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.

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.

H8/300L SLP Series

Example of Connecting the GYROSTAR Angular Velocity Sensor

Introduction

The output of the angular velocity sensor free from the DC-component is amplified by a DC amplification circuit to the required voltage. The amplified signal is input to analog input pin AN0 and A/D converted. You have the angular velocity in decimal number (deg/sec) on an array of 7-segment LEDs.

Target Device

H8/300L Super Low Power Series H8/38024 CPU

Contents

1. Specifications

1. Figure 1 shows an example hardware configuration for connecting a piezoelectric gyroscope. As shown in this figure, the sensor is connected to analog input pin 0 (AN0 pin).
2. The signal input to the AN0 pin is A/D converted and the result is then displayed on an array of 7-segment LEDs connected to the I/O ports.
3. The 7-segment LEDs display the 10-bit result of A/D conversion that indicates a decimal angular velocity. A/D conversion is performed at intervals of 0.5 seconds.

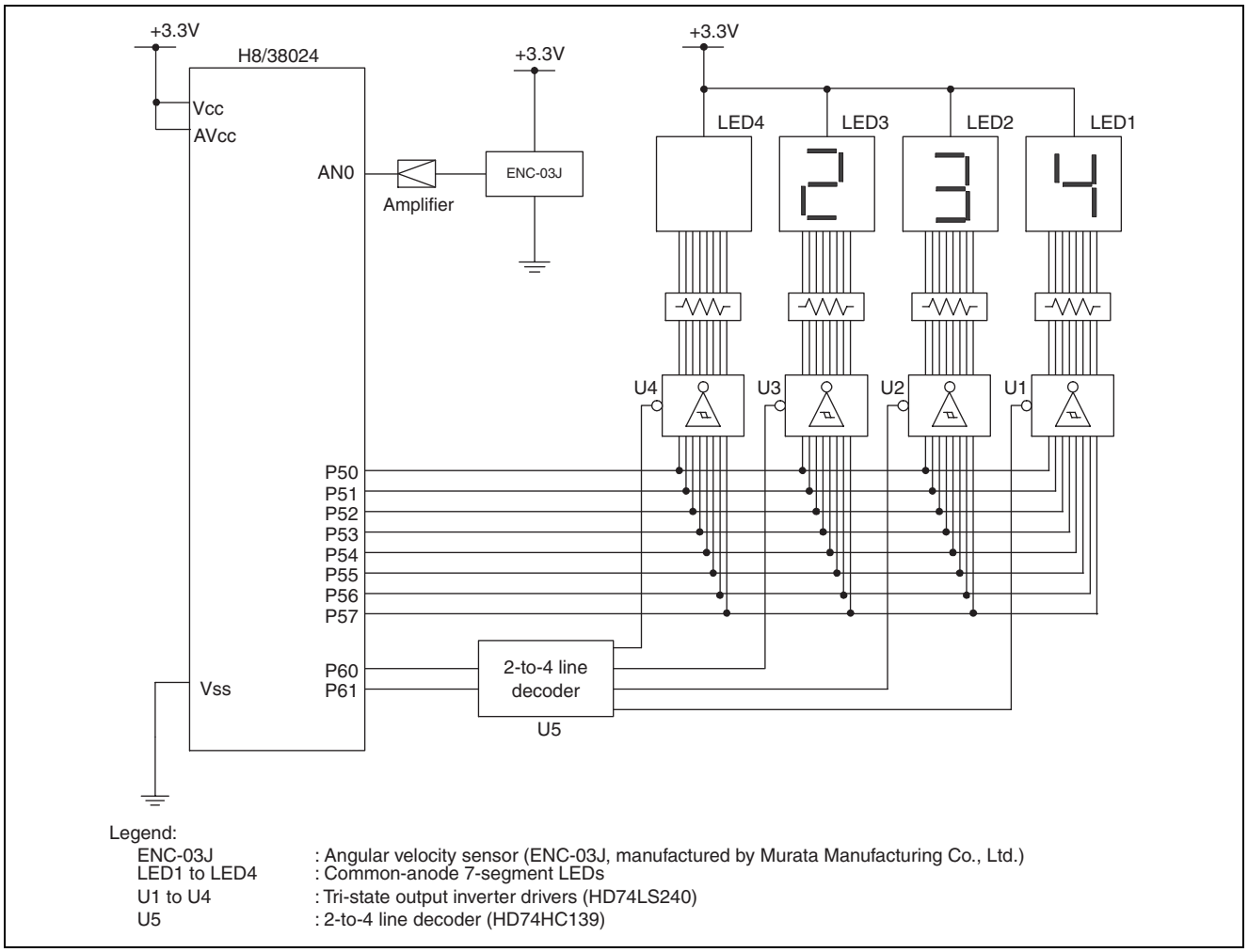


Figure 1 Hardware Configuration

4. In this sample task, the operating voltage (Vcc) and analog power supply voltage (AVcc) of the H8/38024 are both 3.3 V, the OSC clock frequency is 10 MHz, and the watch clock frequency is 32.768 kHz.
5. The sensor used in this sample task is a piezoelectric gyroscope (model ENC-03J) manufactured by Murata Manufacturing Co., Ltd. The specifications of the sensor are listed below.

A. Table 1 lists the specifications of the piezoelectric gyroscope

Table 1 Specifications of the ENC-03J (Reference Values)

Item	Specification
Operating voltage	+3.0 VDC
Current consumption	4.5 mA (max.) (power supply voltage: +3.0 VDC)
Detection range	±300 deg/sec
Sensitivity	0.67 mV/(deg/s) ±20% (initial deviation)
Output in the stationary state	Reference output ±0.6 VDC (within operating temperature range)
Reference output	1.35 ±0.1 VDC (within operating temperature range)
Linearity	Within ±5.0%
Sensitivity fluctuation with temperature	-20 to +10% (within operating temperature range, reference temperature: 25°C)
Response	DC to 50 Hz
Operating temperature range	-5 to +75°C
Storage temperature range	-30 to +85°C
Dimensions	15.5 × 8.0 × 4.3mm
Weight	1.0 g (max.)

Notes: 1. The above values are typical.
2. Unless otherwise specified, these values are defined at ambient temperature (Ta = 25°C).
3. The reference output is grounded through a 4.7-µF capacitor.

B. Figure 2 shows a standard sensor amplification circuit.
This circuit removes the DC and high-frequency noise components from the sensor output. When using this sample circuit, select the constant according to the operation to be detected.

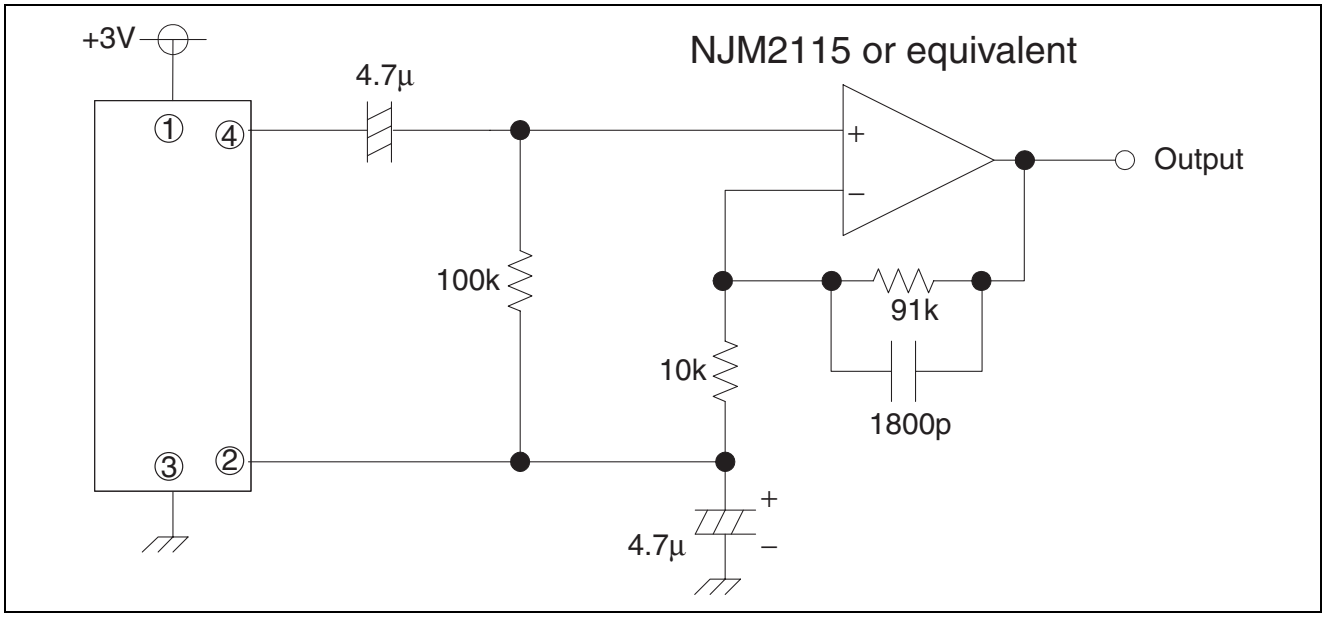


Figure 2 Sensor Standard Amplification Circuit (Sample Circuit)

6. The ENC-03J piezoelectric gyroscope is an oscillation angular velocity sensor that detects the Coriolis force from the oscillation of a ceramic bimorph element. This sensor has a response range from DC to 50 Hz and is suitable for detecting movement, such as that of a video camera or that of pointing devices or HMDs.
7. The circuit in this sample task operates as follows.
   A. The sensor output voltage from the ENC-03J is amplified by the DC amplification circuit. The amplified signal is input to analog input pin 0 (AN0) and then A/D converted.
   B. While the sensor is stationary, the reference voltage (1.35 VDC) shown in Table 1 is input to analog input pin 0 (AN0 pin), "···0" (deg/sec) is displayed on the LED display.
   C. The value "234" in Figure 1 indicates that the angular velocity is 234 (deg/sec).
   D. If the sensor is rotated in the opposite direction, a negative value such as "−178" is displayed, indicating an angular velocity of 178 (deg/sec) in the opposite direction.
8. In this sample task, the 7-segment LED display is set up by connecting the port outputs to the tri-state output inverter drivers (HD74LS240) and connecting the driver outputs to the cathodes of the 7-segment LEDs. In addition, all the ports used for the four 7-segment LEDs are connected to the 7-segment LEDs and the enable pins of the tri-state inverter drivers are used to renew between the 7-segment LEDs. The signal generation for renewing between the LEDs is controlled by the two port outputs of a 2-to-4 line decoder (HD74HC139). Figure 3 illustrates the method of controlling the 7-segment LEDs.

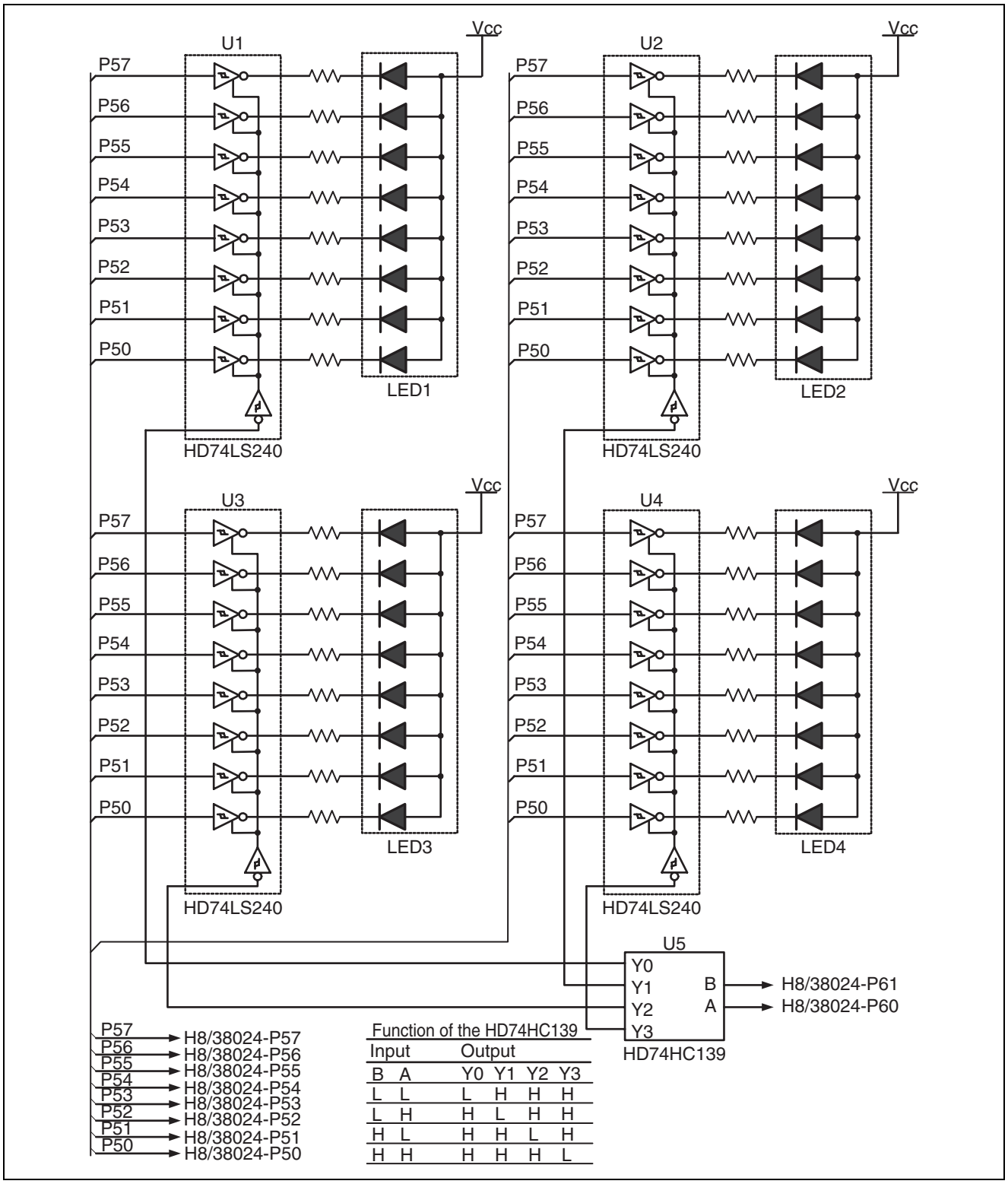


Figure 3 Control of 7-Segment LEDs

9. In this sample task, a decimal angular velocity (deg/sec) is displayed on the 7-segment LEDs. Figure 4 illustrates how the angular velocity is displayed on the LEDs.

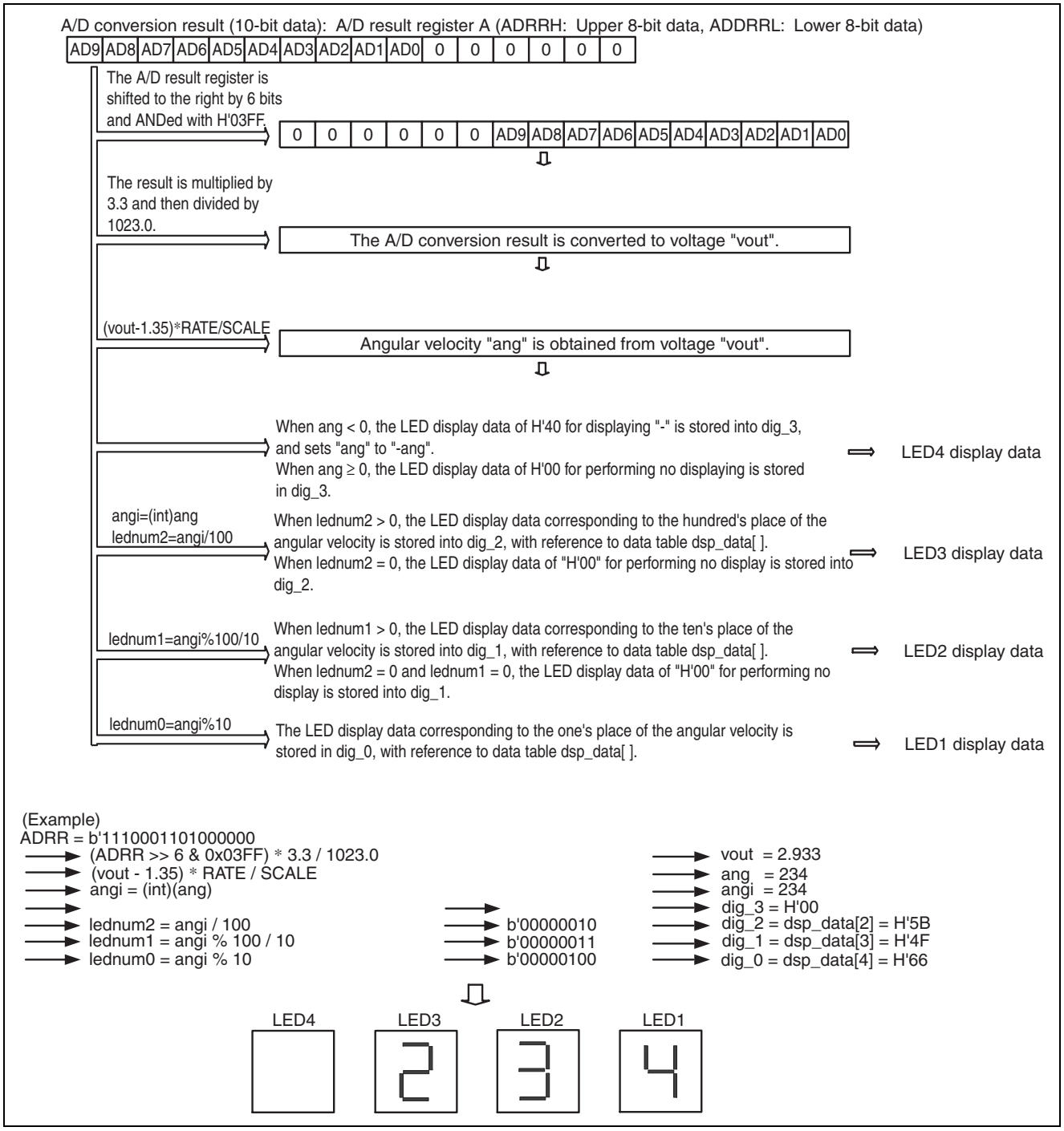


Figure 4 Display of Angular Velocity on the LEDs

2. Description of Functions

1. Figure 6 is a block diagram of the H8/38024 functions used in this task. Table 2 lists the function allocations.

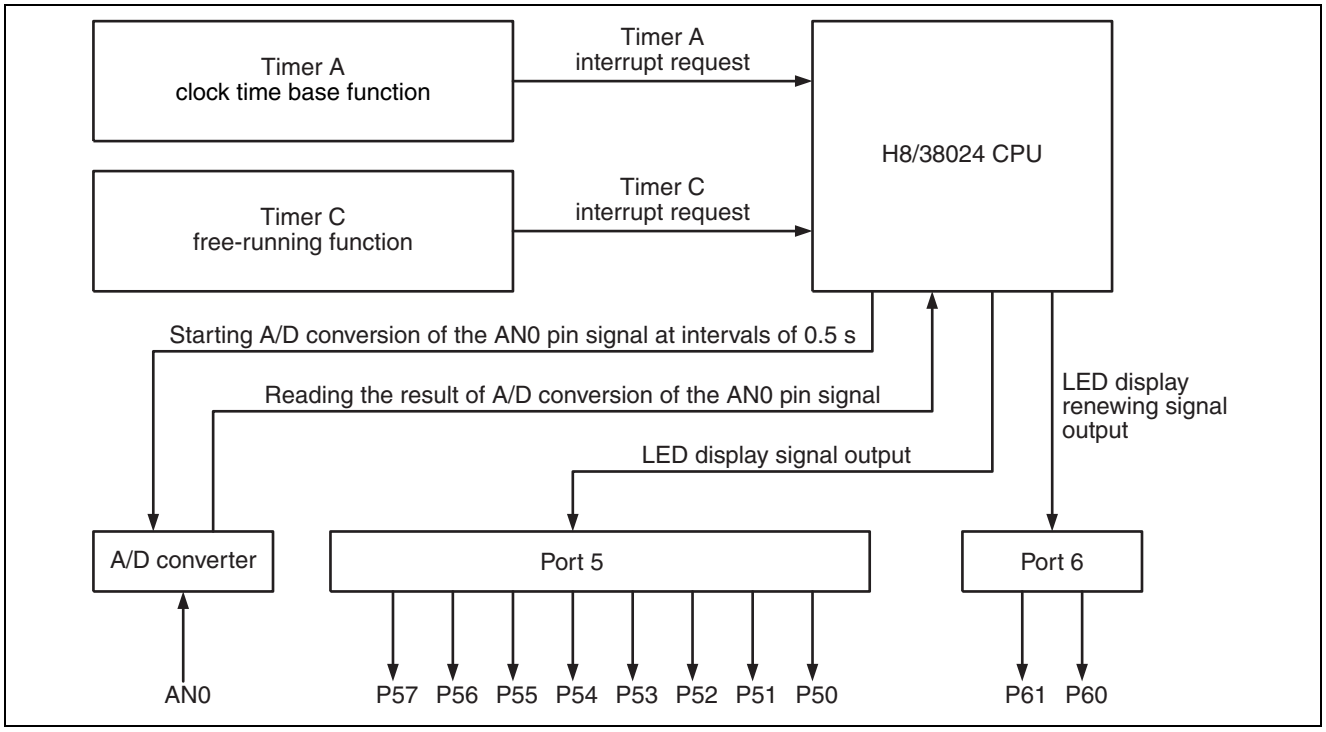


Figure 5 Block Diagram of Functions Used in This Sample Task

Table 2 Function Assignment

Function	Function assignment
Timer A	The clock time base function of timer A is used to measure the 0.5-second interval at which the A/D conversion of the analog input pin 0 (AN0) signal is performed. Timer A interrupt is used to initiate A/D conversion is performed.
Timer C	The free-running function of timer C is used to control the renewing between the 7-segment LEDs. Each of the four 7-segment LEDs is lit in sequence at intervals of 3.2768 ms (the time at which timer C overflows), enabling dynamic illumination of the LEDs.
A/D converter	This function A/D-converts the output voltage from the angular velocity sensor connected to analog input pin 0 (AN0) of the A/D converter.
Port 5	The P50-P57 output pins of port 5 are used to display data on the currently active 7-segment LED. The 10-bit data, obtained as a result of the A/D conversion of the AN0 pin signal, is converted to the angular velocity display data (with decimal places) (deg/sec) and then output to the LED.
Port 6	The P60 and P61 output pins of port 6 are used to renew between the four 7-segment LEDs. The P60 and P61 output pins are connected to the input pins of the 2-to-4 line decoder.

2. Figure 6 shows how the 7-segment LED used in this task is connected. A high output from port 5 lights the corresponding segment of the LED, as shown in the figure. Table 3 lists the relationships between the port 5 outputs and the LED display.

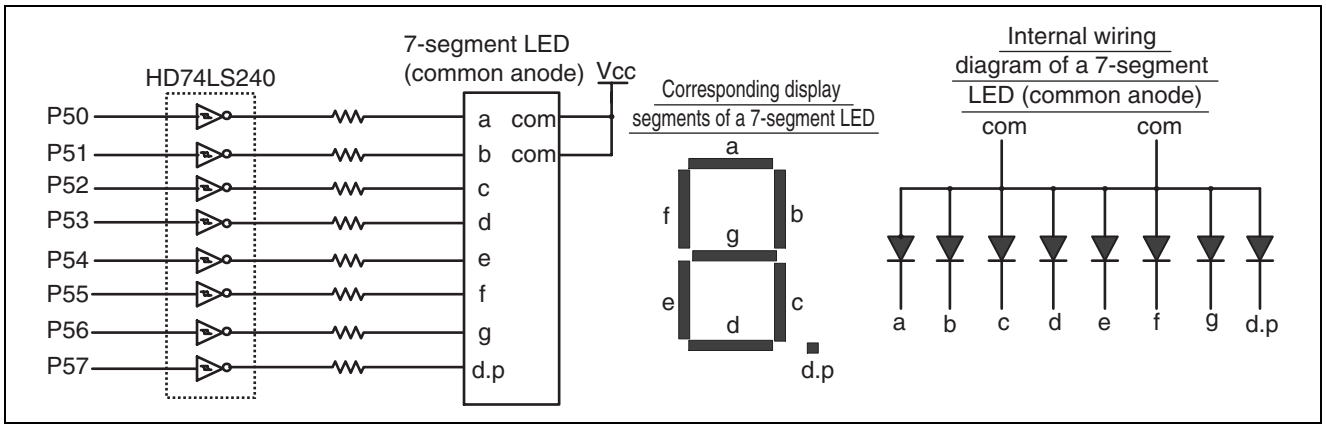
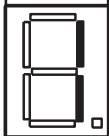
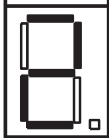
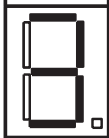
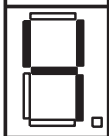
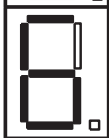
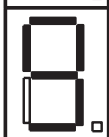


Figure 6 Connection and Internal Connections of 7-Segment LED

Table 3 Relationship between Port 5 and 7-Segment LED 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	0	0	0	0	0	0
	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									

3. Description of Operation

1. Figure 7 shows the description of A/D conversion of the AN0 pin signal when timer A is used. In this sample task, the completion of A/D conversion is determined by the tmra interrupt routine instead of an A/D conversion interrupt, as shown in Figure 7.

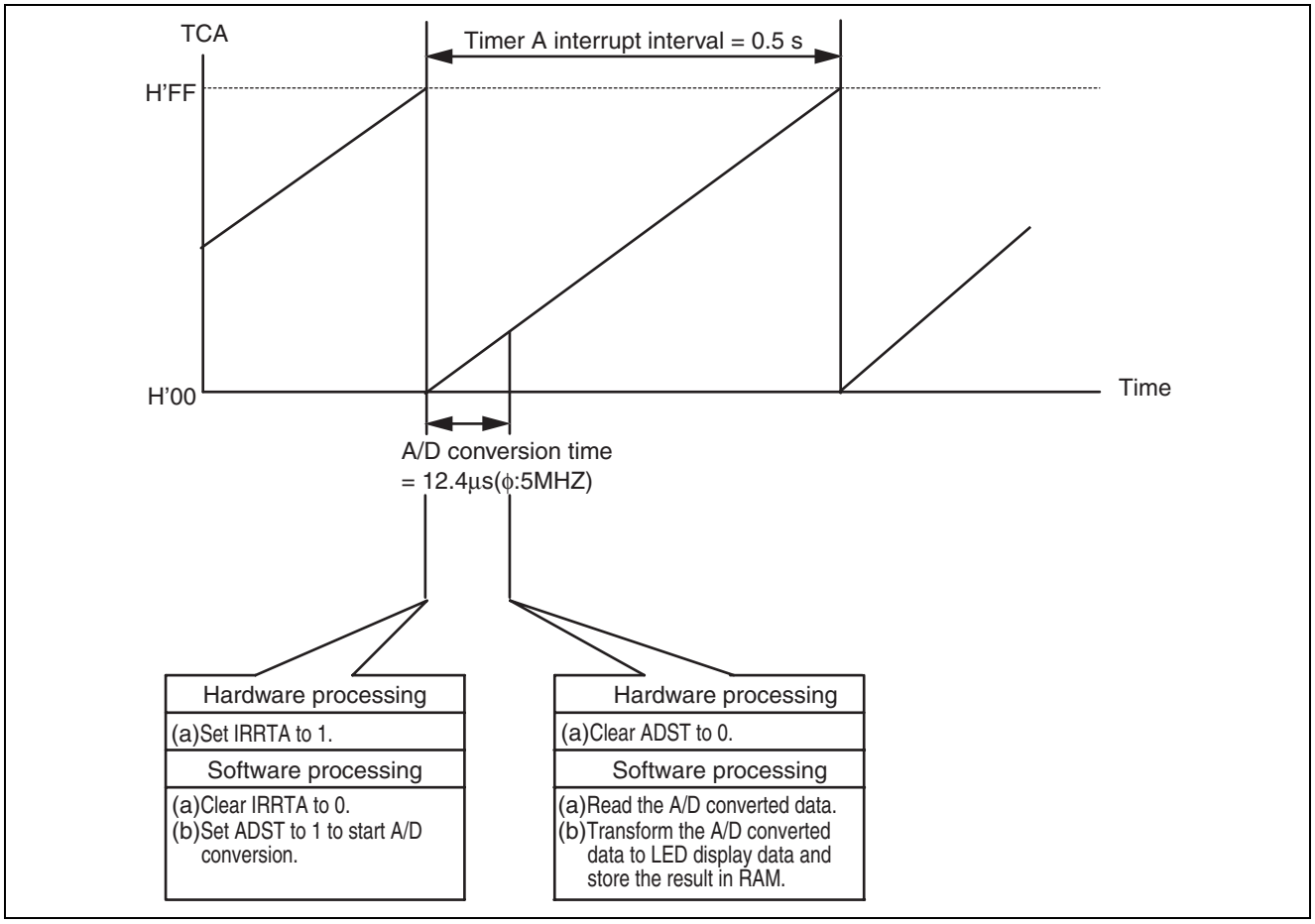


Figure 7 Description of A/D Conversion of the AN0 Pin Signal when Timer A is Used

2. The following describes the descriptions of 7-segment LED operation. Figure 8 shows how a value of ” 234” is displayed on LED4 to LED1. As shown in the figure, each of LED1 to LED4 is lit in sequence at the timer C overflow interval, resulting in dynamic display with the 7-segment LED.

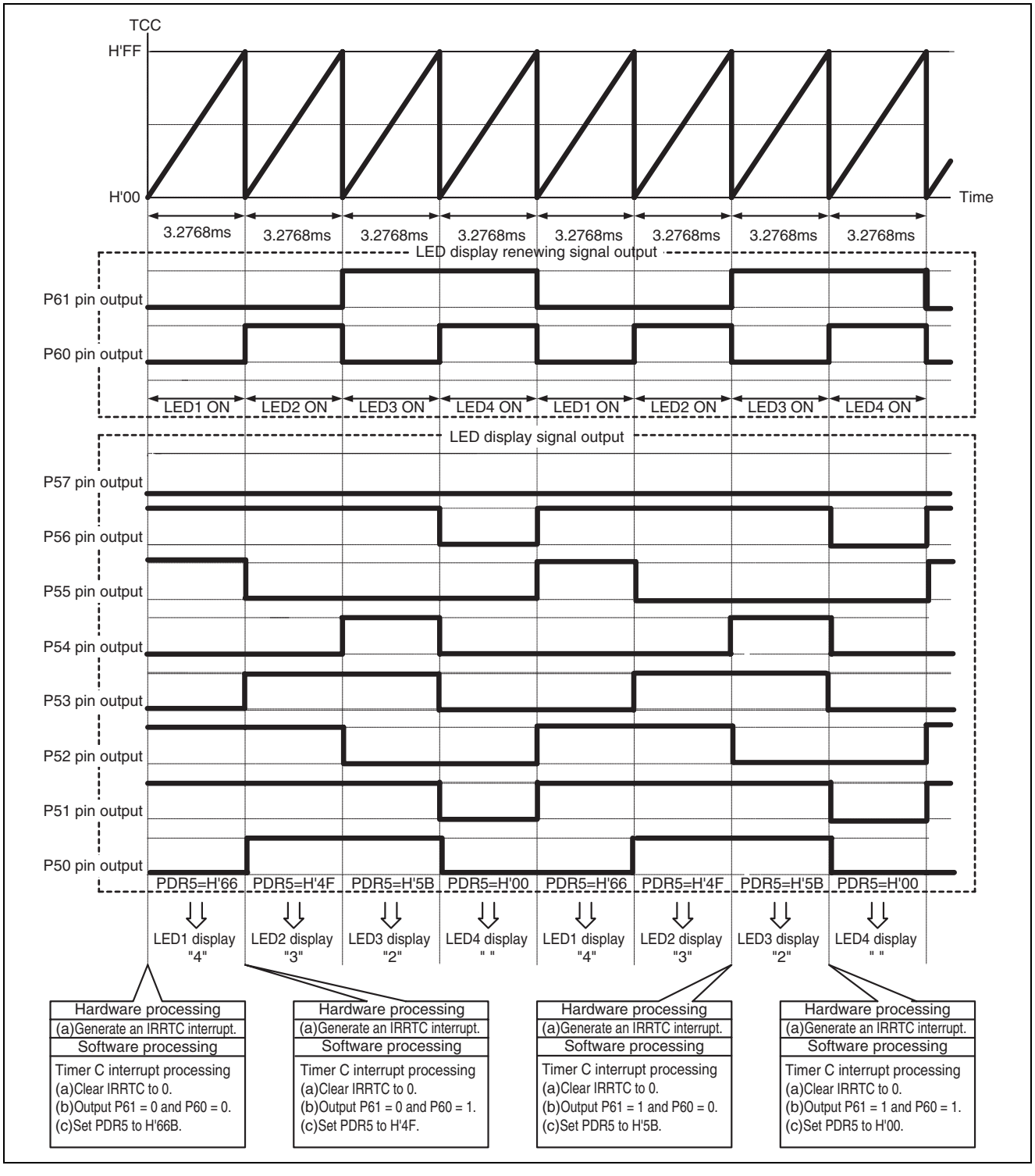


Figure 8 Description of 7-Segment LED Display Control

4. Description of Software

1. Modules
   Table 4 lists the modules used in this sample task.

Table 4 Modules

Module name	Label name	Function
Main routine	main	Makes the initial settings and enables interrupts.
Timer A interrupt processing routine	tmra	Clears the interrupt flags, transforms the A/D-converted data to LED display data, and then stores the result in RAM.
Timer C interrupt processing routine	tmrc	Clears the interrupt flags, outputs LED display data, and controls LED display renewing.

2. Arguments
   No arguments are used in this sample task.
3. Internal Registers
   Table 5 lists the internal registers used in this sample task.

Table 5 Internal Registers

Register name		Description	Address	Setting
TMA		Timer mode register A: Selects a prescaler and input clock.	H'FFB0	H'0C (At initial setting)
	TMA3	Internal clock select 3: Selects the operating mode of timer A. When TMA3 = 1, it functions as a clock time base that counts the output of prescaler W.	Bit 3	1
	TMA2	Internal clock select 2-0: When TMA3 = 1, the clock time base (32.768kHz) is selected. When TMA2 = 1, TMA1 = 0, and TMA0 = 0, TCA reset is selected. When TMA2 = 0, TMA1 = 0, and TMA0 = 1, the TCA overflow period is 0.5 s.	Bit 2	0/1
	TMA1		Bit 1	0
	TMA0		Bit 0	0/1
TMC		Timer mode register C: Selects automatic reloading, controls count-up or count-down of the counter, and controls the input clock.	H'FFB4	H'1B
	TMC7	Selects the automatic reload function: When TMC7 = 0, the interval function is selected.	Bit 7	0
	TMC6	Controls count-up or count-down: When TMC6 = 0 and TMC5 = 0, TCC functions as an up-counter.	Bit 6	0
	TMC5		Bit 5	0
	TMC2	Clock select: When TMC2 = 0, TMC1 = 1, and TMC0 = 1, counting is performed using an internal clock of ϕ/64.	Bit 2	0
	TMC1		Bit 1	1
	TMC0		Bit 0	1
TLC		Timer load register C: Sets the TCC reload value.	H'FFB5	H'00

Register name		Description	Address	Setting
AMR		A/D mode register: Sets the A/D conversion speed, selects an external trigger, and specifies an analog input pin.	H'FFC6	H'34
	CKS	Sets the A/D conversion speed (when ϕ = 5 MHz): When CKS = 0, the speed is 12.4µs.	Bit 7	0
	TRGE	Enables/disables the startup of A/D conversion by an external trigger input: When TRGE = 0, the start of A/D conversion by an external trigger input is disabled.	Bit 6	0
	CH3	Channel select 3-0: When CH3 = 0, CH2 = 1, CH1 = 0, and CH0 = 0, AN0 is selected.	Bit 3	0
	CH2		Bit 2	1
	CH1		Bit 1	0
	CH0		Bit 0	0
ADSR		A/D start register: Specifies the starting or stopping of A/D conversion.	H'FFC7	-
	ADSF	Checks the starting or stopping of A/D conversion. Read: When ADSF = 0, A/D conversion has been completed. When ADSF = 1, A/D conversion is in progress. Write: When ADSF = 0, A/D conversion is forcibly stopped. When ADSF = 1, A/D conversion is started.	Bit 7	0/1
ADRRH		A/D result register: The uppermost eight bits are stored.	H'FFC4	-
ADRRL		A/D result register: The lower two bits are stored in bit 7 and bit 6.	H'FFC5	-
PUCR6		Port pull-up control register 6: Controls, bit-by-bit, the pull-up MOS of each pin of port 6 that is set as an input port. When PUCR6 = H'00, the pull-up MOS for the P67-P60 pins is off.	H'FFE3	H'00
PDR6		Port data register 6: General I/O port data register for port 6	H'FFD9	H'00
PCR6		Port control register 6: Selects, bit-by-bit, whether each pin used as a general I/O port for port 6 is an input or output pin. When PCR6 = H'FF, the P67-P60 pins function as general output pins.	H'FFE9	H'FF
PMR5		Port mode register 5: Sets the pin function of port 5.	H'FFCC	H'00
	WKP7	P57/_WKP7/SEG7 pin function switching: When this bit is 0, the P57 general I/O port function is selected.	Bit 7	0
	WKP6	P56/_WKP6/SEG6 pin function switching: When this bit is 0, the P56 general I/O port function is selected.	Bit 6	0
	WKP5	P55/_WKP5/SEG5 pin function switching: When this bit is 0, the P55 general I/O port function is selected.	Bit 5	0

Register name		Description	Address	Setting
	WKP4	P54/_WKP4/SEG4 pin function switching: When this bit is 0, the P54 general I/O port function is selected.	Bit 4	0
	WKP3	P53/_WKP3/SEG3 pin function switching: When this bit is 0, the P53 general I/O port function is selected.	Bit 3	0
	WKP2	P52/_WKP2/SEG2 pin function switching: When this bit is 0, the P52 general I/O port function is selected.	Bit 2	0
	WKP1	P51/_WKP1/SEG1 pin function switching: When this bit is 0, the P51 general I/O port function is selected.	Bit 1	0
	WKP0	P50/_WKP0/SEG0 pin function switching: When this bit is 0, the P50 general I/O port function is selected.	Bit 0	0
PUCR5		Port pull-up control register 5: Controls, bit-by-bit, the pull-up MOS of each pin of port 5 that is set as an input port. When PUCR5 = H'00, the pull-up MOS for the P57-P50 pins is off.	H'FFE2	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: Selects, bit-by-bit, whether each pin used as a general I/O port for port 5 is an input or output pin. When PCR5 = H'FF, the P57-P50 pins function as general output pins.	H'FFE8	H'FF
IENR1		Interrupt enable register 1: Specifies whether an interrupt request is enabled or disabled.	H'FFF3	-
	IENTA	Timer A interrupt request enable: When this bit is 1, a timer A overflow interrupt request is enabled.	Bit 5	1
IRR1		Interrupt request register 1: When an interrupt request for timer A, IRQ4, IRQ3, IRQAEC, IRQ1, or IRQ0 is issued, the corresponding flag is set to 1.	H'FFF6	-
	IRRTA	Timer A interrupt request flag: This bit is set to 1 when the counter value of timer A overflows (H'FF → H'00). This bit is cleared to 0 when 0 is written to IRRTA.	Bit 7	0/1
IENR2		Interrupt enable register 2: Specifies whether an interrupt request is enabled or disabled.	H'FFF4	-
	IENTC	Timer C interrupt request enable When this bit is 1, a timer C overflow or underflow interrupt request is enabled.	Bit 1	1

Register name		Description	Address	Setting
IRR2		Interrupt request register 2: When an interrupt request for direct transition, A/D converter, timer G, timer FH, timer FL, timer C, or asynchronous event counter is issued, the corresponding flag is set to 1.	H'FFF7	-
	IRRTC	Timer C interrupt request flag: This bit is set to 1 when the counter value of timer C overflows (H'FF → H'00) or underflows (H'00 → H'FF). This bit is cleared to 0 when 0 is written to IRRTC.	Bit 7	0/1

4. Description of RAM
   Table 6 describes the RAM used in this sample task.

Table 6 Description of RAM

Label name	Description	Address	Module label name
vout	Voltage	H'FB80	tmra
ang	Angular velocity	H'FB84	tmra
angi	10-times angular velocity	H'FB88	tmra
SCALE	Scale factor	H'FB8A	main, tmra
RATE	Amplification rate	H'FB8E	main, tmra
*ptr	Location where the address of dig_0 is stored	H'FB92	tmrc
dig_0	Stores LED1 display data (1 byte)	H'FB94	main, tmra, tmrc
dig_1	Stores LED2 display data (1 byte)	H'FB95	main, tmra
dig_2	Stores LED3 display data (1 byte)	H'FB96	main, tmra
dig_3	Stores LED4 display data (1 byte)	H'FB97	main, tmra
cnt	8-bit counter for switching display between LED1-LED4 (1 byte)	H'FB98	main, tmrc
lednum0	LED1 display data	H'FB99	tmra
lednum1	LED2 display data	H'FB9A	tmra
lednum2	LED3 display data	H'FB9B	tmra

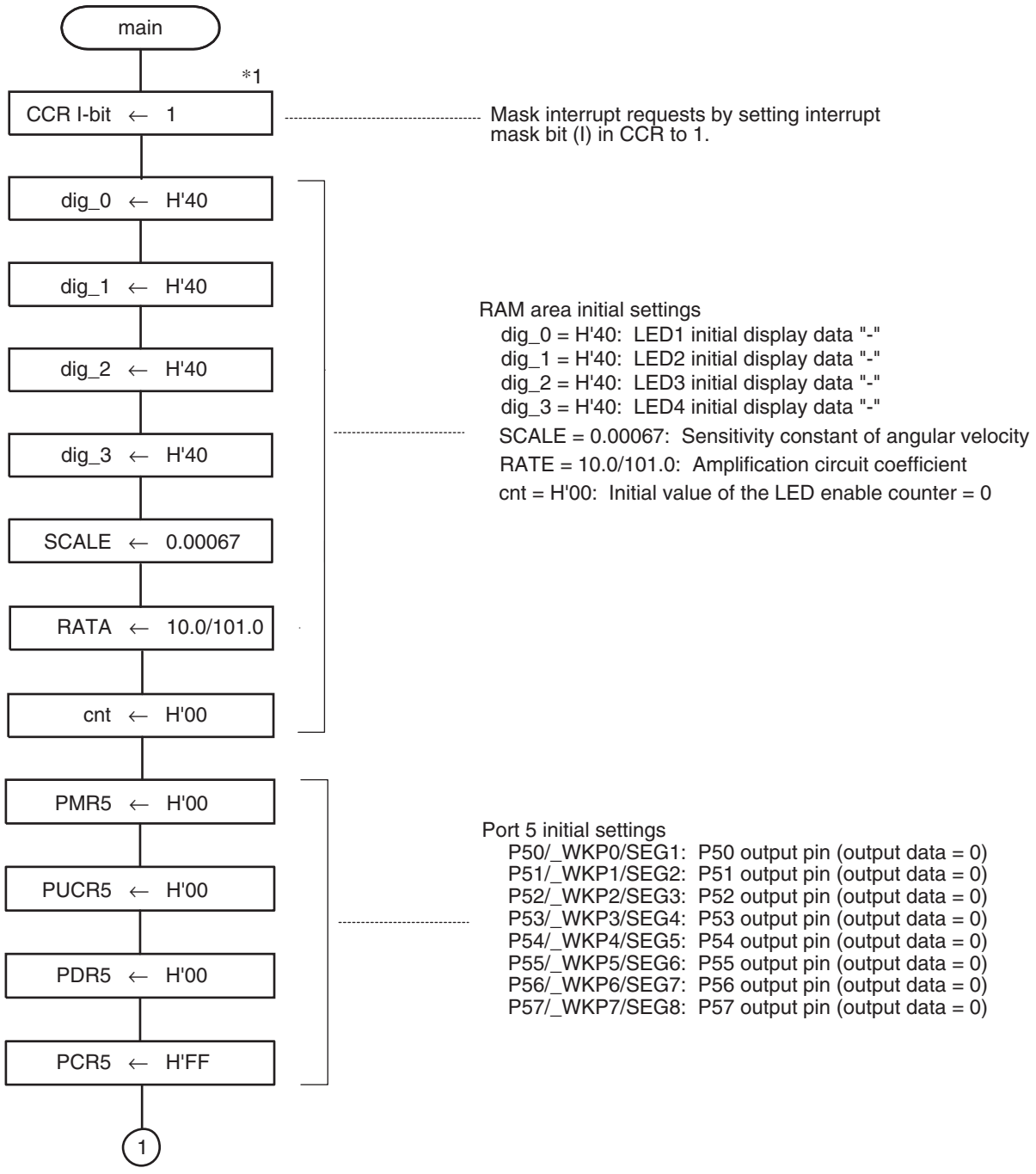
5. Description of data tables
   In this sample task, the display data for the 7-segment LEDs is stored in ROM as a 1-dimensional array (data table). Table 7 describes the data table for the 7-segment LED display (dsp_data [ ]).

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

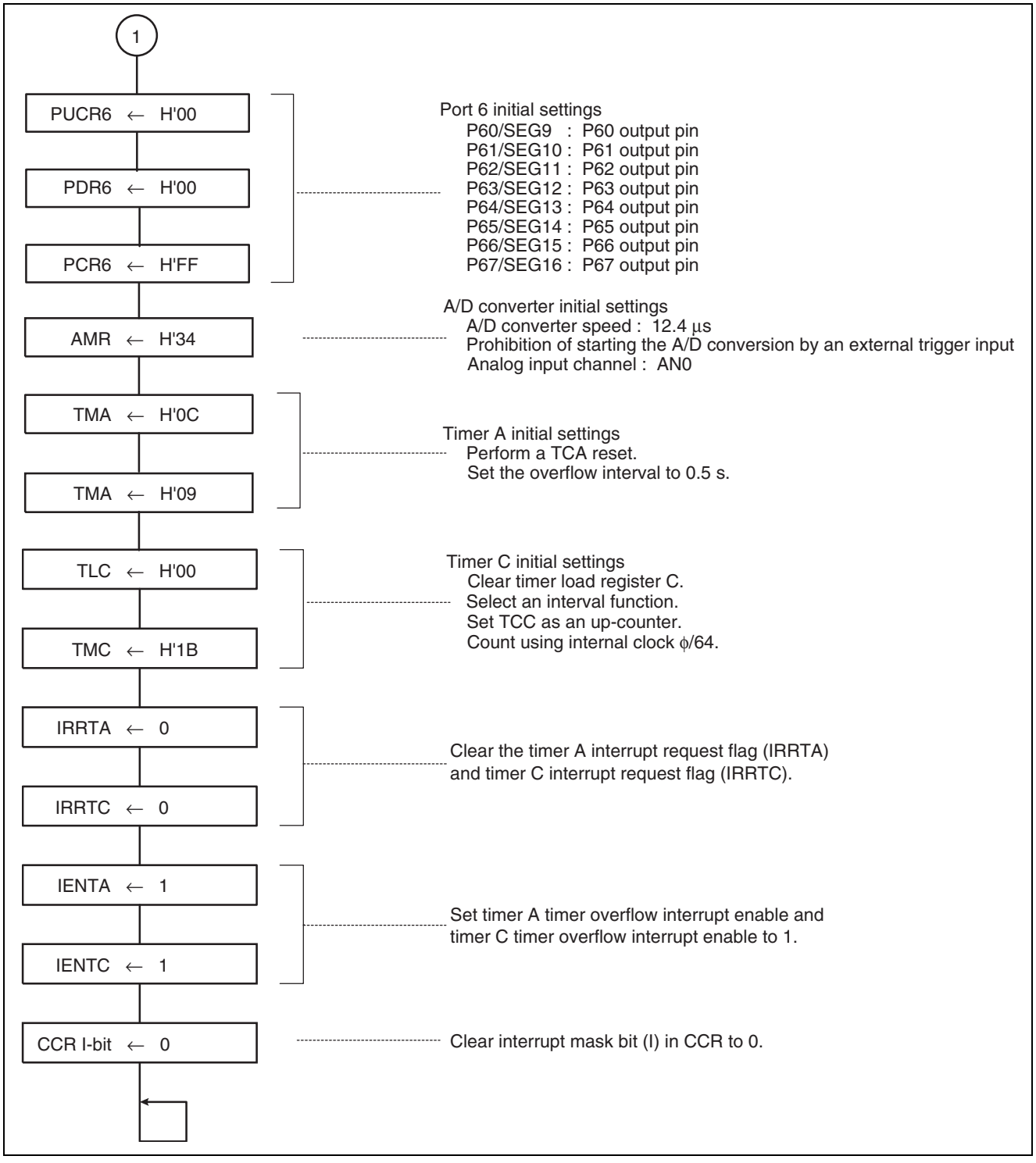
Element	Data	Description	Data size	Address
dsp_data[0]	H'3F	Port 5 output data for displaying "0" on a LED	1 byte	H'10FC
dsp_data[1]	H'06	Port 5 output data for displaying "1" on a LED	1 byte	H'10FD
dsp_data[2]	H'5B	Port 5 output data for displaying "2" on a LED	1 byte	H'10FE
dsp_data[3]	H'4F	Port 5 output data for displaying "3" on a LED	1 byte	H'10FF
dsp_data[4]	H'66	Port 5 output data for displaying "4" on a LED	1 byte	H'1100
dsp_data[5]	H'6D	Port 5 output data for displaying "5" on a LED	1 byte	H'1101
dsp_data[6]	H'7D	Port 5 output data for displaying "6" on a LED	1 byte	H'1102
dsp_data[7]	H'27	Port 5 output data for displaying "7" on a LED	1 byte	H'1103
dsp_data[8]	H'7F	Port 5 output data for displaying "8" on a LED	1 byte	H'1104
dsp_data[9]	H'6F	Port 5 output data for displaying "9" on a LED	1 byte	H'1105

5. Flowchart

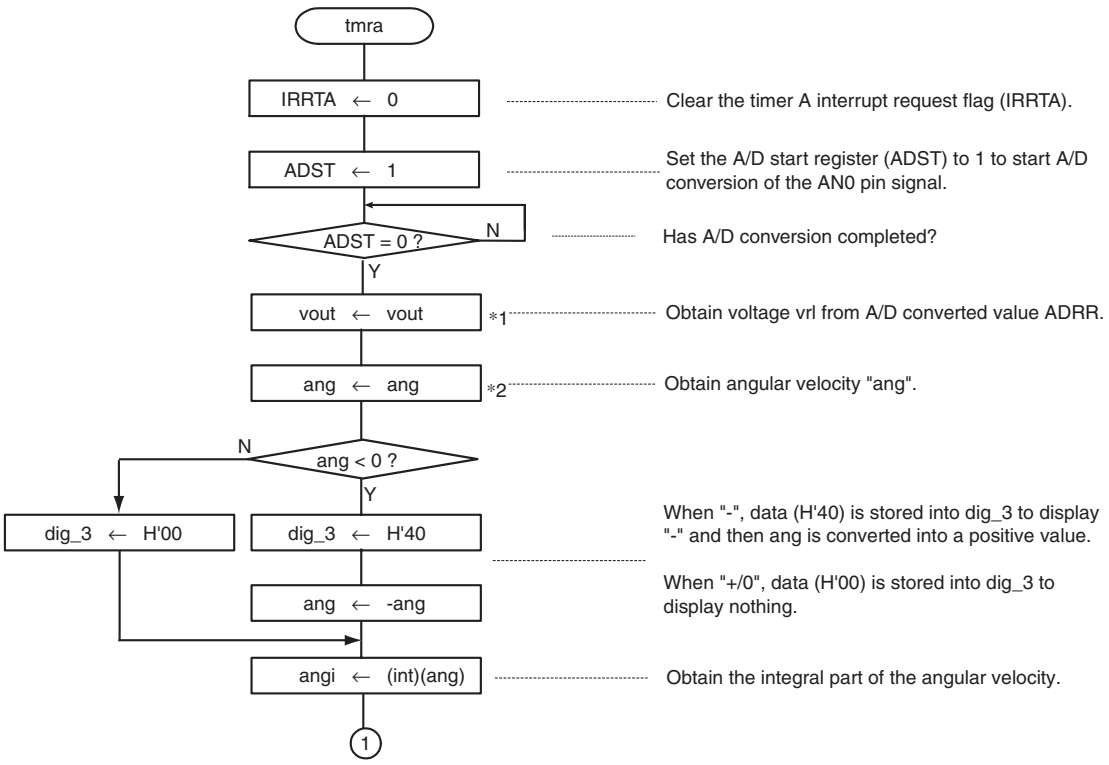
1. Main Routine (main)

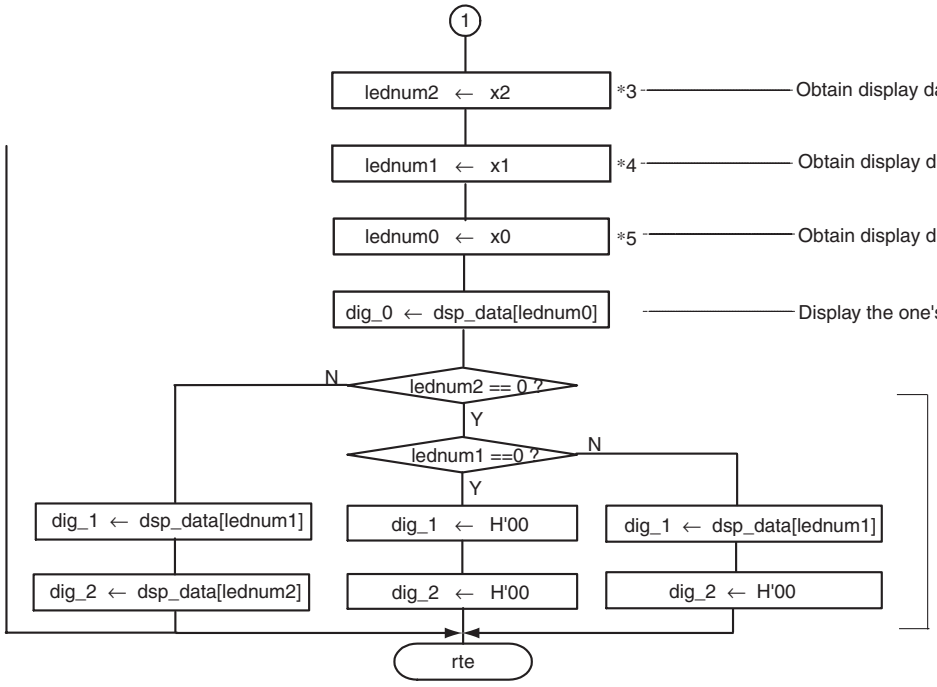


*1 In this sample task, INIT.SRC (assembly language) is used to set stack pointers.


1
PUCR6 ← H'00
PDR6 ← H'00
PCR6 ← H'FF
Port 6 initial settings
P60/SEG9 : P60 output pin
P61/SEG10 : P61 output pin
P62/SEG11 : P62 output pin
P63/SEG12 : P63 output pin
P64/SEG13 : P64 output pin
P65/SEG14 : P65 output pin
P66/SEG15 : P66 output pin
P67/SEG16 : P67 output pin
AMR ← H'34
A/D converter initial settings
A/D converter speed : 12.4 µs
Prohibition of starting the A/D conversion by an external trigger input
Analog input channel : AN0
TMA ← H'0C
TMA ← H'09
Timer A initial settings
Perform a TCA reset.
Set the overflow interval to 0.5 s.
TLC ← H'00
TMC ← H'1B
Timer C initial settings
Clear timer load register C.
Select an interval function.
Set TCC as an up-counter.
Count using internal clock ϕ/64.
IRRTA ← 0
IRRTC ← 0
Clear the timer A interrupt request flag (IRRTA)
and timer C interrupt request flag (IRRTC).
IENTA ← 1
IENTC ← 1
Set timer A timer overflow interrupt enable and
timer C timer overflow interrupt enable to 1.
CCR I-bit ← 0
Clear interrupt mask bit (I) in CCR to 0.

2. Timer A Interrupt Processing Routine (tmra)

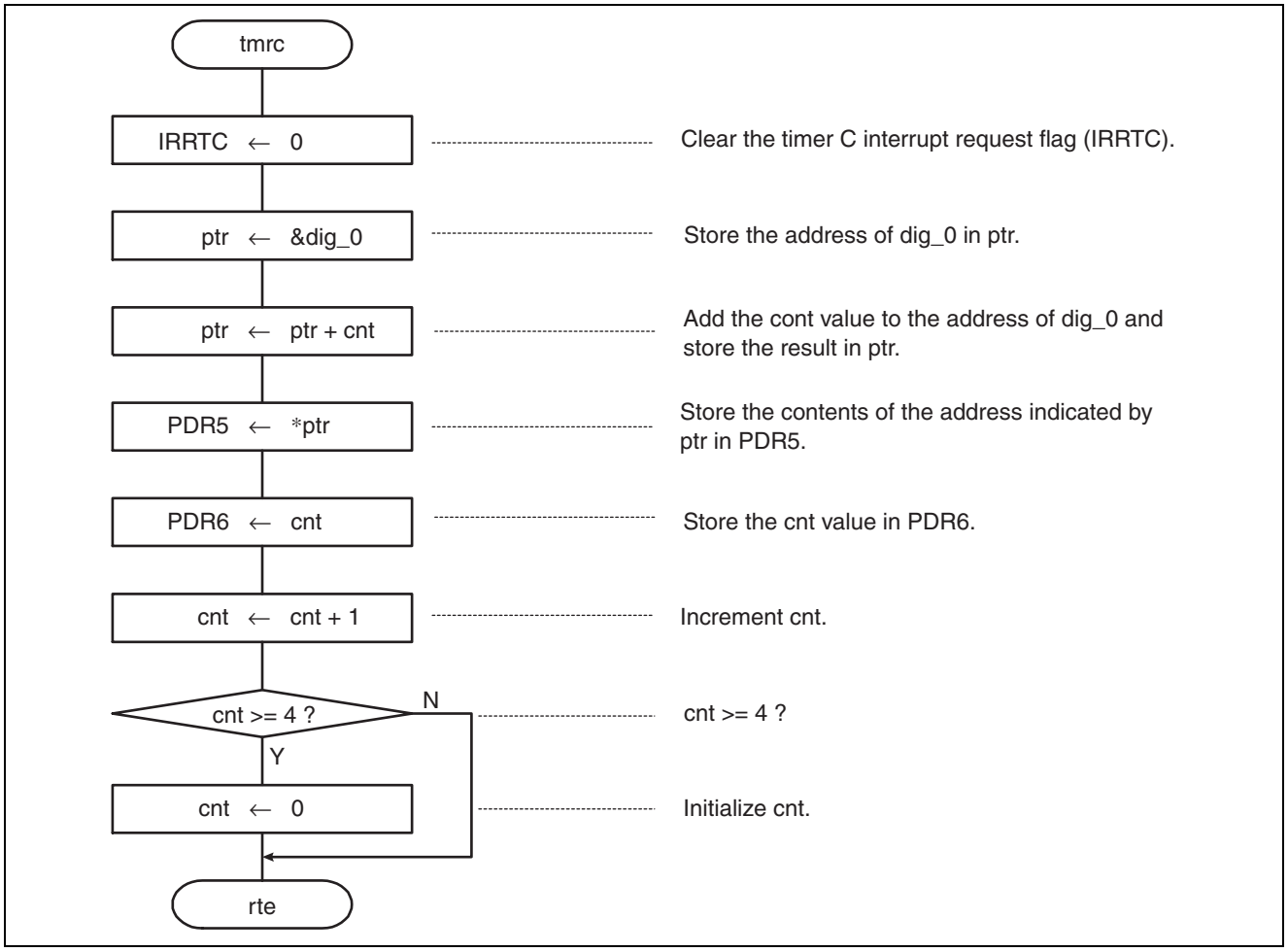




[Note]
*1 : vout= (ADRR >> 6 & 0x03ff) * 3.3 / 1023.0
*2 : ang = (vout - 1.35) * RATE / SCALE
*3 : x2 = (unsigned char)(angi / 100)
*4 : x1 = (unsigned char)(angi % 100 / 10)
*5 : x0 = (unsigned char)(angi % 10)
*6 : When lednum2 = 0
(1) When lednum1 = 0, the hundred's and ten's places are not displayed. Therefore, store data (H'00) into dig_2 and dig_1.
(2) When lednum1 > 0, the hundred's place is not displayed. Therefore, store data (H'00) into dig_2. For the ten's place, store the display data into dig_1 with reference to dsp_data.

When lednum2 > 0, for the hundred's and ten's places, store the display data into dig_2 and dig_1 with reference to dsp_data.

3. Timer C Interrupt Processing Routine (tmrc)



6. Program Listing

INIT.SRC (program listing)

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

```
/* Super Low Power Series -H8/38024- Application note */
/* Application example */
/* Example of connecting the GYROSTAR angular velocity sensor */

#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 PMR5    *(volatile unsigned char *)0xFFCC       /* Port mode register 5 */
#define PUCR5   *(volatile unsigned char *)0xFFE2       /* 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 PUCR6   *(volatile unsigned char *)0xFFE3       /* Port pull-up control register 6 */
#define PDR6    *(volatile unsigned char *)0xFFD9       /* Port data register 6 */
#define PCR6    *(volatile unsigned char *)0xFFE9       /* Port control register 6 */

#define TMA*(volatile unsigned char *)0xFFB0            /* Timer mode register A */
#define CKSTPR1     *(volatile unsigned char *)0xFFFA   /* Clock stop register 1 */

#define TMC*(volatile unsigned char *)0xFFB4            /* Timer mode register C */
#define TLC*(volatile unsigned char *)0xFFB5            /* Timer Load register C */
#define ADRR    *(volatile unsigned int  *)0xFFC4       /* A/D result register(word access) */
#define ADRRH   *(volatile unsigned int  *)0xFFC4       /* A/D result register(byte access) */
#define ADRRL   *(volatile unsigned int  *)0xFFC5       /* A/D result register(byte access) */
#define AMR*(volatile unsigned char *)0xFFC6            /* A/D mode register */
#define ADSR    *(volatile unsigned char *)0xFFC7       /* A/D start register */
#define ADSR_BIT   (*(struct BIT *)0xFFC7)              /* A/D start */
#define ADST    ADSR_BIT.b7                             /* A/D start */
```

```
#define IRR1   *(volatile unsigned char *)0xFFF6          /* Interrupt request register 1 */
#define IRR1_BIT  (*(struct BIT *)0xFFF6)                 /* Interrupt request register 1 */
#define IRRTA  IRR1_BIT.b7                                /* Timer A interrupt request flag */
#define IENR1  *(volatile unsigned char *)0xFFF3          /* Interrupt enable register 1 */
#define IENR1_BIT (*(struct BIT *)0xFFF3)                 /* Interrupt enable register 1 */
#define IENTA   IENR1_BIT.b7                              /* Timer A interrupt enable */

#define IRR2   *(volatile unsigned char *)0xFFF7          /* Interrupt request register 2 */
#define IRR2_BIT  (*(struct BIT *)0xFFF7)                 /* Interrupt request register 2 */

#define IRRTC   IRR2_BIT.b1                               /* Timer C interrupt request flag */
#define IENR2  *(volatile unsigned char *)0xFFF4          /* Interrupt enable register 2 */
#define IENR2_BIT (*(struct BIT *)0xFFF4)                 /* Interrupt enable register 2 */
#define IENTC  IENR2_BIT.b1                               /* Timer C interrupt enable */

#pragma interrupt (tmra)
#pragma interrupt (tmrc)

/* Function definition */
extern void INIT(void);                                   /* Stack pointer set */
void main(void);                                          /* main routine */
void tmra(void);                                          /* Timer A interrupt routine */
void tmrc(void);                                          /* Timer C interrupt routine */

/* Data table */
const unsigned char dsp_data[10] =
{
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" */
};

/* 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 */
float vout;                                               /* OPamp analog voltage output */
float ang;                                                /* angular velocity result */
int   angi;                                               /* intger angular velocity result */
float SCALE;                                              /* Scale factor const */
float RATE;                                               /* Amplify rate const */
unsigned char lednum0;                                    /* Dig-0 LED display data */
unsigned char lednum1;                                    /* Dig-1 LED display data */
unsigned char lednum2;                                    /* Dig-2 LED display data */
unsigned char *ptr;                                       /* Pointer set */
```

```
/* 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) = {
    tmra                                                    /* H'0016 Timer A interrupt vector */
};
#pragma section V3                                          /* Vector section set */
void (*const VEC_TBL3[])(void) = {
    tmrc                                                    /* H'001a Timer C 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 */
    SCALE = 0.00067;                                        /* Scale factor  */
    RATE  = 10.0/101.0;                                     /* Amplify rate */
    cnt   = 0x00;                                           /* Used RAM area initialize */

    PMR5  = 0x00;                                           /* Port 5 initialize */
    PUCR5 = 0x00;
    PDR5  = 0x00;
    PCR5  = 0xff;

    PUCR6 = 0x00;                                           /* Port 6 initialize */
    PDR6  = 0x00;
    PCR6  = 0xFF;

    AMR   = 0x34;                                           /* A/D converter initialize (AN0) */

    TMA   = 0x0c;                                           /* Clear Timer Counter A to 0 */
    TMA   = 0x09;                                           /* Timer A initialize */
    TLC   = 0x00;                                           /* Clear Timer Load register C */
    TMC   = 0x1b;                                           /* Timer C initialize */

    IRRTA = 0;                                              /* Clear IRRTA to 0 */
    IRRTC = 0;                                              /* Clear IRRTC to 0 */
    IENTA = 1;                                              /* Timer A interrupt enable */
    IENTC = 1;                                              /* Timer C interrupt enable */

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

    while (1);
```

```
}
/************************************************************/
/* Timer A Interrupt                                        */
/************************************************************/
void tmra(void)
{
    IRRTA = 0;                                          /* Clear IRRTA to 0 */
ADST  = 1;                                              /* A/D converter start */
    While (ADST == 1);                                  /* A/D converter end ? */
    vout  = (ADRR >> 6 & 0x03ff) * 3.3 / 1023.0;        /* Output voltage */
    ang  = (vout - 1.35) * RATE / SCALE;                /* Angular velocity sensor result */
    if (ang < 0) {
        dig_3 = 0x40;                                   /* Dig-3 LED display data set */
        ang = -ang;                                     /* ang = -ang */
    }else{
        dig_3 = 0x00;                                   /* Dig-3 LED display data set */
    }
    angi = (int)(ang);                                  /* int angular velocity result */
    lednum2 = (unsigned char)(angi / 100);              /* Compute Dig-2 LED display data */
    lednum1 = (unsigned char)(angi % 100 / 10);         /* Compute Dig-1 LED display data */
    lednum0 = (unsigned char)(angi % 10);               /* Compute Dig-0 LED display data */
    dig_0 = dsp_data[lednum0];                          /* Dig-0 LED display data set */
    if (lednum2 == 0){
        if (lednum1 == 0){
            dig_1 = 0x00;                               /* Dig-1 LED display data set */
        dig_2 = 0x00;                                   /* Dig-2 LED display data set */
         }else{
            dig_1 = dsp_data[lednum1];                  /* Dig-1 LED display data set */
            dig_2 = 0x00;                               /* Dig-2 LED display data set */
         }
    }else{
        dig_1 = dsp_data[lednum1];                      /* Dig-1 LED display data set */
         dig_2 = dsp_data[lednum2];                     /* Dig-2 LED display data set */
    }
}
/************************************************************/
/* Timer C Interrupt                                        */
/************************************************************/
void tmrc(void)
{
    IRRTC = 0;                                          /* Clear IRRTC to 0 */

    ptr = &dig_0;                                       /* LED data store address set */
    ptr += cnt;                                         /* LED display data read */
    PDR5 = *ptr;                                        /* LED display data output */
    PDR6 = cnt;                                         /* LED enable data output */

    cnt++;                                              /* "cnt"increment */
    if (cnt >= 4){                                      /* 4 times end ? */
        cnt = 0;                                        /* "cnt" initialize */
   }
}
```

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Sept.19.03	—	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.