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.

H8/300H Tiny 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. You have the angular velocity in decimal number (deg/sec) on an array of 7-segment LED.

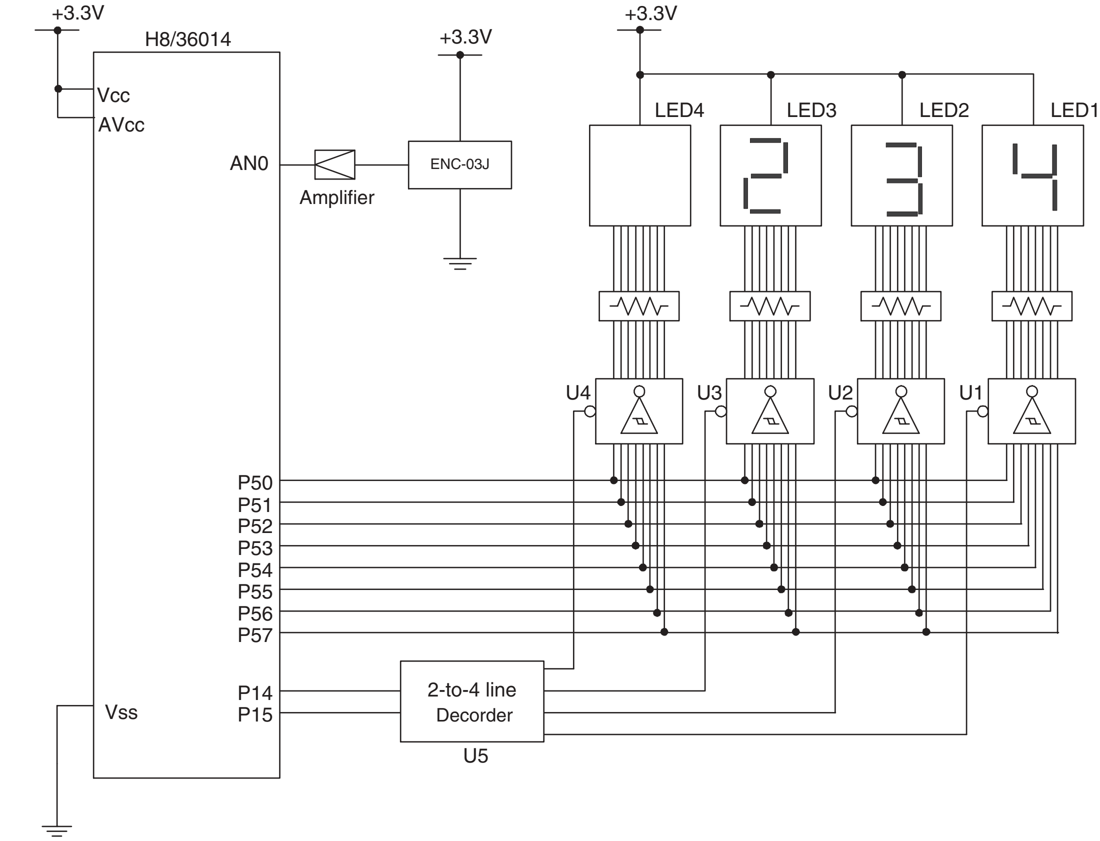
Target Device

H8/300H Tiny Series H8/36014 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 approx. 78ms.



Legend:

ENC-03J	: Angular velocity sensor (ENC-03J, manufactured by Murata Manufacturing Co., Ltd.)
LED1 to LED4	: Common-anode 7-segment LEDs
U1 to U4	: Tri-state output inverter drivers (HD74LS240)
U5	: 2-to-4 line decoder (HD74HC139)

Figure 1 Hardware Configuration

4. In this sample task, the operating voltage (Vcc) and analog power supply voltage (AVcc) of the H8/36014 are 3.3 V and the OSC clock frequency is 10 MHz.
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) ±20̣ (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 ±5.0̣
Sensitivity fluctuation with temperature	-20 to +10̣ (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.

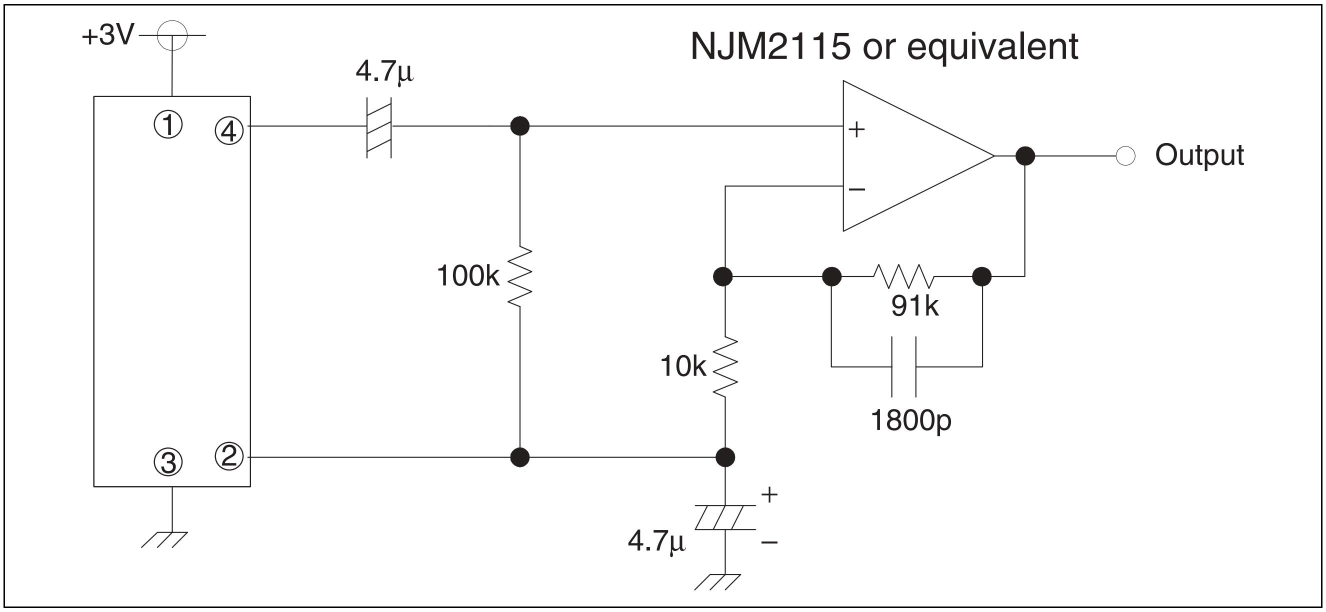
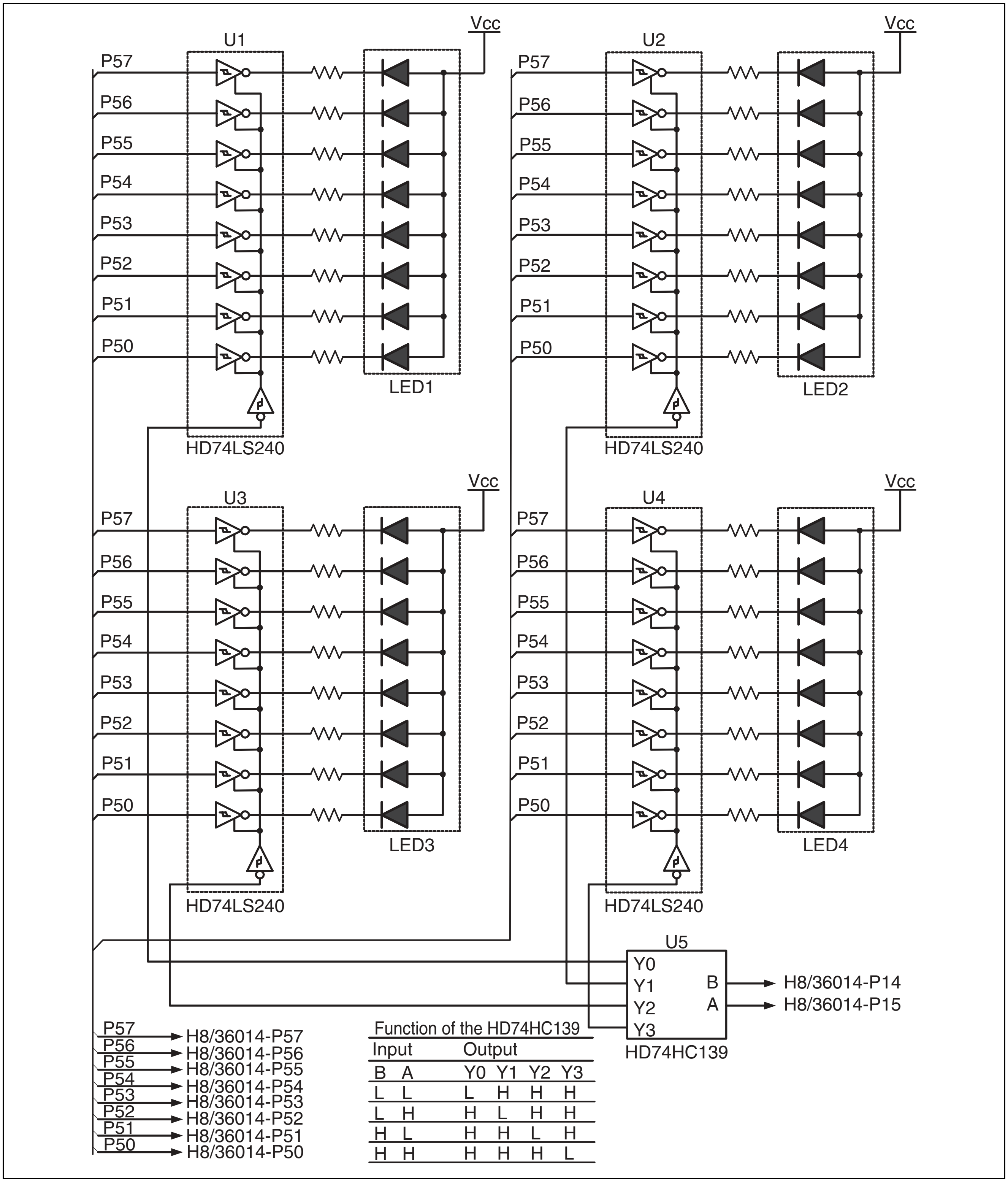


Figure 2 Sensor Standard Amplification Circuit (Sample Circuit)

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.

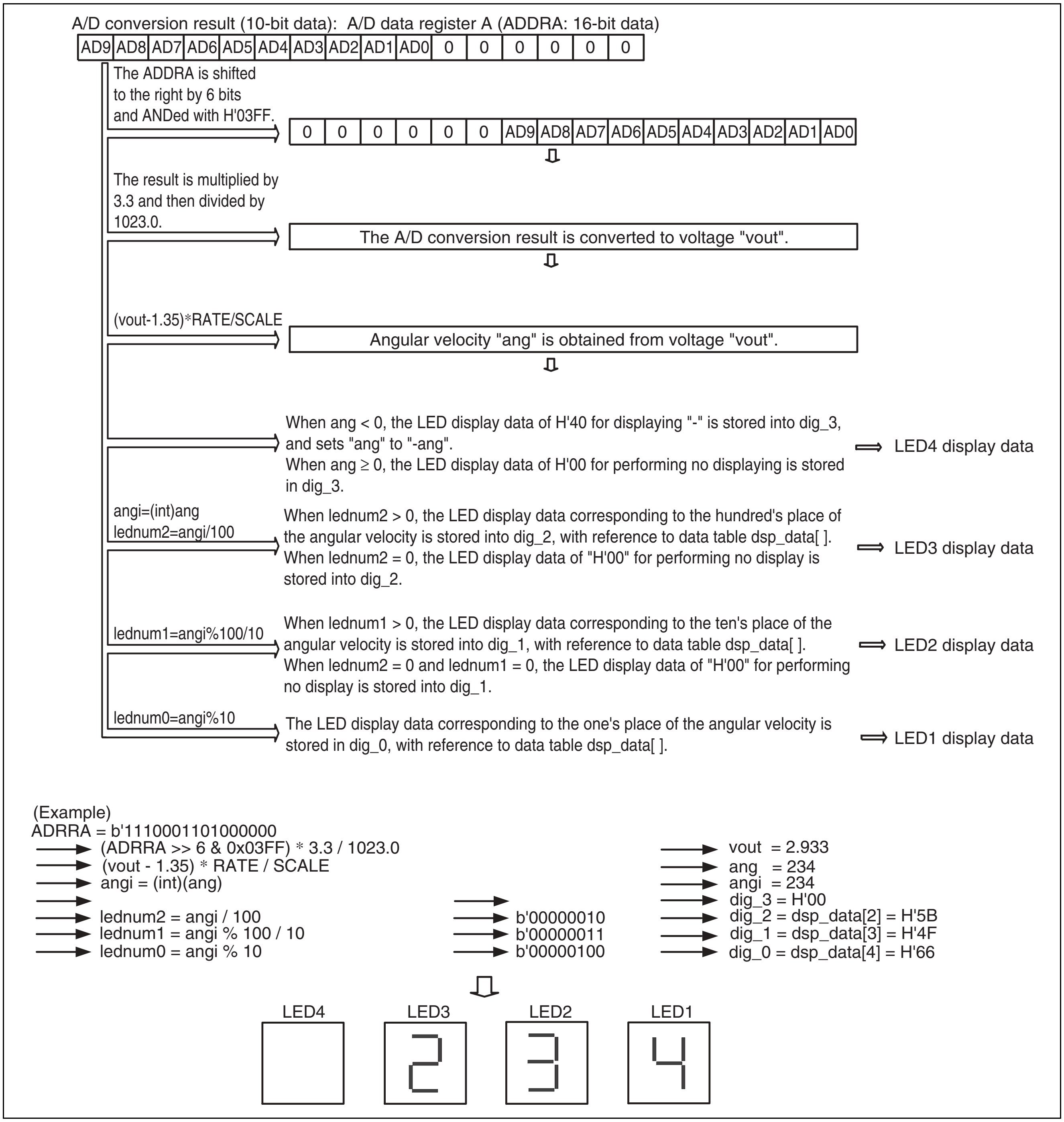


Function of the HD74HC139

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 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.


A/D conversion result (10-bit data): A/D data register A (ADDRA: 16-bit data)
AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0 0 0 0 0 0 0
0 0 0 0 0 0 AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0
(vout-1.35)*RATE/SCALE
angi=(int)ang
lednum2=angi/100
lednum1=angi%100/10
lednum0=angi%10
LED4 display data
LED3 display data
LED2 display data
LED1 display data
LED4
LED3
LED2
LED1

The ADDRA is shifted to the right by 6 bits and ANDed with H'03FF.

The result is multiplied by 3.3 and then divided by 1023.0.

The A/D conversion result is converted to voltage "vout".

Angular velocity "ang" is obtained from voltage "vout".

When ang < 0, the LED display data of H'40 for displaying "-" is stored into dig_3, and sets "ang" to "-ang".
When ang ≥ 0, the LED display data of H'00 for performing no displaying is stored in dig_3.

When lednum2 > 0, the LED display data corresponding to the hundred's place of the angular velocity is stored into dig_2, with reference to data table dsp_data[ ].
When lednum2 = 0, the LED display data of "H'00" for performing no display is stored into dig_2.

When lednum1 > 0, the LED display data corresponding to the ten's place of the angular velocity is stored into dig_1, with reference to data table dsp_data[ ].
When lednum2 = 0 and lednum1 = 0, the LED display data of "H'00" for performing no display is stored into dig_1.

The LED display data corresponding to the one's place of the angular velocity is stored in dig_0, with reference to data table dsp_data[ ].

(Example)
ADRRA = b'1110001101000000

```
──► (ADRRA >> 6 & 0x03FF) * 3.3 / 1023.0          ──► vout  = 2.933
──► (vout - 1.35) * RATE / SCALE                  ──► ang   = 234
──► angi = (int)(ang)                             ──► angi  = 234
──►                                  ──►          ──► dig_3 = H'00
──► lednum2 = angi / 100             ──► b'00000010  ──► dig_2 = dsp_data[2] = H'5B
──► lednum1 = angi % 100 / 10        ──► b'00000011  ──► dig_1 = dsp_data[3] = H'4F
──► lednum0 = angi % 10              ──► b'00000100  ──► dig_0 = dsp_data[4] = H'66
```

Figure 4 Display of Angular Velocity on the LEDs

2. Description of Functions

1. Figure 5 is a block diagram of the H8/36014 functions used in this task. Table 2 lists the function allocations.

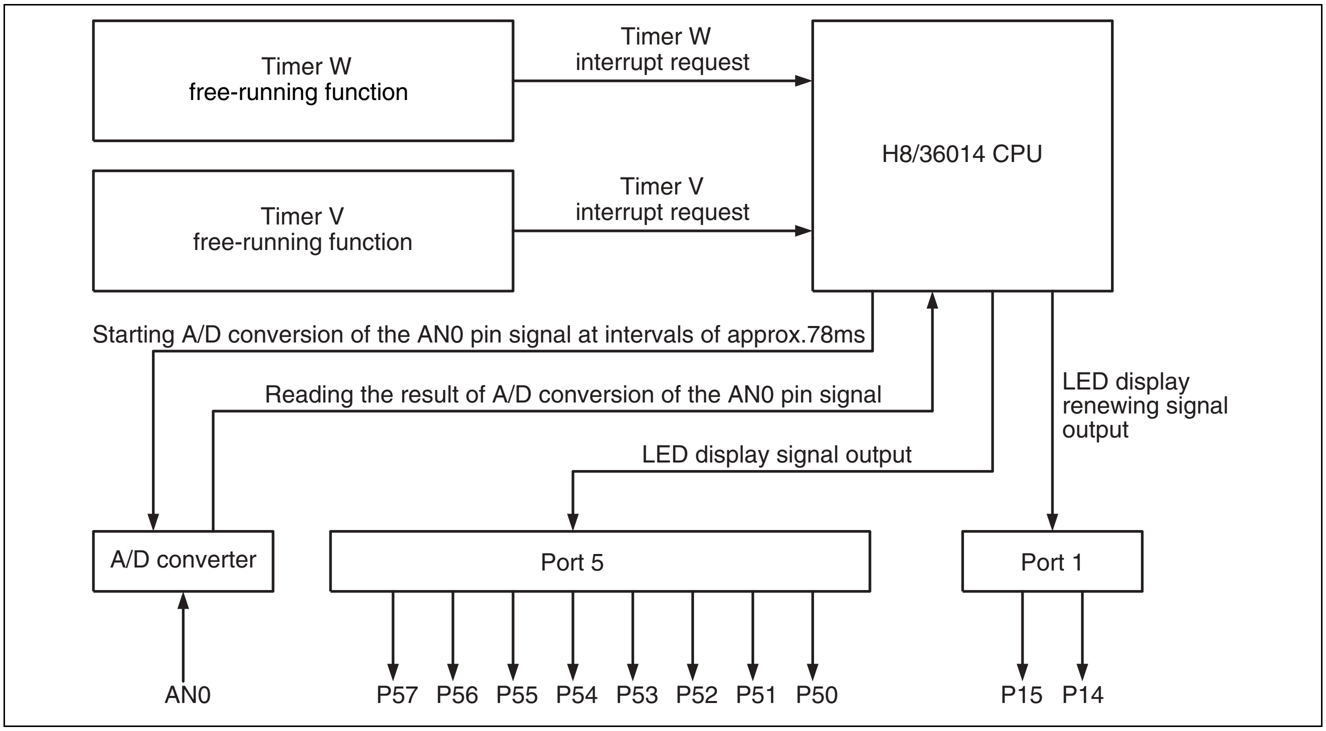


Figure 5 Block Diagram of Functions Used in This Sample Task

Table 2 Function Assignment

Function	Function assignment
Timer W	The free-running function of timer W is used to perform A/D conversion of analog input pin 0 (AN0). The A/D conversion period is measured at intervals of approx. 52.4ms of the timer W overflow period.
Timer V	The free-running function of timer V 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 V 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 1	The P14 and P15 output pins of port 1 are used to renew the display on the four 7-segment LEDs. The P14 and P15 output pins are connected to the I/O pins of a 2-to-4-line decoder.
Port 5	The P50 to P57 output pins of port 5 are used to display data on the four 7-segment LEDs. The 10-bit data, obtained as a result of the A/D conversion of the AN0 pin, is converted to the angular velocity display data with decimal places (deg/sec) and then output to the LEDs.

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.

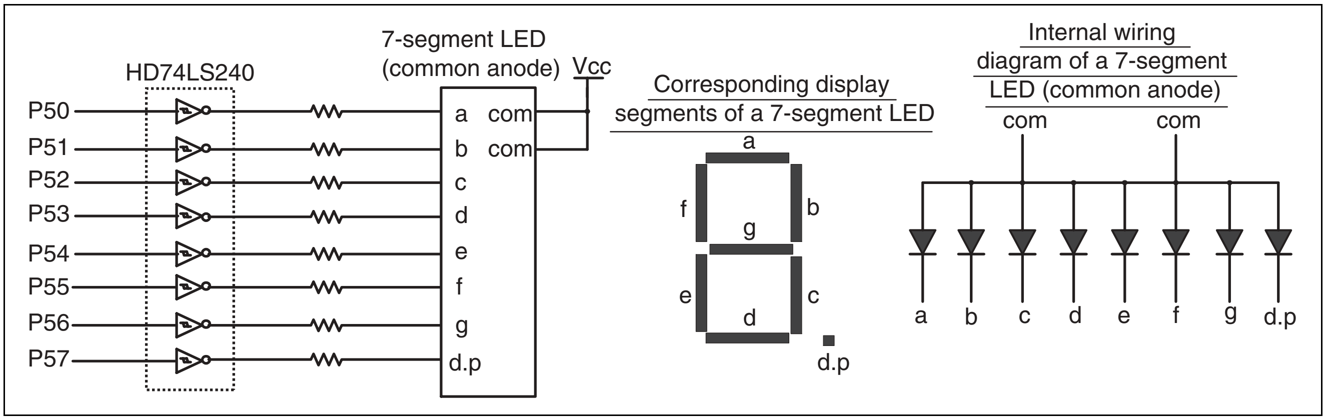
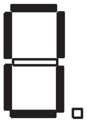
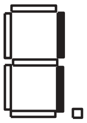
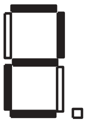
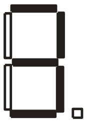
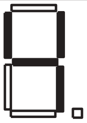
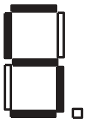
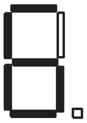
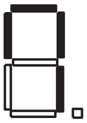
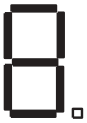
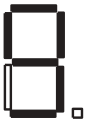
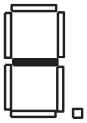

HD74LS240
7-segment LED
(common anode)
Vcc
P50
P51
P52
P53
P54
P55
P56
P57
a
b
c
d
e
f
g
d.p
com
com
Corresponding display segments of a 7-segment LED
a
b
c
d
e
f
g
d.p
Internal wiring diagram of a 7-segment LED (common anode)
com
com
a
b
c
d
e
f
g
d.p

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							
	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	0	0	0	0	0	0

3. Description of Operation

1. Figure 7 shows the description of A/D conversion of the AN0 pin signal when timer W is used. In this sample task, the completion of A/D conversion and the A/D conversion period (78ms) are determined in the tmrw routine by the timer W overflow flag without using an A/D conversion interrupt.

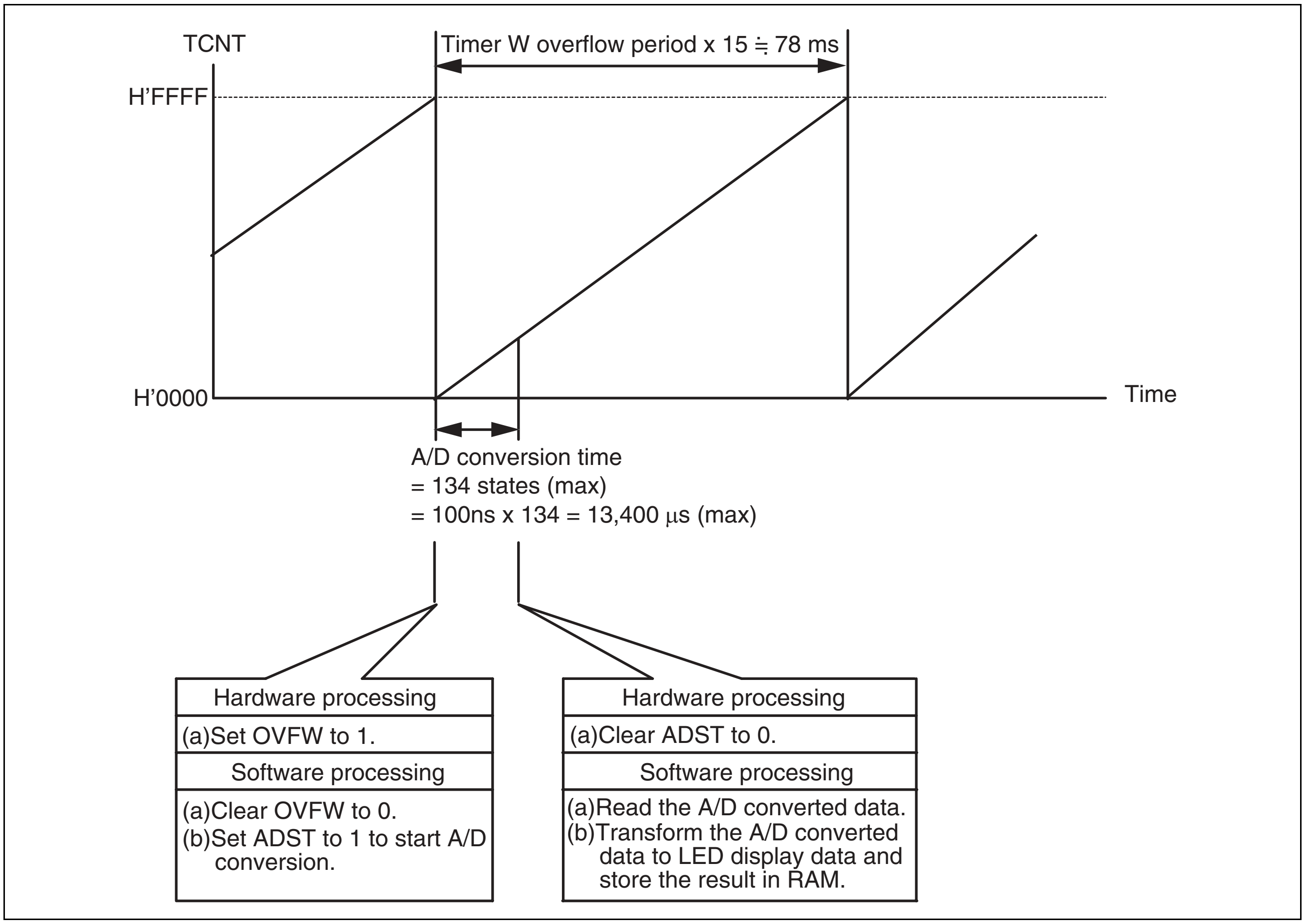
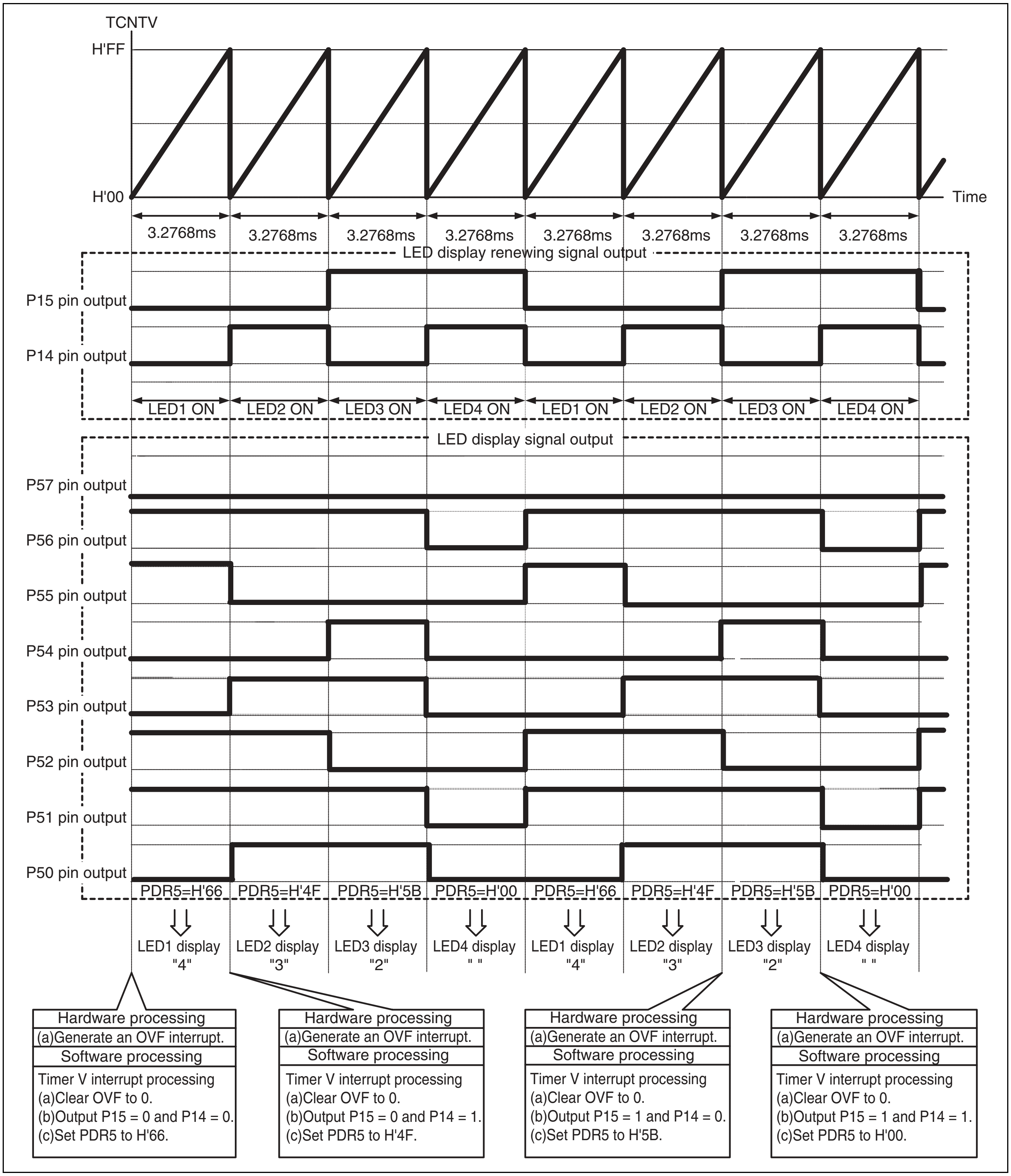

TCNT
Timer W overflow period x 15 ≒ 78 ms
H'FFFF
H'0000
Time
A/D conversion time
= 134 states (max)
= 100ns x 134 = 13,400 µs (max)
Hardware processing
(a)Set OVFW to 1.
Software processing
(a)Clear OVFW to 0.
(b)Set ADST to 1 to start A/D conversion.
Hardware processing
(a)Clear ADST to 0.
Software processing
(a)Read the A/D converted data.
(b)Transform the A/D converted data to LED display data and store the result in RAM.

Figure 7 A/D Conversion of the AN0 Pin Signal when Timer W 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 V overflow interval, resulting in dynamic display with the 7-segment LED.



Hardware processing
(a)Generate an OVF interrupt.
Software processing
Timer V interrupt processing
(a)Clear OVF to 0.
(b)Output P15 = 0 and P14 = 0.
(c)Set PDR5 to H'66.

Hardware processing
(a)Generate an OVF interrupt.
Software processing
Timer V interrupt processing
(a)Clear OVF to 0.
(b)Output P15 = 0 and P14 = 1.
(c)Set PDR5 to H'4F.

Hardware processing
(a)Generate an OVF interrupt.
Software processing
Timer V interrupt processing
(a)Clear OVF to 0.
(b)Output P15 = 1 and P14 = 0.
(c)Set PDR5 to H'5B.

Hardware processing
(a)Generate an OVF interrupt.
Software processing
Timer V interrupt processing
(a)Clear OVF to 0.
(b)Output P15 = 1 and P14 = 1.
(c)Set PDR5 to H'00.

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 W interrupt processing routine	tmrw	Clears the interrupt flags, transforms the A/D-converted data to LED display data, and then stores the result in RAM.
Timer V interrupt processing routine	tmrv	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
TCRV0		Timer control register V0: Selects an input clock of TCNTV, specifies the clear conditions and controls each interrupt request.	H'FFA0	H'03 (At initial setting)
	CMIEB	Compare match interrupt enable B: Prohibits an interrupt request by CMFB of TCSRV when the bit is 0.	Bit 7	0
	CMIEA	Compare match interrupt enable A: Prohibits an interrupt request by CMFA of TCSRV when the bit is 0.	Bit 6	0
	OVIE	Timer overflow interrupt enable Prohibits an interrupt request by OVF of TCSRV when the bit is 0. Enables an interrupt request by OVF of TCSRV when the bit is 1.	Bit 5	0/1
	CCLR1	Counter clear 1 to 0: Specifies the clear conditions of TCNTV. When CCLR1=0 and CCLR0=0 are set, clearing TCNTV is prohibited.	Bit 4	0
	CCLR0		Bit 3	0
	CKS2	Clock select 2 to 0: Selects the clock and count conditions to input to TCNTV in combination with TCRV1 and ICKS0. When CKS2 = 0, CKS1 = 1, CKS0 = 1, and ICKS0 = 1 are set, TCNTV performs a count at the falling edge whose internal clock is ϕ/128.	Bit 2	0
	CKS1		Bit 1	1
	CKS0		Bit 0	1

Register name		Description	Address	Setting
TCSRV		Timer control/status register V: Displays a status flag and controls the output by a compare match.	H'FFA1	H'10
	CMFB	Compare match flag B: Sets to 1 when the values of TCNTV and TCORB match.	Bit 7	0
	CMFA	Compare match flag A: Sets to 1 when the values of TCNTV and TCORA match.	Bit 6	0
	OVF	Timer overflow flag: Sets to 1 when the value of TCNTV overflows. Clears to 0 when a zero is written to OVF after reading OVF in the OVF=1 status.	Bit 5	0
	OS3	Output select 3 to 2: Sets the output levels of the TMOV pins with compare match B. When OS3=0 and OS2=0 are sets, nothing changes.	Bit 3	0
	OS2		Bit 2	0
	OS1	Output select 1 to 0: Sets the output levels of the TMOV pin with compare match A. When OS1 = 0 and OS0 = 0 are set, nothing changes.	Bit 1	0
	OS0		Bit 0	0
TCRV1		Timer control register V1: Selects the edge of the TRGV pin, enables the input, and selects the input clock of TCNTV.	H'FFA5	H'E3
	TVEG1	TRGV input edge select 1 to 0: Selects the input edge of the TRGV pins When TREG1 = 0 and TREG0 = 0 are set, the trigger input from the TRGV pins is prohibited.	Bit 4	0
	TVEG0		Bit 3	0
	TRGE	TRGV input enable: Enables or prohibits the TCNTV count-up by the edge input selected in TVEG1 and TVEG0. When TREG = 0 is set, the startup of the TCNTV count-up by the pin input of TRGV and the stop of TCNTV count-up are prohibited if TCNTV is cleared by a compare match.	Bit 2	1
	ICKS0	Internal clock select 0: Selects the clock and count conditions to input to TCNTV depending on the combination of CKS2 to CKS0 of TCRV0. When CKS2 = 0, CKS1 = 1, CKS0 = 1, and ICKS0 = 1 are set, TCNTV performs a count at the falling edge whose internal clock is ϕ/128.	Bit 0	1

Register name		Description	Address	Setting
TMRW		Timer mode register W: Selects the general register function and the output mode.	H'FF80	H'80
	CTS	Count startup When CTS = 1, TCNT indicates the startup of the counter. When CTS = 0, TCNT indicates the stop of the counter.	Bit 7	1
TCRW		Timer control register W: Selects the counter clock. Sets the clear conditions of a counter and the output level of a counter.	H'FF81	H'30
	CKS2	Clock select: When CKS2 = 0, CKS1 = 1, and CKS0 = 1, this function sets the TCNT input clock to the 8-frequency divided clock of the system clock.	Bit 6	0
	CKS1		Bit 5	1
	CKS0		Bit 4	1
TIEWR		Timer interrupt enable register W: Controls the interrupt request to timer W.	H'FF82	H'00 (At initial setting)
	OVIE	Timer overflow interrupt enable: Prohibits an interrupt request by OVF when OVIE = 0. Enables an interrupt request by OVF when OVIE = 1	Bit 7	0/1
TSRW		Indicates the interrupt request status.	H'FF83	H'00
	OVF	Timer overflow: Indicates that TCNT does not overflow when OVF = 0. Indicates that TCNT overflows when OVF = 1.	Bit 7	0
TCNT		Timer counter: This is a 16-bit up-counter used as the input of the 8-frequency divided clocks of the system clock.	H'FF86	H'00
ADCSR		A/D control and status registers: This consists of the control bit of an A/D converter and the conversion status bit.	H'FFB8	H'00 (At initial setting)
	ADF	A/D end flag: Sets to 1 when the A/D conversion terminates in the single mode. Sets to 0 when a zero is written after reading the ADF = 1 status.	Bit 7	0
	ADIE	A/D interrupt enable Prohibits the request for the A/D conversion end interrupt when the bit is 0.	Bit 6	0
	ADST	A/D start Starts the A/D conversion by setting to 1. When the A/D conversion terminates in the single mode, the bit is automatically cleared to 0.	Bit 5	0/1
	SCAN	Scan mode The A/D conversion is in the single mode when the bit is 0.	Bit 4	0

Register name		Description	Address	Setting
	CKS	Clock select The A/D conversion time is 134 states at maximum when the bit is 0.	Bit 3	0
	CH2	Channel select Selects an analog input channel. When CH2 = 0, CH1 = 0 and CH0 = 0 are set (SCAN = 0), AN0 is selected.	Bit 2	0
	CH1		Bit 1	0
	CH0		Bit 0	0
ADCR		A/D control register: Sets the A/D conversion startup by an external trigger to an enable.	H'FFB9	H'7E
	TRGE	Trigger enable: Prohibits the A/D conversion startup by the edge input of the external trigger pin (ADTRG) when the bit is 0.	Bit 7	0
ADDRA		A/D data register A: Stores the A/D conversion results for AN0.	H'FFB0	-
PMR1		Port mode register 1: Sets the pin function for ports 1, 2, and 7.	H'FFE0	H'00
	IRQ3	Switching the pin function of P17/_IRQ3/TRGV: Provides the function of the general I/O port of P17 when the bit is 0.	Bit 7	0
	IRQ0	Switching the pin function of P14/_IRQ0: Provides the function of the general I/O port of P14 when the bit is 0.	Bit 4	0
	TXD2	Switching the pin function of P72/TXD_2: Provides the function of the general I/O port of P72 when the bit is 0.	Bit 3	0
	TXD	Switching the pin function of P22/TXD	Bit 1	0
PCR1		Port control register: Selects the I/O of the pin used as a general I/O port of port 1 for each bit. When PCR1=H'38, P17 and P16, and P12 to P10 function as a general input pin. P14 and P15 function as a general output pin.	H'FFE4	H'38
PDR1		Port data register 1: This is a general I/O port data register of port 1.	H'FFD4	H'08
PUCR1		Port pull-up control register 1: Controls the pull-up MOS of each pin of port 1 set to the input port for the bit. When PUCR1 = H'08, the pull-up MOS for P17 to P14 and P12 to P10 are set to off.	H'FFD0	H'08
PMR5		Port mode register 5: Sets the pin function of port 5.	H'FFE1	H'00
	POF7	Switching the function of P57: Provides the general I/O port function of P57 when the bit is 0.	Bit 7	0
	POF6	Switching the function of P56: Provides the general I/O port function of P56 when the bit is 0.	Bit 6	0

Register name		Description	Address	Setting
	WKP5	Switching the pin function of P55/_WKP5/_ADTRG: Provides the general I/O port function of P55 when the bit is 0.	Bit 5	0
	WKP4	Switching the pin function of P54/_WKP4: Provides the general I/O port function of P54 when the bit is 0.	Bit 4	0
	WKP3	Switching the pin function of P53/_WKP3: Provides the general I/O port function of P53 when the bit is 0.	Bit 3	0
	WKP2	Switching the pin function of P52/_WKP2: Provides the general I/O port function of P52 when the bit is 0.	Bit 2	0
	WKP1	Switching the pin function of P51/_WKP1: Provides the general I/O port function of P51 when the bit is 0.	Bit 1	0
	WKP0	Switching the pin function of P50/_WKP0: Provides the general I/O port function of P50 when the bit is 0.	Bit 0	0
PUCR5		Port pull-up control register 5: Controls the pull-up MOS of port 5 set to the input port for the bit. When PUCR5 = H'00, the pull-up MOS of P57 to P50 is set to off.	H'FFD1	H'00
PDR5		Port data register 5: This is a general I/O port data register of port 5.	H'FFD8	H'00
PCR5		Port control register 5: Selects the I/O of the pins used as a general I/O port of port 5. When PCR5 = H’FF, P57 to P50 function as a general I/O pin.	H'FFE8	H'FF

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	tmrw
ang	Angular velocity	H'FB84	tmrw
angi	10-times angular velocity	H'FB88	tmrw
SCALE	Scale factor	H'FB8A	main, tmrw
RATE	Amplification rate	H'FB8E	main, tmrw
*ptr	Location where the address of dig_0 is stored	H'FB92	tmrv
dig_0	Stores LED1 display data (1 byte)	H'FB94	main, tmrw, tmrv
dig_1	Stores LED2 display data (1 byte)	H'FB95	main, tmrw
dig_2	Stores LED3 display data (1 byte)	H'FB96	main, tmrw
dig_3	Stores LED4 display data (1 byte)	H'FB97	main, tmrw
cnt	8-bit counter for switching display between LED1-LED4 (1 byte)	H'FB98	main, tmrv
counter_sub	8-bit counter to adjust an A/D acquisition interval (1 byte)	H'FB99	main, tmrw
lednum0	LED1 display data	H'FB9A	tmrw
lednum1	LED2 display data	H'FB9B	tmrw
lednum2	LED3 display data	H'FB9C	tmrw

5. Description of data tables
   In this task sample, the display data and the display switching signals for the 7-segment LEDs are stored in ROM as a data table of one-dimensional array. Table 7 shows the 7-segment LED display data table (dsp_data[]) and Table 8 shows the data table of the 7-segment display switching signals (led_data[]).

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

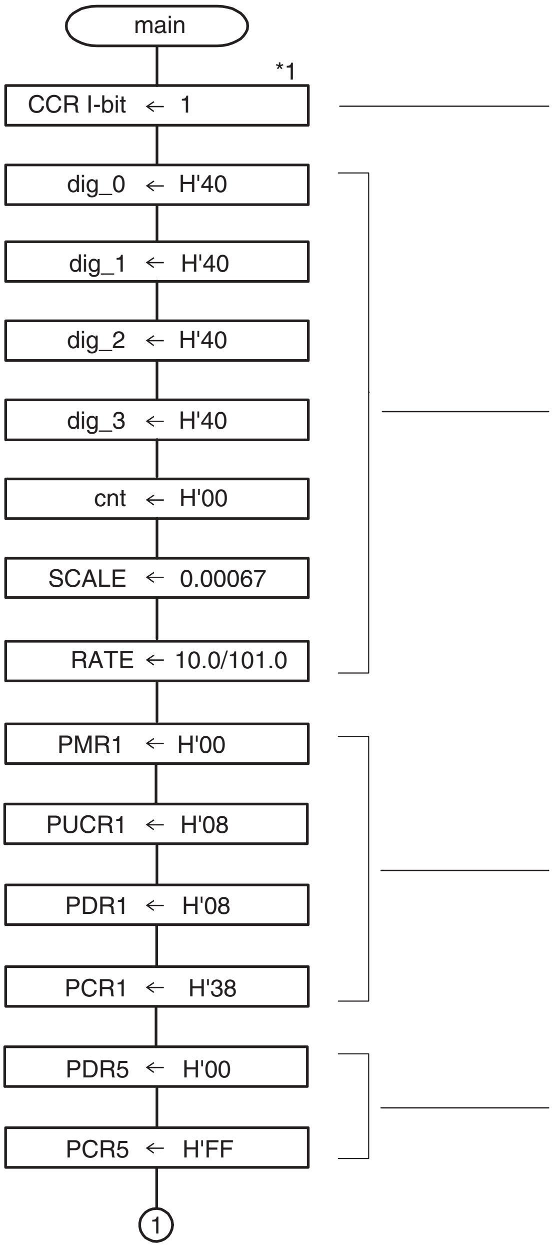
Element	Data	Description	Data size	Address
dsp_data[0]	H'3F	Port 5 output data for displaying ”0” on a LED	1 byte	H'E18
dsp_data[1]	H'06	Port 5 output data for displaying ”1” on a LED	1 byte	H'E19
dsp_data[2]	H'5B	Port 5 output data for displaying ”2” on a LED	1 byte	H'E1A
dsp_data[3]	H'4F	Port 5 output data for displaying ”3” on a LED	1 byte	H'E1B
dsp_data[4]	H'66	Port 5 output data for displaying ”4” on a LED	1 byte	H'E1C
dsp_data[5]	H'6D	Port 5 output data for displaying ”5” on a LED	1 byte	H'E1D
dsp_data[6]	H'7D	Port 5 output data for displaying ”6” on a LED	1 byte	H'E1E
dsp_data[7]	H'27	Port 5 output data for displaying ”7” on a LED	1 byte	H'E1F
dsp_data[8]	H'7F	Port 5 output data for displaying ”8” on a LED	1 byte	H'E20
dsp_data[9]	H'6F	Port 5 output data for displaying ”9” on a LED	1 byte	H'E21

Table 8 Description of Data Table (led_data[]) for 7-segment LED Display Switching

Element	Data	Description	Data size	Address
dsp_data[0]	H’08	Port 1 output data that turns on LED1 (P14 = 0, P15 = 0)	1 byte	H’E22
dsp_data[1]	H’18	Port 1 output data that turns on LED2 (P14 = 0, P15 = 1)	1 byte	H’E23
dsp_data[2]	H’28	Port 1 output data that turns on LED3 (P14 = 1, P15 = 0)	1 byte	H’E24
dsp_data[3]	H’38	Port 1 output data that turns on LED4 (P14 = 1, P15 = 1)	1 byte	H’E25

5. Flowchart

1. Main Routine (main)



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

RAM area initial settings

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 the LED enable counter = 0
SCALE = 0.00067	: Sensitivity constant of angular velocity
RATE = 10.0/101.0	: Amplification circuit coefficient

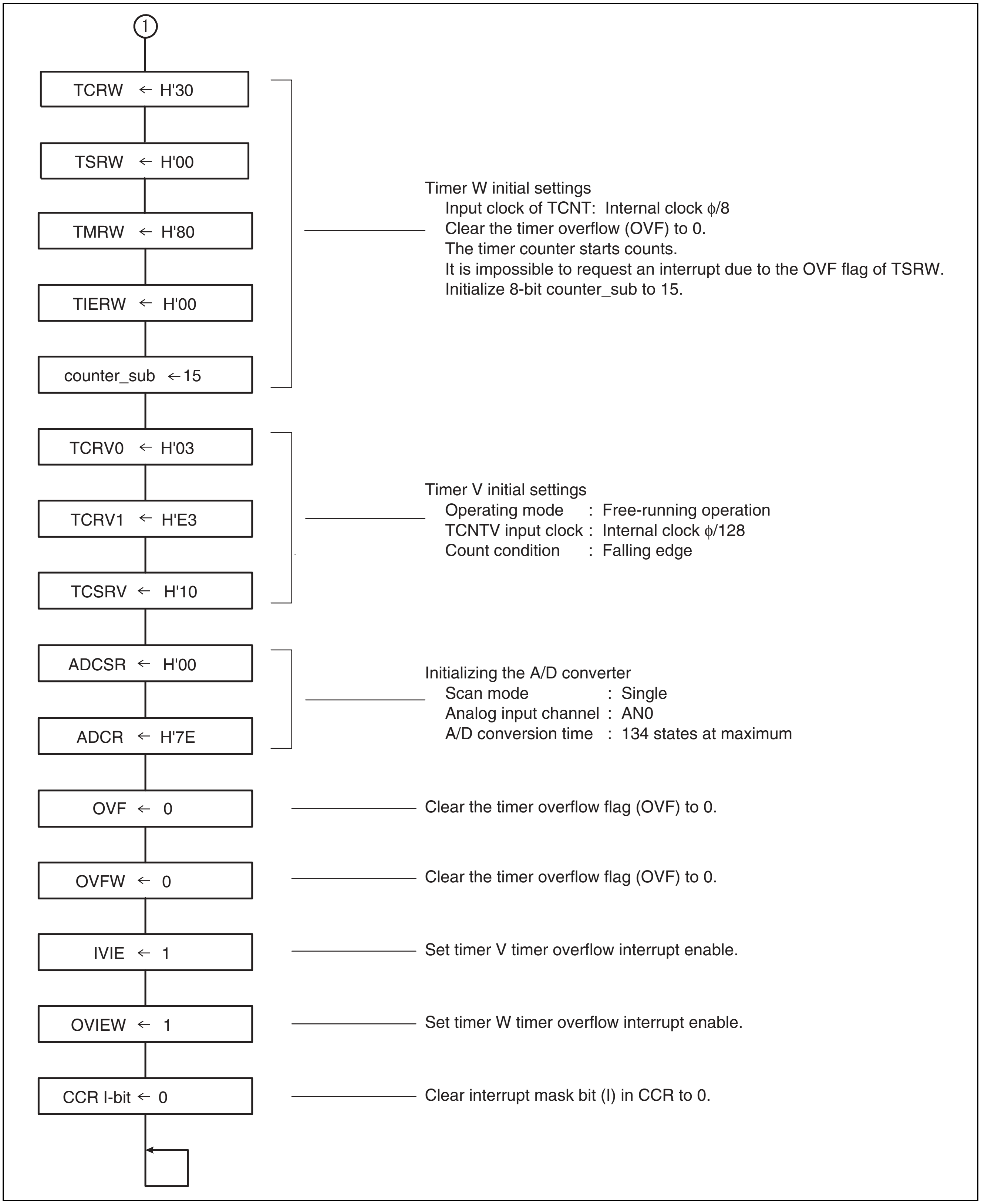
Port 1 initial settings

P10	: P10 input pin (Pull-up MOS = OFF)
P11	: P11 input pin (Pull-up MOS = OFF)
P12/SCK3_3	: P12 input pin (Pull-up MOS = OFF)
P14/_IRQ0	: P14 output pin (Pull-up MOS = OFF)
P15	: P15 output pin (Pull-up MOS = OFF)
P16	: P16 input pin (Pull-up MOS = OFF)
P17/_IRQ0/TRGV	: P17 input pin (Pull-up MOS = OFF)

Port 5 initial settings

P50/_WKP0	: P50 output pin (output data = 0)
P51/_WKP1	: P51 output pin (output data = 0)
P52/_WKP2	: P52 output pin (output data = 0)
P53/_WKP3	: P53 output pin (output data = 0)
P54/_WKP4	: P54 output pin (output data = 0)
P55/_WKP5/_ADTRG	: P55 output pin (output data = 0)
P56	: P56 output pin (output data = 0)
P57	: P57 output pin (output data = 0)

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



Timer W initial settings
Input clock of TCNT: Internal clock ϕ/8
Clear the timer overflow (OVF) to 0.
The timer counter starts counts.
It is impossible to request an interrupt due to the OVF flag of TSRW.
Initialize 8-bit counter_sub to 15.

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

Initializing the A/D converter
Scan mode : Single
Analog input channel : AN0
A/D conversion time : 134 states at maximum

Clear the timer overflow flag (OVF) to 0.

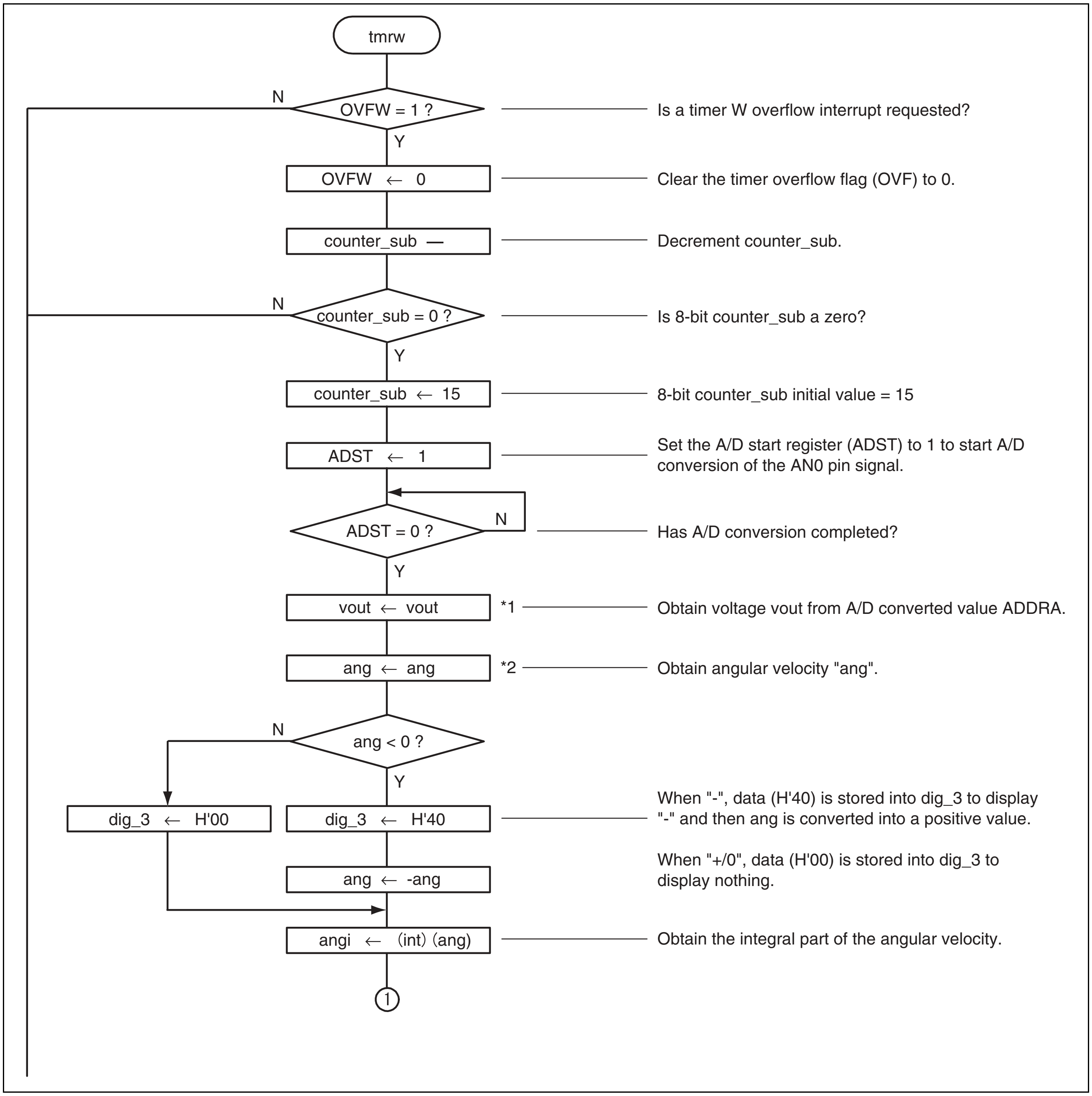
Clear the timer overflow flag (OVF) to 0.

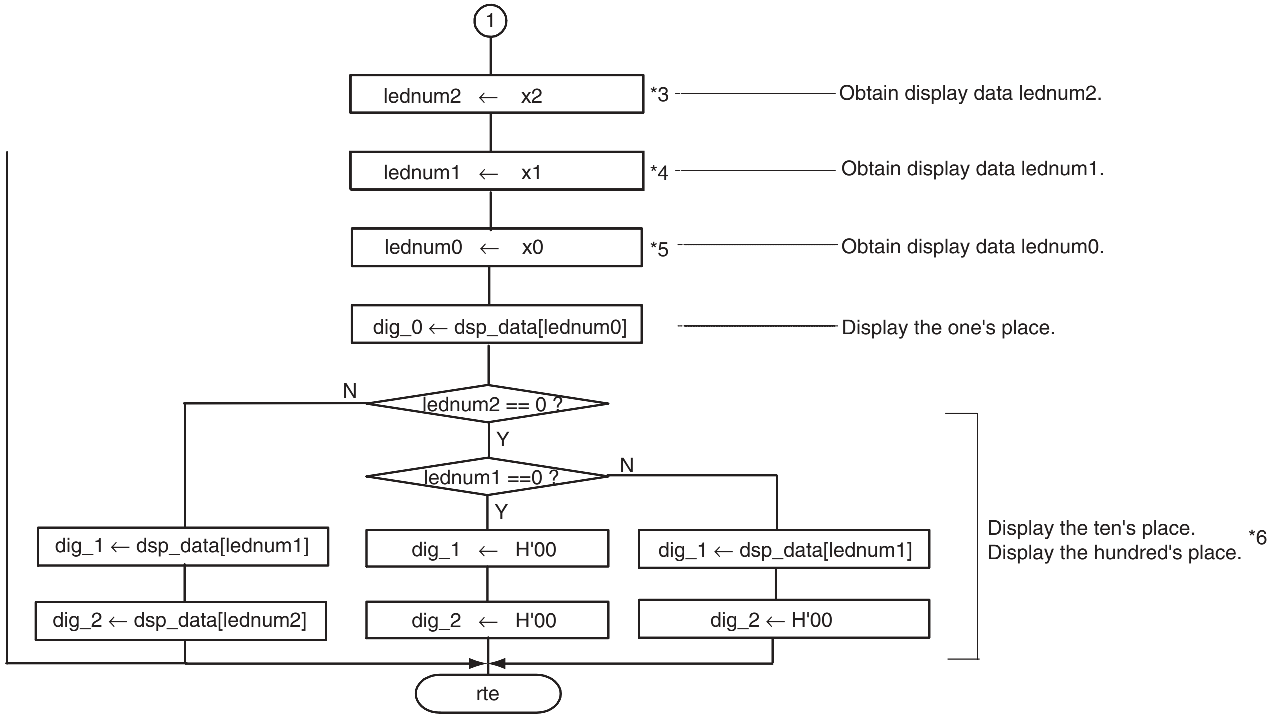
Set timer V timer overflow interrupt enable.

Set timer W timer overflow interrupt enable.

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

2. Timer W Interrupt Processing Routine (tmrw)

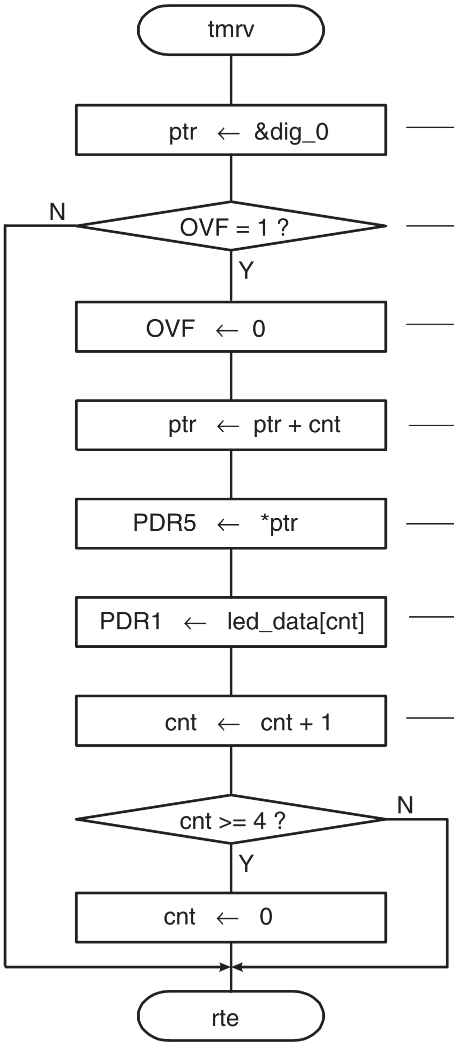

tmrw
OVFW = 1 ?
N
Y
Is a timer W overflow interrupt requested?
OVFW ← 0
Clear the timer overflow flag (OVF) to 0.
counter_sub —
Decrement counter_sub.
counter_sub = 0 ?
N
Y
Is 8-bit counter_sub a zero?
counter_sub ← 15
8-bit counter_sub initial value = 15
ADST ← 1
Set the A/D start register (ADST) to 1 to start A/D conversion of the AN0 pin signal.
ADST = 0 ?
N
Y
Has A/D conversion completed?
vout ← vout
*1
Obtain voltage vout from A/D converted value ADDRA.
ang ← ang
*2
Obtain angular velocity "ang".
ang < 0 ?
N
Y
dig_3 ← H'00
dig_3 ← H'40
When "-", data (H'40) is stored into dig_3 to display "-" and then ang is converted into a positive value.
ang ← -ang
When "+/0", data (H'00) is stored into dig_3 to display nothing.
angi ← (int) (ang)
Obtain the integral part of the angular velocity.
1


1
lednum2 ← x2
*3
Obtain display data lednum2.
lednum1 ← x1
*4
Obtain display data lednum1.
lednum0 ← x0
*5
Obtain display data lednum0.
dig_0 ← dsp_data[lednum0]
Display the one's place.
N
lednum2 == 0 ?
Y
lednum1 ==0 ?
N
Y
dig_1 ← dsp_data[lednum1]
dig_1 ← H'00
dig_1 ← dsp_data[lednum1]
dig_2 ← dsp_data[lednum2]
dig_2 ← H'00
dig_2 ← H'00
Display the ten's place.
Display the hundred's place.
*6
rte

[Note]
*1 : vout= (ADDRA >> 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 V Interrupt Processing Routine (tmrv)



Store the address of dig_0 in ptr.

Is a timer V timer overflow interrupt requested?

Clear the timer overflow flag (OVF) to 0.

Add the cnt value to the address of dig_0 and store the result in ptr.

Store the contents of the address indicated by ptr in PDR5.

Reference the data for the cnt value and then store it in PDR1.

Increment cnt.

cnt >= 4 ?

Initialize cnt.

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 */
/* Example of connecting the GYROSTAR angular velocity 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 PDR1   *(volatile unsigned char *)0xFFD4   /* Port data register 1 */
#define PCR1   *(volatile unsigned char *)0xFFE4   /* Port control register 1 */
#define PMR1   *(volatile unsigned char *)0xFFE0   /* Port mode register 1 */
#define PUCR1  *(volatile unsigned char *)0xFFD0   /* Port pull-up control register 1 */
#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 InterruptEnable 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 OVFTCSRV_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[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" */
};
const unsigned char led_data[4] =
{
    0x08,                                          /* Dig-0 LED enable data */
    0x18,                                          /* Dig-1 LED enable data */
    0x28,                                          /* Dig-2 LED enable data */
    0x38                                           /* Dig-3 LED enable data */
};
/* 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;                         /* Timer W interrupt counter */
float vout;                                        /* OPamp analog voltage output */
float ang;                                         /* Angular velocity result */
int angi;                                          /* 10times angular velocity result */
float SCALE;                                       /* Scale factor const  */
float RATE;                                        /* Amplify rate const*/
unsigned char *ptr;                                /* Pointer set */
unsigned char lednum0;                             /* Dig-0 LED display data */
unsigned char lednum1;                             /* Dig-1 LED display data */
unsigned char lednum2;                             /* Dig-2 LED display data */
/* 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 */
    SCALE = 0.00067;                               /* Scale factor const */
    RATE  = 10.0/101.0;                            /* Amplify rate const */

    PMR1  = 0x00;                                  /* Port 1 initialize */
    PUCR1 = 0x08;
    PDR1  = 0x08;
    PCR1  = 0x38;

    PDR5  = 0x00;                                  /* Port 5 initialize */
    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 = 0x00;                                  /* A/D converter initialize */
    ADCR  = 0x7e;
    OVF   = 0;                                     /* Clear OVF to 0 */
    OVFW  = 0;                                     /* Clear OVF to 0 */
    OVIE  = 1;                                     /* Timer V OVF interrupt enable */
    OVIEW = 1;                                     /* Timer W OVF interrupt enable */

    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 ? */
        vout = (ADDRA >> 6 & 0x03ff) * 3.3 / 1023.0;   /* change into voltage output */
        ang  = (vout - 1.35) * RATE / SCALE;       /* angular velocity */
        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 = 0x0                        /* 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 V Interrupt                                         */
/*************************************************************/
void tmrv(void)
{
    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 */
        PDR1 = led_data[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	Dec.20.03	—	First edition issued

Keep safety first in your circuit designs!

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

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.