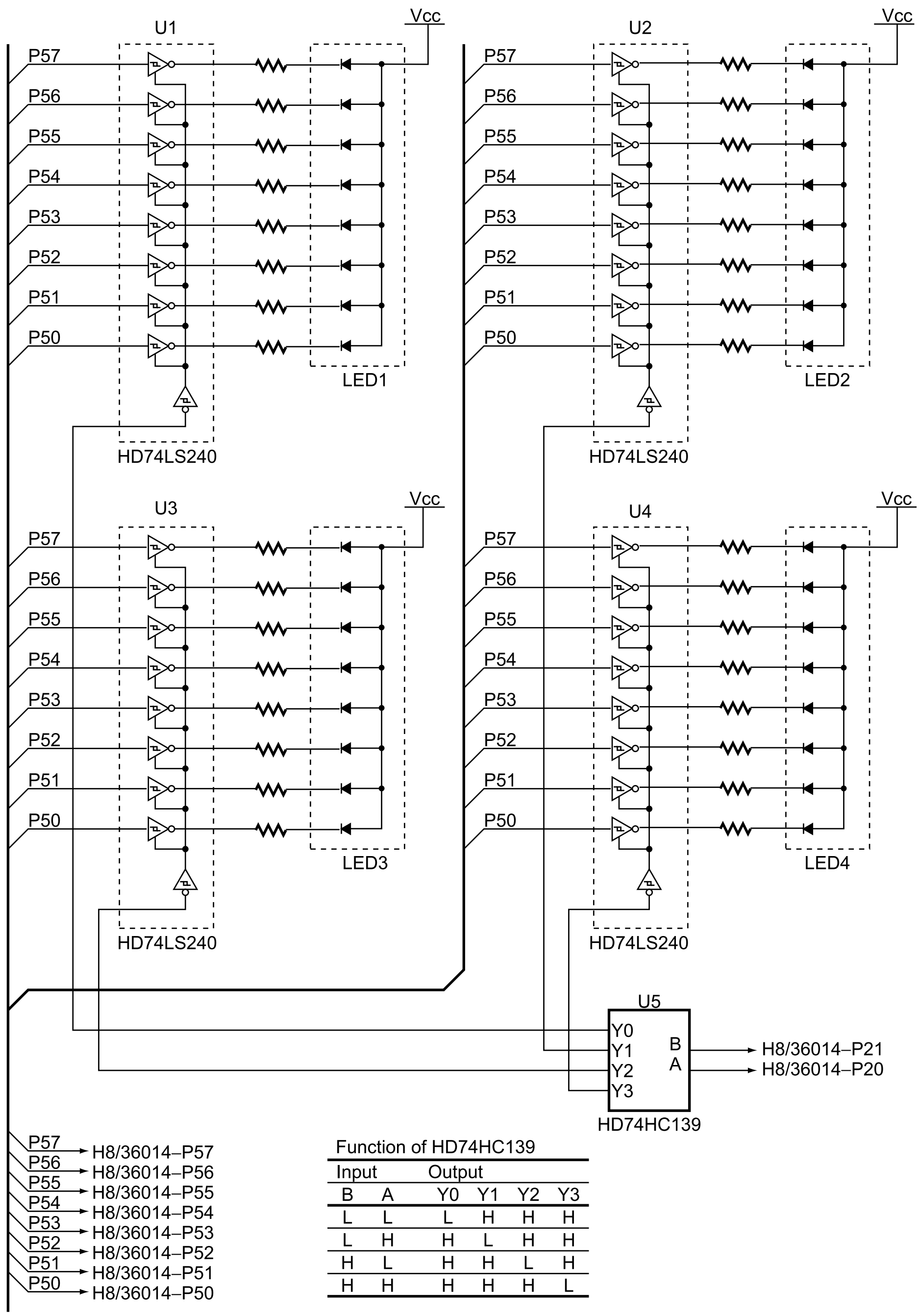
Figure 1.2 Timing Chart

7. Operation of the sample task is described below.
   A. When the sensor does not detect a human body, or when a body dose not move, there is no output from the sensor, and so a voltage equal to one-half the power supply voltage (the offset value) is input to the analog port. The offset value for the H8/36014 is 5 V/2 = 2.5 V, and the LED display is "H'200".

$$\frac{2.5\ (V) \times 1023}{5\ (V)} = 512 = H'200$$

   B. When the sensor detects a body, the LED displays a value larger than "H'200".
   C. When a body passes by, the LED display momentarily rises to a value larger than "H'200", which corresponds to the offset voltage, and then falls to a value below "H'200". Finally, the output converges on the offset voltage with the LEDs displaying "H'200".

8. In this sample task, display on the 7-segment LED is handled by attaching port outputs 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 same eight 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 a 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

9. 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.

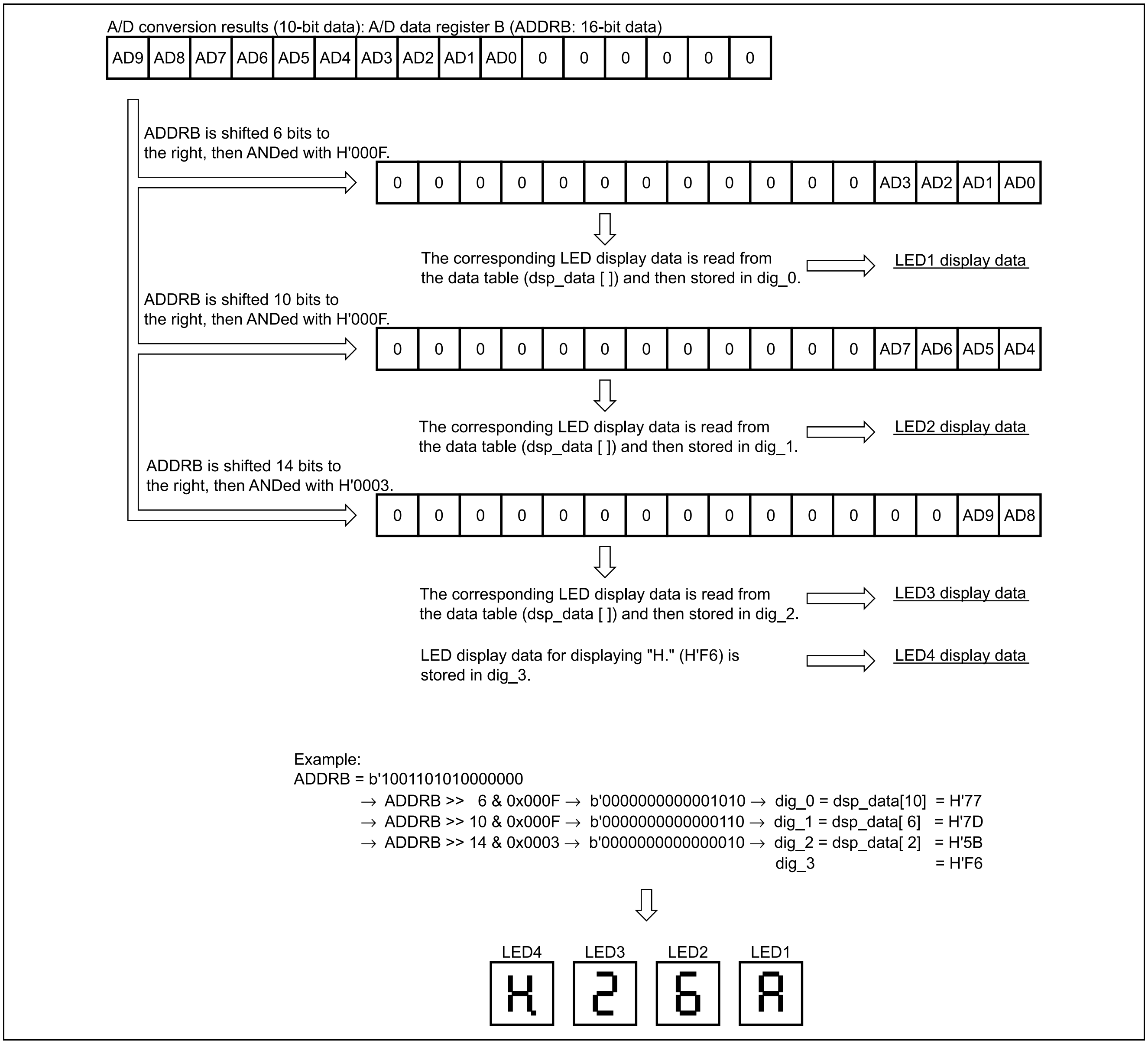


Figure 1.4 How A/D Conversion Results are Displayed on the LEDs

2. Description of Functions

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.

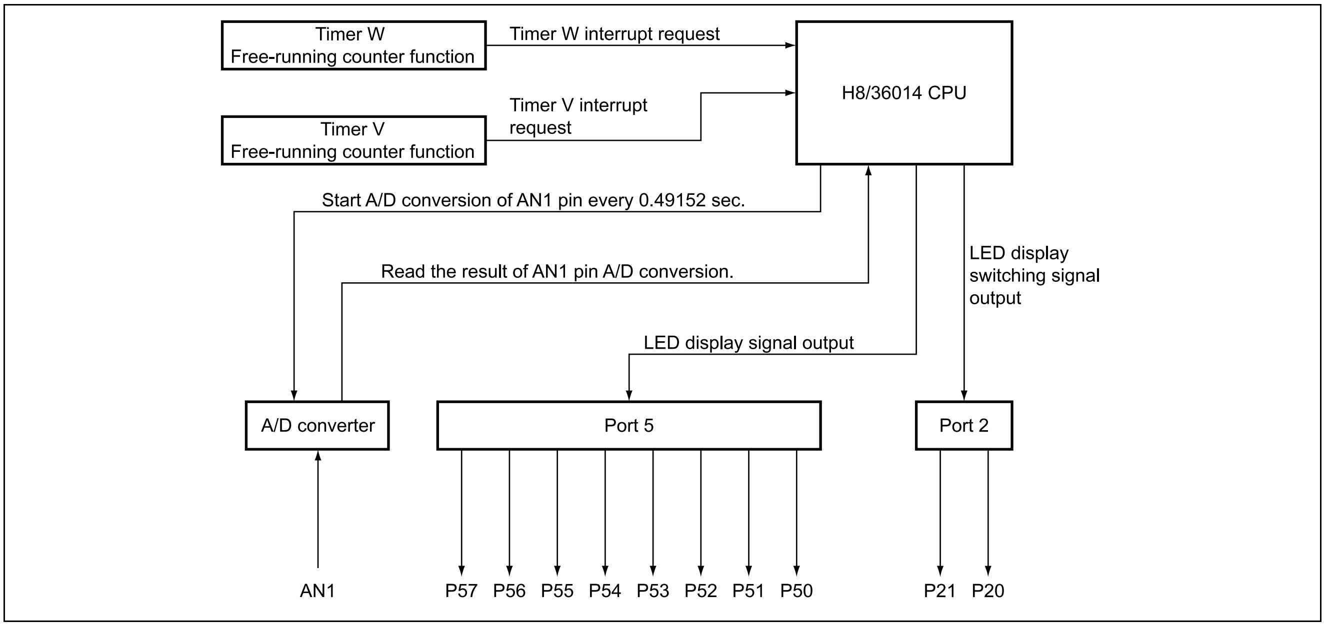


Figure 2.1 Block Diagram of the Functions in Use

Table 2.1 Function Allocation

Function	Function Allocation
Timer W	The timer W's free-running counter function is used to perform A/D conversion of the signal on the analog input pin 1 (AN1). The A/D conversion period is detected in terms of the timer W's overflow cycle of 32.768 ms.
Timer V	Timer V's free-running function is used to control switching of the 7-segment LED display. Each of the four 7-segment LEDs is lit in sequence at an interval of 2.048 ms, which is the time taken for timer V to overflow. This obtains dynamic illumination from the LEDs.
A/D converter	This unit A/D-converts the sensor output voltage that varies in response to the amount of infrared detected by the pyroelectric infrared sensor connected to analog input pin 1 (AN1) of the A/D converter.
Port 2	The active 7-segment LED display is switched by the P20 and P21 output pins of port 2. These pins are connected to the input/output pins of the 2-to-4-line decoder.
Port 5	The data for display on the currently active 7-segment LED is placed on the P50 to P57 output pins of port 5. The 10 bits of data produced by A/D conversion of the value on the AN1 pin are converted to 3 digits of hexadecimal data for display, this is then converted to obtain the data that places the corresponding figure on the LED.

2. Figure 2.2 shows how the 7-segment LED used in this task is connected. A high output from port 5 lights up the corresponding segment as shown by the figure. Table 2.2 shows the relationship between the output from port 5 and the display on the LED.

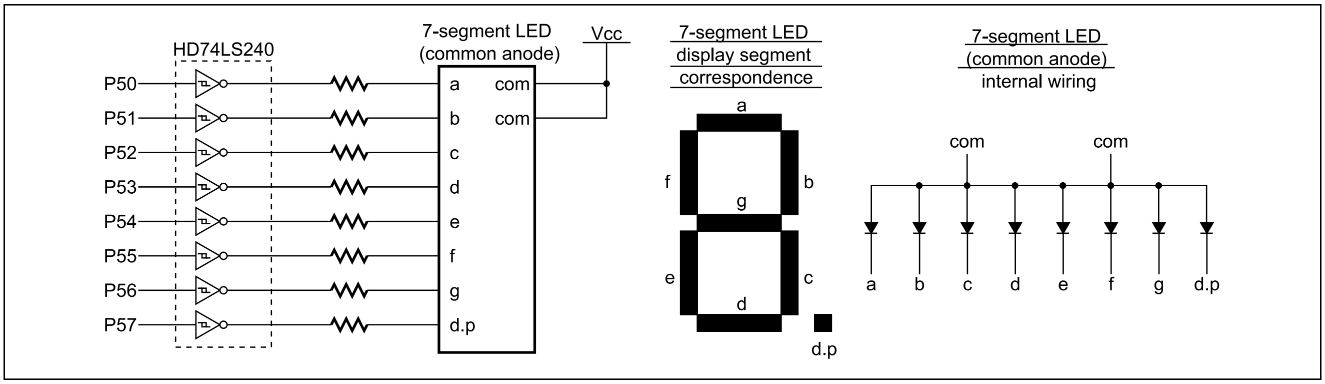
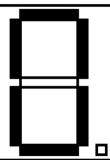
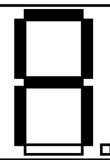
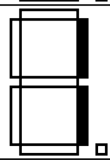
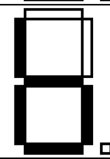
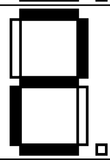
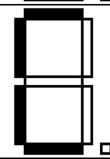
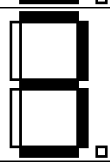
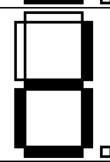
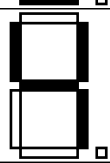
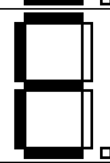
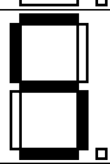
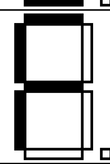
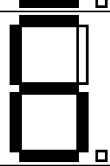
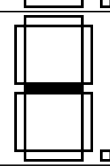
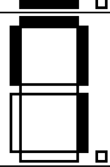
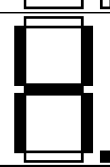
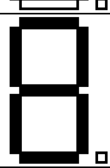
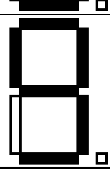


Figure 2.2 7-Segment LED Connections and Internal Wiring

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

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

3. Principle of Operation

1. Figure 3.1 shows the principle of operation in the use of timer W and A/D conversion carried out on the AN1 pin. The A/D conversion interrupt is not used in this sample task. Instead, the time to start A/D conversion (0.49152 s) and completion of A/D conversion are detected by polling of the timer W overflow flag in the tmrw routine.

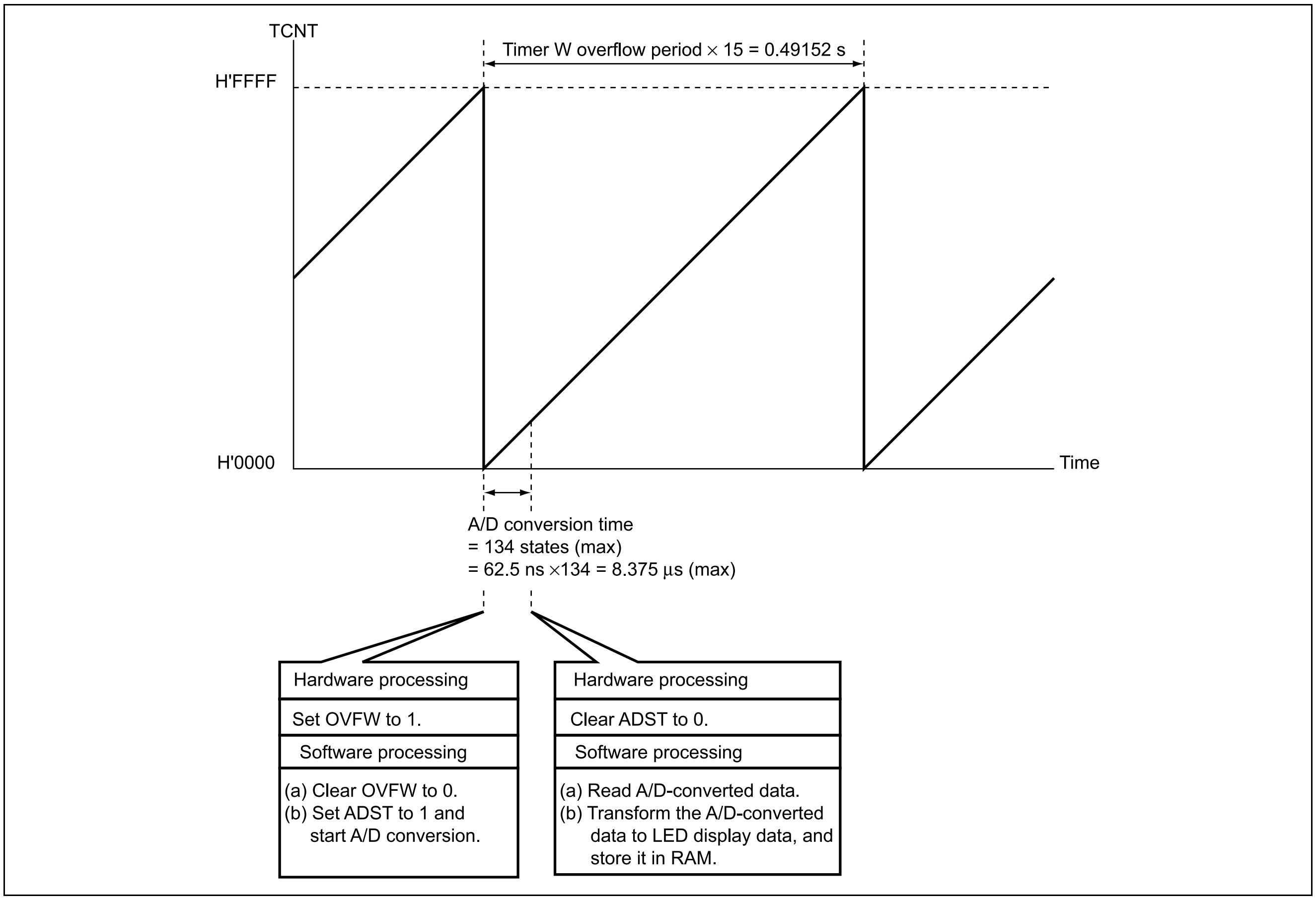


Figure 3.1 Principle of Operation of A/D Conversion of AN1-Pin Signal Using Timer W

2. The principle applied in controlling the 7-segment displays is explained below. Figure 3.2 depicts the situation where 3210 is being displayed on LED4 to LED1. As the figure shows, the next display in sequence of LED1 to LED4 is lit up each time a timer-V overflow period elapses, creating a dynamic display on the 7-segment LED.

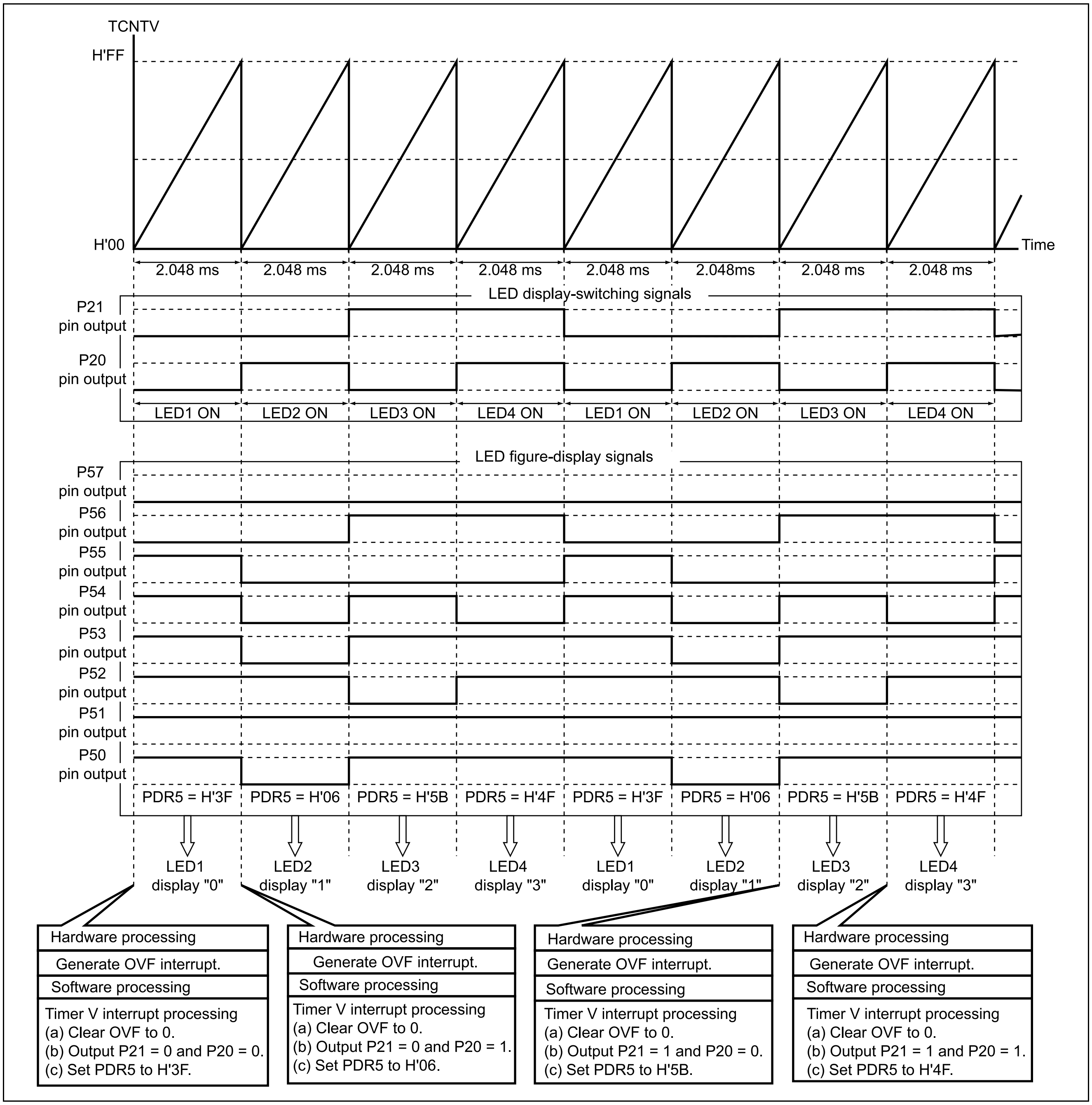


Figure 3.2 Principle of Operation of 7-Segment LED Display Control

4. Description of Software

4.1 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	Makes initial settings and enables interrupts.
Timer W interrupt processing routine	tmrw	Clears the interrupt flags, transforms A/D-converted data into LED-display data, and stores the result in RAM.
Timer V interrupt processing routine	tmrv	Clears the interrupt flags and controls output of LED-display data and switching of display from LED to LED.

4.2 Arguments

No arguments are used in this sample task.

4.3 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 condition for clearing of TCNTV, and controls the individual timer V interrupt requests.	H'FFA0	H'03 (initial setting)
	CMIEB	Compare Match Interrupt Enable B: When CMIEB = 0, interrupt requests on setting of the CMFB bit in TCSRV are disabled.	Bit 7	0
	CMIEA	Compare Match Interrupt Enable A: When CMIEA = 0, interrupt requests on setting of the CMFA bit in TCSRV are disabled.	Bit 6	0
	OVIE	Timer Overflow Interrupt Enable: When OVIE = 0, interrupt requests on setting of the OVF bit in TCSRV are disabled. When OVIE = 1, interrupt requests on setting of the OVF bit in TCSRV are enabled.	Bit 5	0/1
	CCLR1	Counter Clear 1 and 0: Sets the conditions for clearing of TCNTV. When CCLR1 = 0 and CCLR0 = 0 are both set, clearing of TCNTV is disabled.	Bit 4	0
	CCLR0		Bit 3	0
	CKS2	Clock Select 2 to 0: In combination with the ICKS0 bit of TCRV1, selects the clock signal for input to TCNTV and the condition to be counted. With CKS2 = 0, CKS1 = 1, CKS0 = 1 and ICKS0 = 1, counting by TCNTV is of the falling edges of the internal clock ϕ/128.	Bit 2	0
	CKS1		Bit 1	1
	CKS0		Bit 0	1

Register Name		Function	Address	Setting
TCSRV		Timer Control/Status Register V: Holds the overflow and compare-match status flags for timer V and controls output in response to a match.	H'FFA1	H'10
	CMFB	Compare Match Flag B: When the values of TCNTV and TCORB match, CMFB is set to 1.	Bit 7	0
	CMFA	Compare Match Flag A: When the values of TCNTV and TCORA match, CMFA is set to 1.	Bit 6	0
	OVF	Timer Overflow Flag: When the value of TCNTV overflows, OVF is set to 1. This bit is cleared by reading it while it is set and then writing a 0 to it.	Bit 5	0
	OS3	Output Select Bits 3 and 2:	Bit 3	0
	OS2	Selects the output level on the TMOV pin in response to a match for compare-match B. When OS3 = 0 and OS2 = 0: No change	Bit 2	0
	OS1	Output Select Bits 1 and 0:	Bit 1	0
	OS0	Selects the output level on the TMOV pin in response to a match for compare-match A. When OS = 0 and OS0 = 0: No change	Bit 0	0
TCRV1		Timer Control Register V1: Selects the input edge of the TRGV pin, enables TRGV input, and selects the TCNTV input clock.	H'FFA5	H'E3
	TVEG1	TRGV Input Edge Selects 1 and 0:	Bit 4	0
	TVEG0	Selects the input edge of the TRGV pin. In this case TVEG1 = 0 and TVEG0 = 0, so trigger input from the TRGV pin is disabled.	Bit 3	0
	TRGE	TRGV Input Enable: Enables and disables incrementing of TCNTV count on the edge input selected by TVEG1 and TVEG0. In this case TRGE = 0, so counting by TCNTV is not affected by the input to the TRGV pin.	Bit 2	0
	ICKS0	Internal Clock Select 0: Selects the clock input to TCNTV and condition counted, in combination with the CKS2 to CKS0 bits of TCRV0. In this case CKS2 = 0, CKS1 = 1, CKS0 = 1 and ICKS0 = 1, so counting by TCNTV is of the falling edges of the internal clock $\phi/128$.	Bit 0	1
TMRW		Timer mode register W: Selects general register functions and the 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

Register Name		Function	Address	Setting
TCRW		Timer control register W: Selects the input clock for the counter and sets counter clearing conditions and the level of timer output.	H'FF81	H'30
	CKS2	Clock select:	Bit 6	0
	CKS1	When CKS2 = 0, CKS1 = 1 and CKS0 = 1, specifies the TCNT	Bit 5	1
	CKS0	input clock as the system clock divided by 8.	Bit 4	1
TIERW		Timer interrupt enable register W: Controls timer W interrupt requests.	H'FF82	H'00 (at 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
TSRW		Timer status register W Indicates interrupt request statuses.	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 upward counter which takes as input the system clock 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'01 (initial setting)
	ADF	A/D End Flag: On completion of A/D conversion in single mode, ADF is set to 1. Clear this bit by writing 0 to it after reading it as 1.	Bit 7	0
	ADIE	A/D Interrupt Enable: When ADIE = 0, A/D Conversion End interrupt requests by the ADF are disabled.	Bit 6	0
	ADST	A/D Start: Initiates A/D conversion by setting this bit to 1. This is automatically cleared to 0 on completion of A/D conversion in single mode.	Bit 5	0/1
	SCAN	Scan Mode: When SCAN = 0, A/D conversion is in the single mode.	Bit 4	0
	CKS	Clock Select: When CKS = 0, the A/D conversion period = 134 cycles (max.).	Bit 3	0
	CH2	Channel Select Bits 2 to 0:	Bit 2	0
	CH1	Select the analog input channel.	Bit 1	0
	CH0	CH2 = 0, CH1 = 0 and CH0 = 1 (when SCAN = 0) are set to select AN1.	Bit 0	1
ADCR		A/D Control Register: Enables the start of A/D conversion in response to an external trigger.	H'FFB9	H'7E
	TRGE	Trigger Enable: When TRGE = 0, starting A/D conversion in response to edge input on an external trigger pin ($\overline{ADTRG}$) is disabled.	Bit 7	0

Register Name		Function	Address	Setting
ADDRB		A/D data register B: Stores the results of A/D conversion of the signal on AN1.	H'FFB2	—
PCR2		Port control register 2: Provides pin-by-pin control of input/output selection for the pins of port 2 that have been set as general purpose I/O pins. When PCR2 = H'03, pin P22 functions as a general-purpose input pin, while P21 and P20 function as general-purpose 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 switching: POF7 = 0 selects the general-purpose I/O port function for P57.	Bit 7	0
	POF6	P56 pin function switching: POF6 = 0 selects the general-purpose I/O port function for P56.	Bit 6	0
	WKP5	P55/$\overline{WKP5}$/$\overline{ADTRG}$ Pin Function Switching: WKP5 = 0 selects the general-purpose I/O port function for P55.	Bit 5	0
	WKP4	P54/$\overline{WKP4}$ Pin Function Switching: WKP4 = 0 selects the general-purpose I/O port function for P54.	Bit 4	0
	WKP3	P53/$\overline{WKP3}$ Pin Function Switching: WKP3 = 0 selects the general-purpose I/O port function for P53.	Bit 3	0
	WKP2	P52/$\overline{WKP2}$ Pin Function Switching: WKP2 = 0 selects the general-purpose I/O port function for P52.	Bit 2	0
	WKP1	P51/$\overline{WKP1}$ Pin Function Switching: WKP1 = 0 selects the general-purpose I/O port function for P51.	Bit 1	0
	WKP0	P50/$\overline{WKP0}$ Pin Function Switching: WKP0 = 0 selects the general-purpose I/O port function for P50.	Bit 0	0
PUCR5		Port pull-up control register 5: Provides pin-by-pin control of the pull-up MOSFET for the pins of port 5 that have been set as inputs. When PUCR5 = H'00, the pull-up MOSFETs of the P55 to P50 pins are turned off.	H'FFD1	H'00
PDR5		Port data register 5: General I/O port data register for port 5	H'FFD8	H'00
PCR5		Port control register 5: Provides pin-by-pin control of input/output selection for the pins of port 2 that have been set as general purpose I/O pins. When PCR5 = H'FF, the pins P57 to P50 function as general-purpose 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	Stores LED1 display data. (1 byte)	H'FB80	main, tmrw
dig_1	Stores LED2 display data. (1 byte)	H'FB81	main, tmrw
dig_2	Stores LED3 display data. (1 byte)	H'FB82	main, tmrw
dig_3	Stores LED4 display data. (1 byte)	H'FB83	main, tmrw
cnt	8-bit counter used in switching display from LED1 to LED4. (1 byte)	H'FB84	main, tmrv
counter_sub	8-bit counter used for adjustment of A/D data acquisition intervals. (1 byte)	H'FB85	main, tmrw

4.5 Description of Data Table

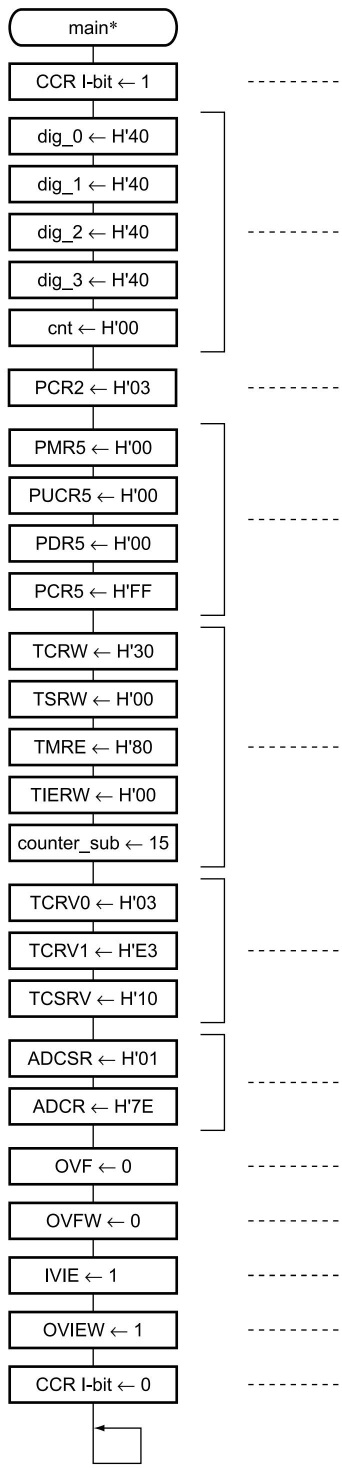
In this sample task, display data for the 7-segment LED displays are stored in the ROM as a 1-dimensional array (data table). Table 4.4 describes the data table of display data (dsp_data [ ]).

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"	1 byte	H'021C
dsp_data[1]	H'06	Data output from port 5 to display "1"	1 byte	H'021D
dsp_data[2]	H'5B	Data output from port 5 to display "2"	1 byte	H'021E
dsp_data[3]	H'4F	Data output from port 5 to display "3"	1 byte	H'021F
dsp_data[4]	H'66	Data output from port 5 to display "4"	1 byte	H'0220
dsp_data[5]	H'6D	Data output from port 5 to display "5"	1 byte	H'0221
dsp_data[6]	H'7D	Data output from port 5 to display "6"	1 byte	H'0222
dsp_data[7]	H'27	Data output from port 5 to display "7"	1 byte	H'0223
dsp_data[8]	H'7F	Data output from port 5 to display "8"	1 byte	H'0224
dsp_data[9]	H'6F	Data output from port 5 to display "9"	1 byte	H'0225
dsp_data[10]	H'77	Data output from port 5 to display "A"	1 byte	H'0226
dsp_data[11]	H'7C	Data output from port 5 to display "b"	1 byte	H'0227
dsp_data[12]	H'39	Data output from port 5 to display "C"	1 byte	H'0228
dsp_data[13]	H'5E	Data output from port 5 to display "d"	1 byte	H'0229
dsp_data[14]	H'79	Data output from port 5 to display "E"	1 byte	H'022A
dsp_data[15]	H'71	Data output from port 5 to display "F"	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/$\overline{\text{WKP0}}$: P50 output pin (output data = 0)
P51/$\overline{\text{WKP1}}$: P51 output pin (output data = 0)
P52/$\overline{\text{WKP2}}$: P52 output pin (output data = 0)
P53/$\overline{\text{WKP3}}$: P53 output pin (output data = 0)
P54/$\overline{\text{WKP4}}$: P54 output pin (output data = 0)
P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$: 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/8
Clear timer overflow flag (OVF) to "0".
Start timer counter's counting operation.
Disable interrupt requests on setting of OVF flag in TSRW.
Initialize 8-bit counter_sub 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: AN1
A/D conversion time: 134 states (max)

Clear the timer overflow flag (OVF) to 0.

Clear the timer overflow flag (OVF) to 0.

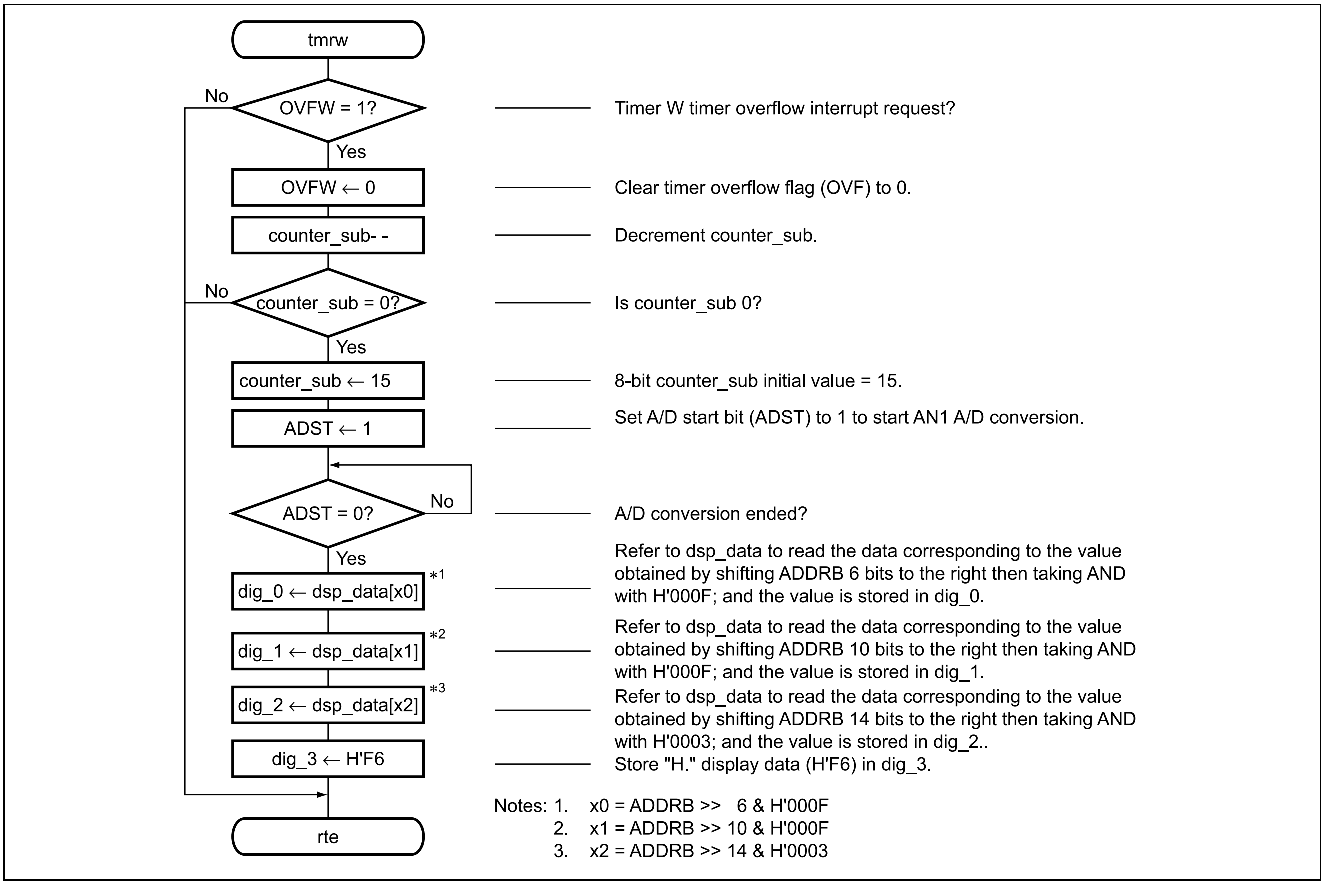
Enable timer V timer overflow interrupt requests.

Enable timer W timer overflow interrupt requests.

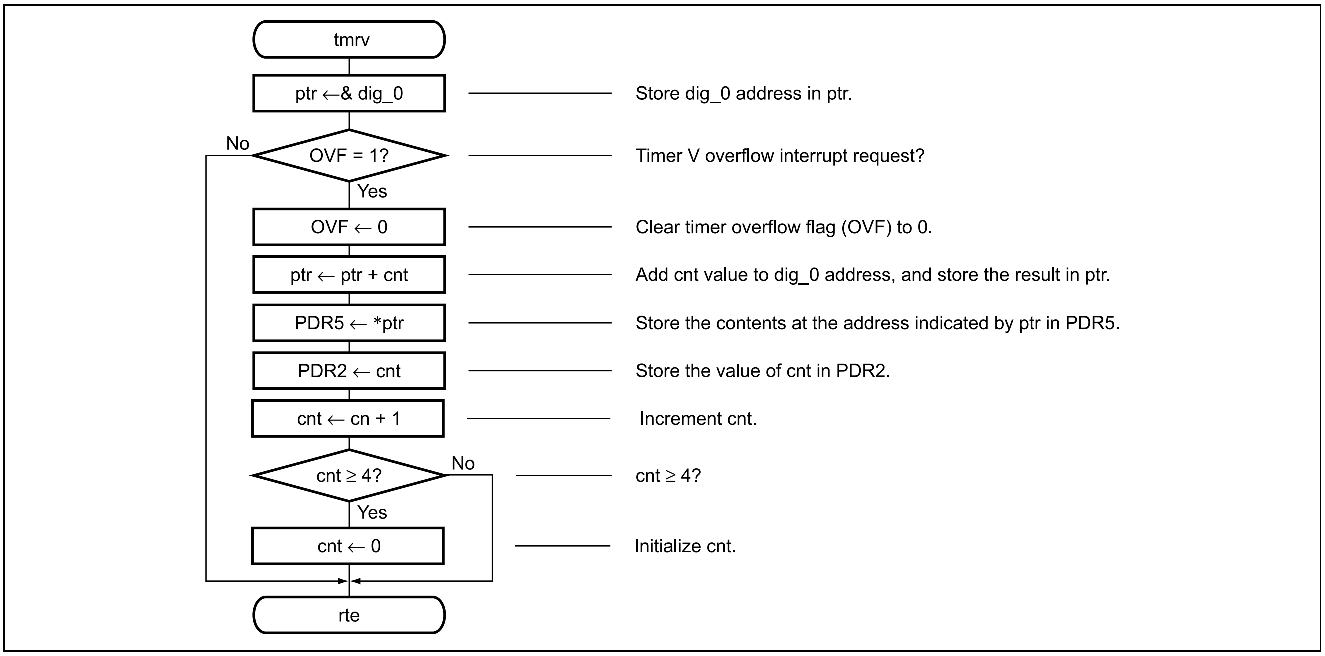
Clear the interrupt mask bit (I) in CCR to 0.

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 motion sensor                                                                            */

#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 = 0x01;                                          /* 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[ADDRB >>  6 & 0x000f]; /* Dig-0 LED display data set        */
                dig_1 = dsp_data[ADDRB >> 10 & 0x000f]; /* Dig-1 LED display data set        */
                dig_2 = dsp_data[ADDRB >> 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.