To our customers,

Old Company Name in Catalogs and Other Documents

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

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

Notice

H8/300L SLP Series

Example of Connecting the TGS2600 Gas Sensor for Detecting Air Contaminants

Introduction

The voltage across a load resistor connected in series with the TGS2600 gas sensor is input to analog input pin AN0 to display a resistance value (kilo-ohms) in decimal on an array of 7-segment LEDs.

Target Device

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

Contents

1. Specifications 2

2. Description of Functions 8

3. Description of Operation 11

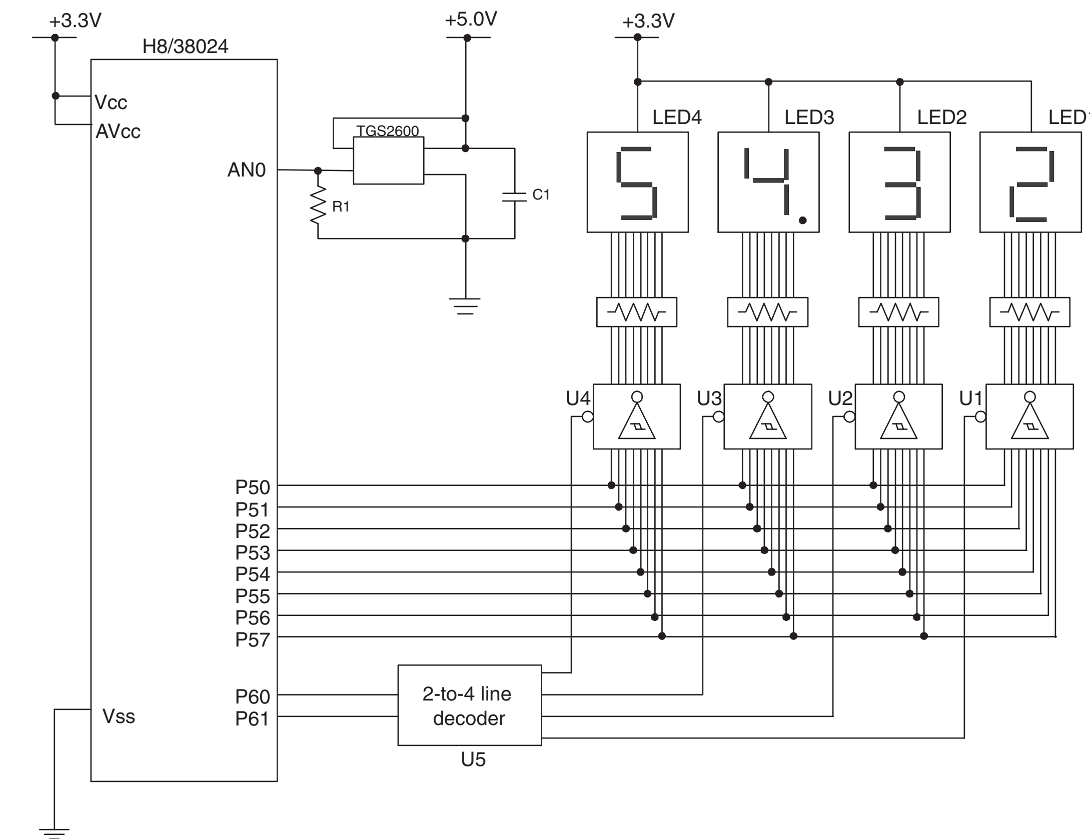
4. Description of Software 13

5. Flowchart 18

6. Program Listing 23

1. Specifications

1. Figure 1 shows an example of the hardware configuration of a metal oxide semiconductor gas sensor. As shown in the 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 the resistance value in decimal. A/D conversion is performed at intervals of 0.5 seconds.



Legend:

TGS2600	: Gas sensor (TGS2600, manufactured by FIGARO Engineering Inc.)
R1	: Resistor (R1 = 10 kΩ, for reference)
C1	: Capacitor (C1 = 0.1 µF, for reference)
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 (V_{CC}) and analog power supply voltage (AV_{CC}) 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 gas sensor used in this sample task is a metal oxide semiconductor gas sensor (model TGS2600) manufactured by FIGARO Engineering Inc. The specifications of the sensor are listed below.
   A. Table 1 lists the specifications of the metal oxide semiconductor gas sensor.

Table 1 Specifications of the TGS2600 (Reference Values)

Model number			TGS2600	
Sensing element type			D1	
Standard package			T0-5 metal can	
Target gases			Hydrogen, alcohol, etc.	
Detection range			1 to 10 ppm	
Standard circuit conditions	Heater voltage	V_H	5.0 ±0.2 VDC/VAC	
	Circuit voltage	V_C	5.0 ±0.2 VDC	$P_S \leq$ 15 mW
	Load resistance	R_L	Variable	$P_S \leq$ 15 mW
Electrical characteristics under standard test conditions	Heater resistance	R_H	83Ω (room temp.)	
	Heater current	I_H	42 mA	
	Heater power consumption	P_H	210 mW	V_H = 5.0 VDC/VAC
	Sensor resistance	R_S	10 to 90 kΩ in air	
	Sensitivity (change ratio of R_S)		0.3 to 0.6	R_S (10 ppm of H_2) / RS (air)
Standard test conditions	Test gas conditions		At 20 ±2°C, 65 ±5% RH	
	Circuit conditions		V_C = 5.0 ±0.2 VDC V_H = 5.0V ±0.2 VDC/VAC	
	Heating time before test		At least 36 hours	

   B. Figure 2 shows a basic measurement circuit for the sensor.
   This sensor requires two voltage inputs: the heater voltage (V_H) and circuit voltage (V_C). The following example uses a common power circuit.

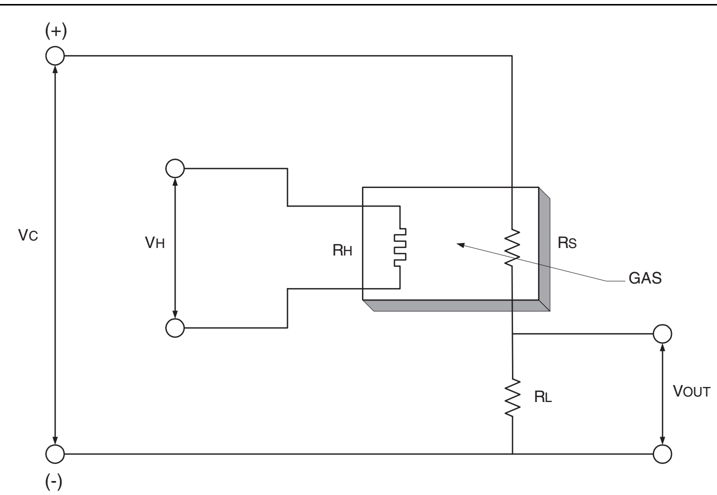
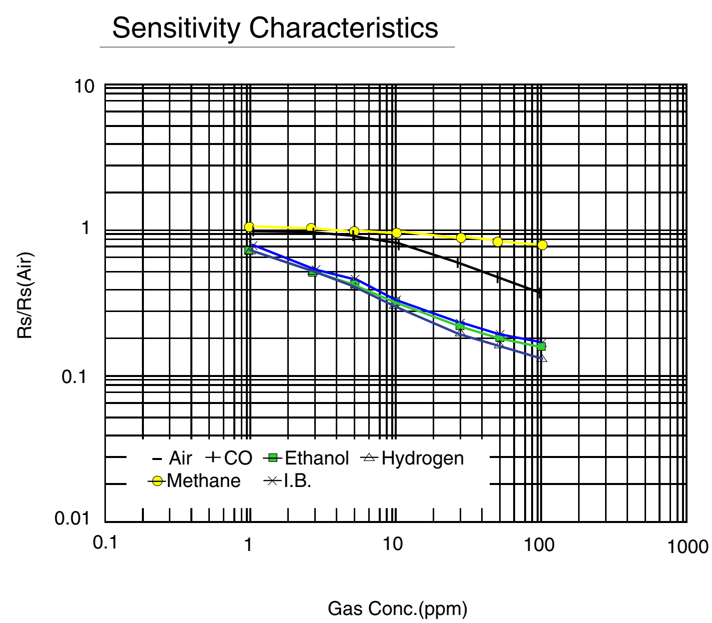


Figure 2 Sample Basic Measurement Circuit for the Sensor

C. Figure 3 shows the sensitivity characteristics and sensitivity to cigarette smoke.



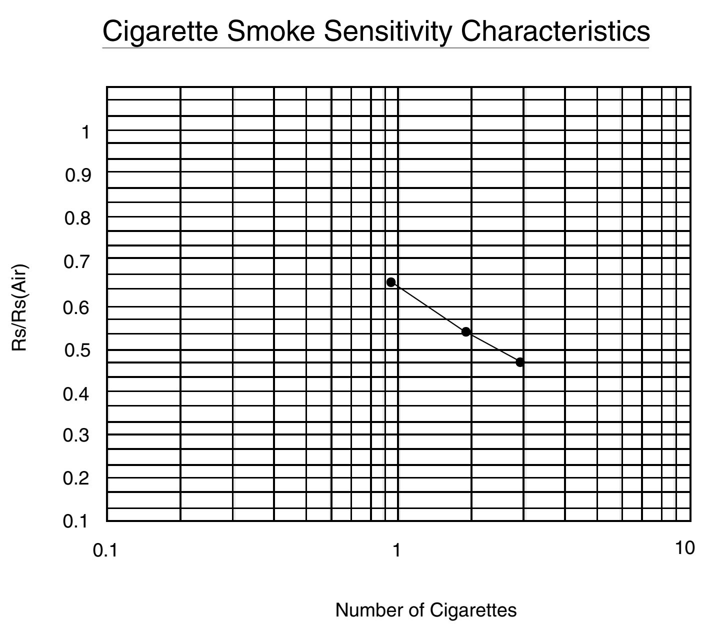
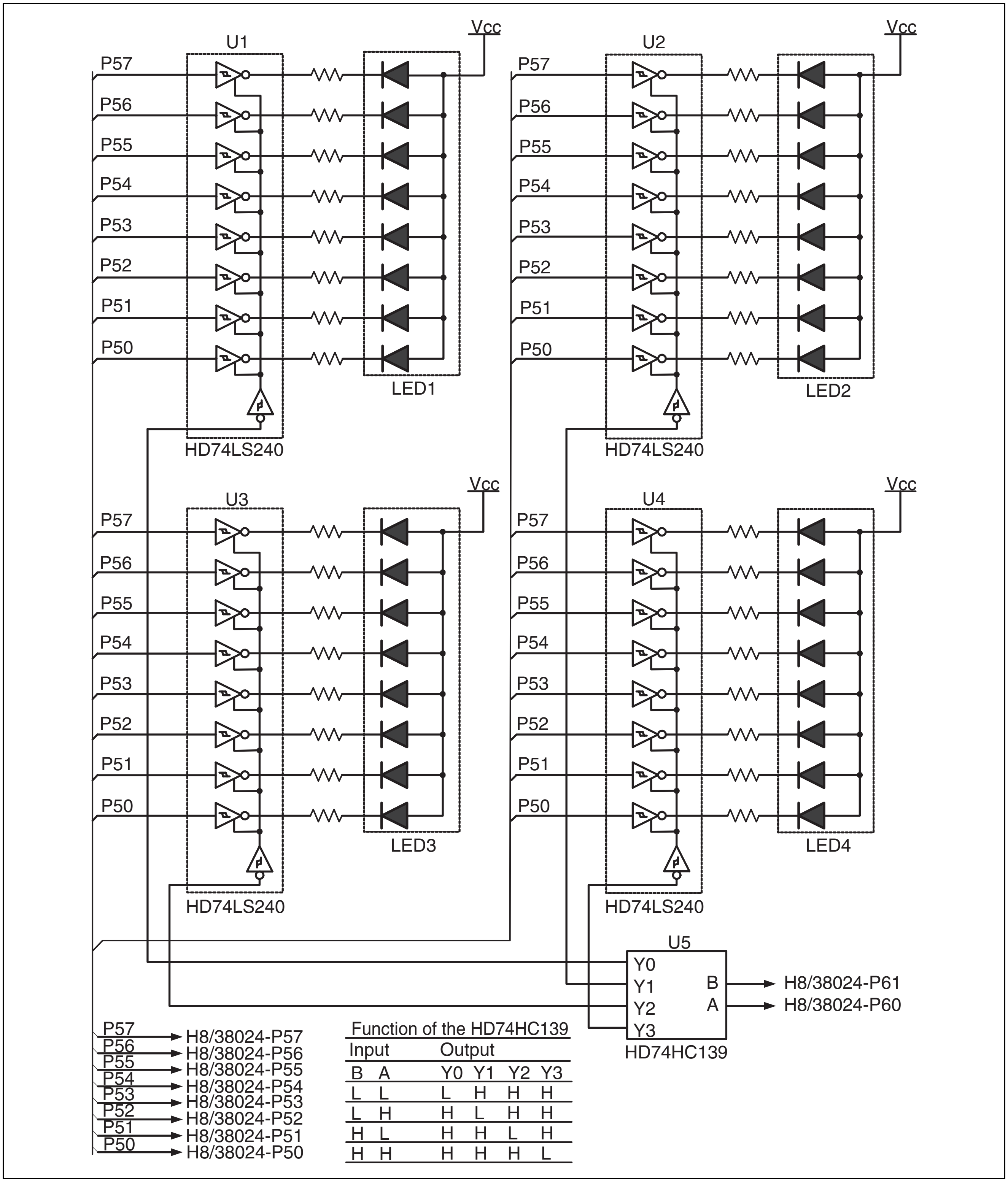


Figure 3 Sensor Sensitivity Characteristics (Reference Values)

D. The sensing element of the TGS2600 consists of a metal oxide semiconductor layer formed on an alumina substrate together with an integral heater. When a detectable gas is present, the sensor conductivity increases together with the amount of gas in the air. This sensor detects relative values and is highly sensitive to cigarette smoke and cooking odors. The sensor is used for controlling air conditioners or ventilators as well as monitoring room air.

6. The circuit in this sample task operates as follows.
   A. The TGS2600 divides the applied voltage (V_C) between the sensor resistor (RS: 10 to 90 kΩ) and the load resistor (RL: 10 kΩ) and then generates an output voltage (Vout).
   B. The number "54.32" in Figure 1 indicates that the sensor resistance is 54.32 kΩ.
   C. Blowing a gas such as cigarette smoke over the sensor causes the gas concentration to rise and thus reduces the sensor resistance. The value displayed by the LEDs changes from "54.32" to, for example, "41.32".
7. 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 4 illustrates the method of controlling the 7-segment LEDs.



Function of the HD74HC139

Input B	Input A	Output Y0	Output Y1	Output Y2	Output 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 4 Method of Controlling 7-Segment LEDs

8. In this sample task, the sensor resistance (kΩ) can be given to an accuracy of two decimal places on the 7-segment LEDs. Figure 5 illustrates how the resistance is displayed on the LEDs.

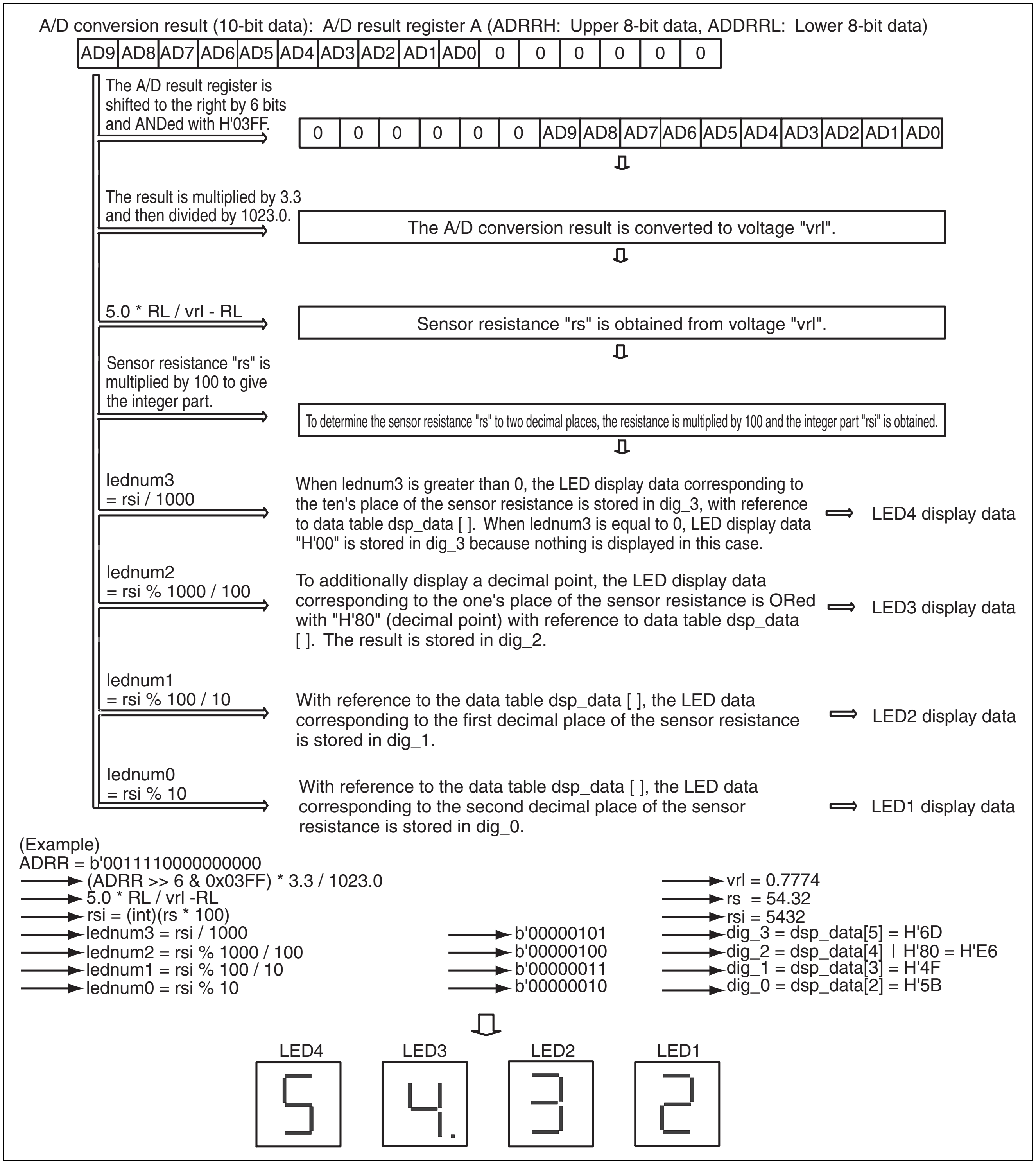


Figure 5 How the Resistance is Displayed on the LEDs

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.

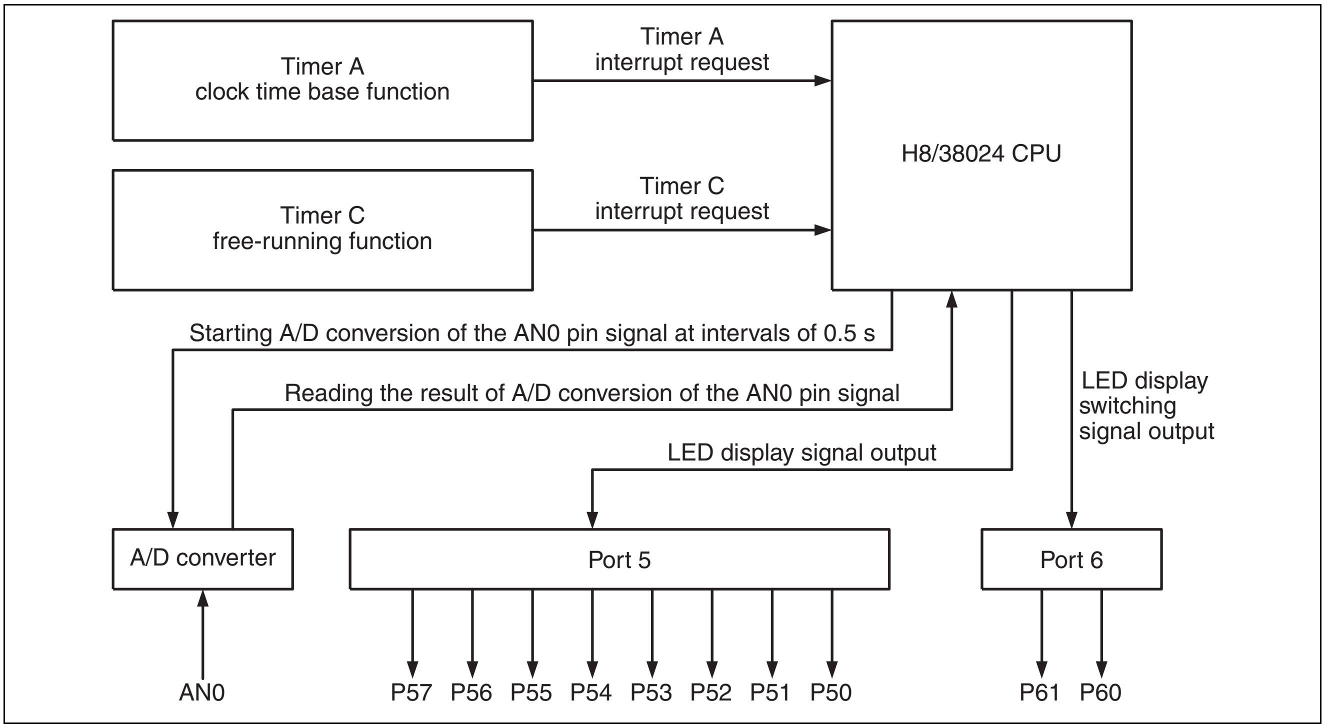


Figure 6 Block Diagram of Functions Used in This Sample Task

Table 2 Function Assignments

Function	Function assignment
Timer A	The clock time base function of timer A is used to measure the 0.5-second interval at which A/D conversion is applied to the signal that is input to analog input pin 0 (AN0). Timer A interrupt is used to initiate A/D conversion.
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 voltage across the load resistor that is connected in series with the TGS2600 sensor connected to analog input pin 0 (AN0) of the A/D converter.
Port 5	The P50 to 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 sensor resistance display data (with decimal places) 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 7 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.

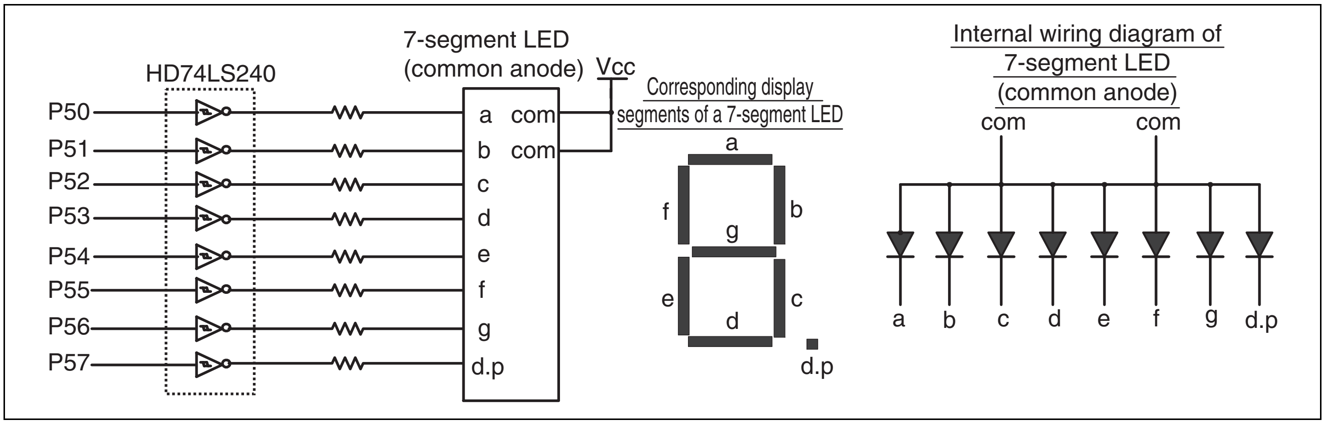
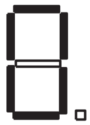
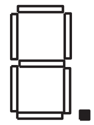
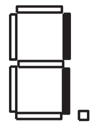
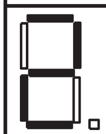
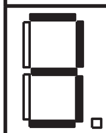
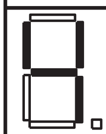
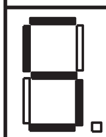
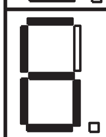
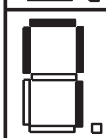
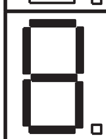
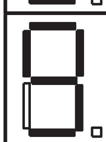


Figure 7 Connection Diagram and Internal Connections of 7-Segment LED

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									

Note: The first digit of the integer part is ORed with the decimal point.

3. Description of Operation

1. Figure 8 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 8.

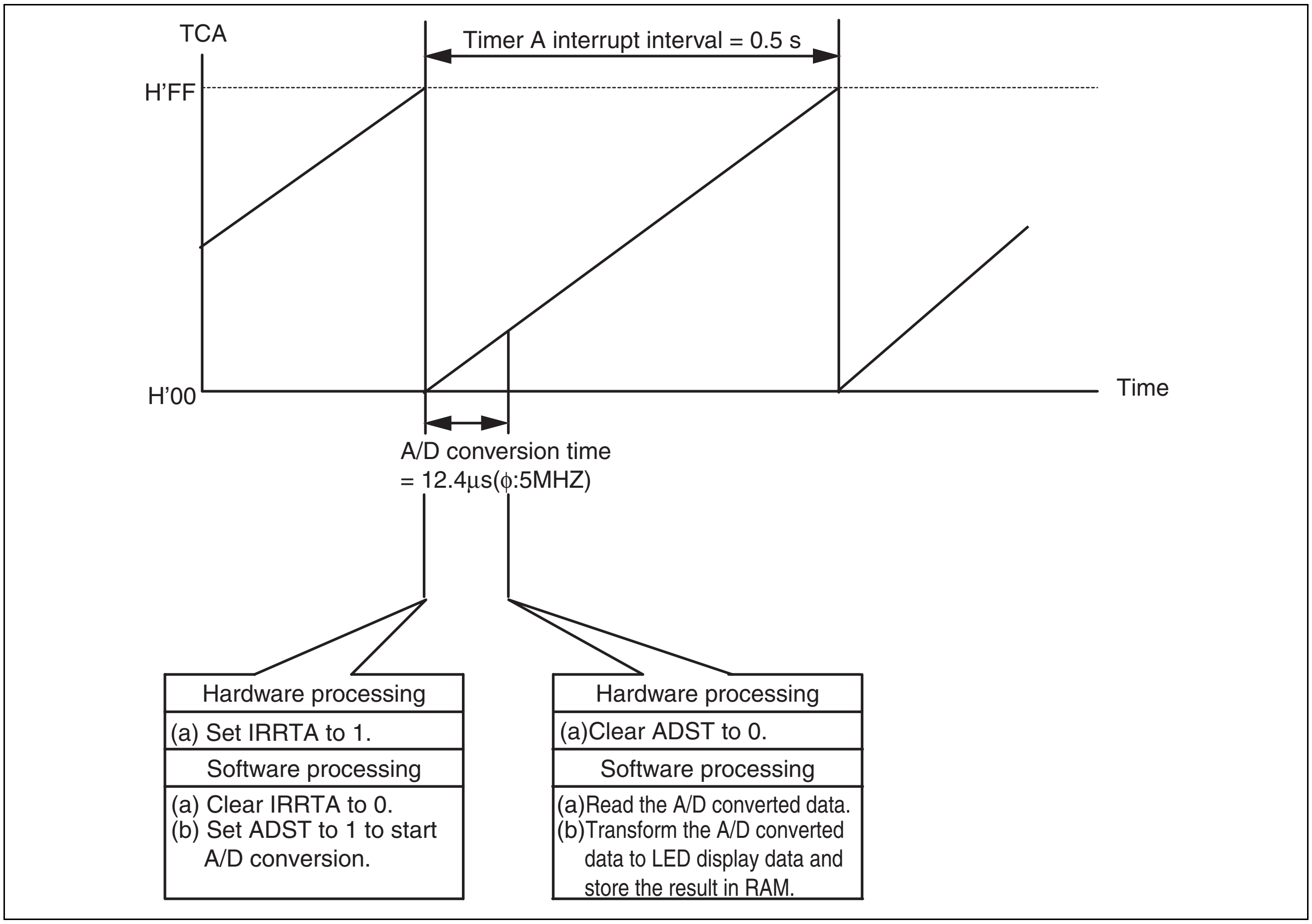


Figure 8 Description of A/D Conversion of the AN0 Pin Signal when Timer A is Used

2. The following describes the descriptions of 7-segment LED operation. Figure 9 shows how a value of "54.32" 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.

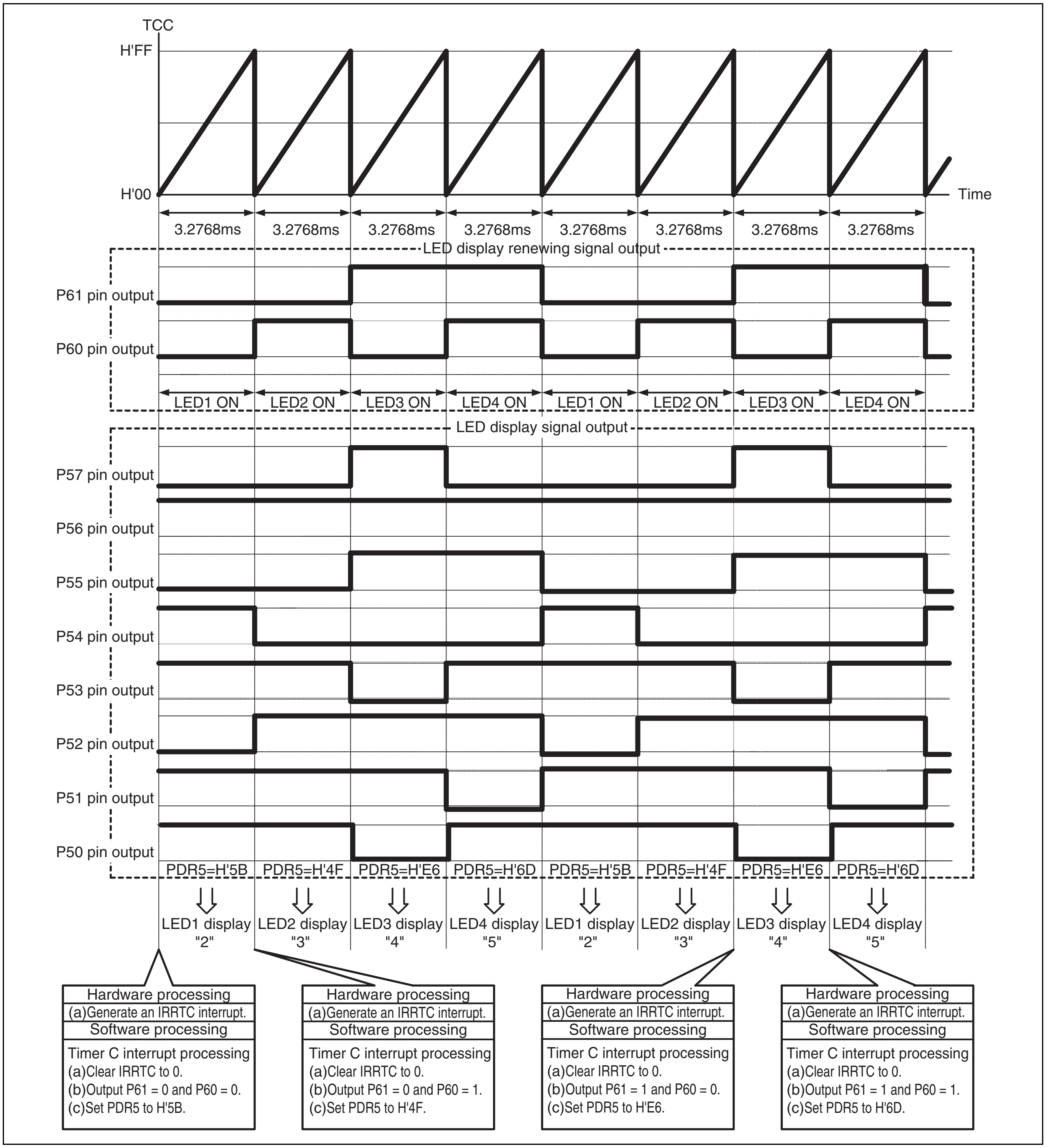


Figure 9 Description of 7-Segment LED Display Control

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 switching.

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 to 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, it selects the interval function.	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 $\phi/64$.	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 to 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 to 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 to P60 pins function as general output pins.	H'FFE9	H'FF

Register name		Description	Address	Setting
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
	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 to 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 to 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

Register name		Description	Address	Setting
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
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
vrl	Voltage of load resister RL	H'FB80	tmra
rs	Resistance of the sensor	H'FB84	tmra
RL	Load resistance	H'FB88	tmra
rsi	Resistance of × 100 sensor	H'FB8C	tmra
ptr	Location where the address of dig_0 is stored	H'FB8E	tmrc
dig_0	Stores LED1 display data (1 byte)	H'FB90	main, tmra, tmrc
dig_1	Stores LED2 display data (1 byte)	H'FB91	main, tmra
dig_2	Stores LED3 display data (1 byte)	H'FB92	main, tmra
dig_3	Stores LED4 display data (1 byte)	H'FB93	main, tmra
cnt	8-bit counter for switching display LED1 to LED4 (1 byte)	H'FB94	main, tmrc
lednum0	LED1 display data	H'FB95	tmra
lednum1	LED2 display data	H'FB96	tmra
lednum2	LED3 display data	H'FB97	tmra
lednum3	LED4 display data	H'FB98	tmra

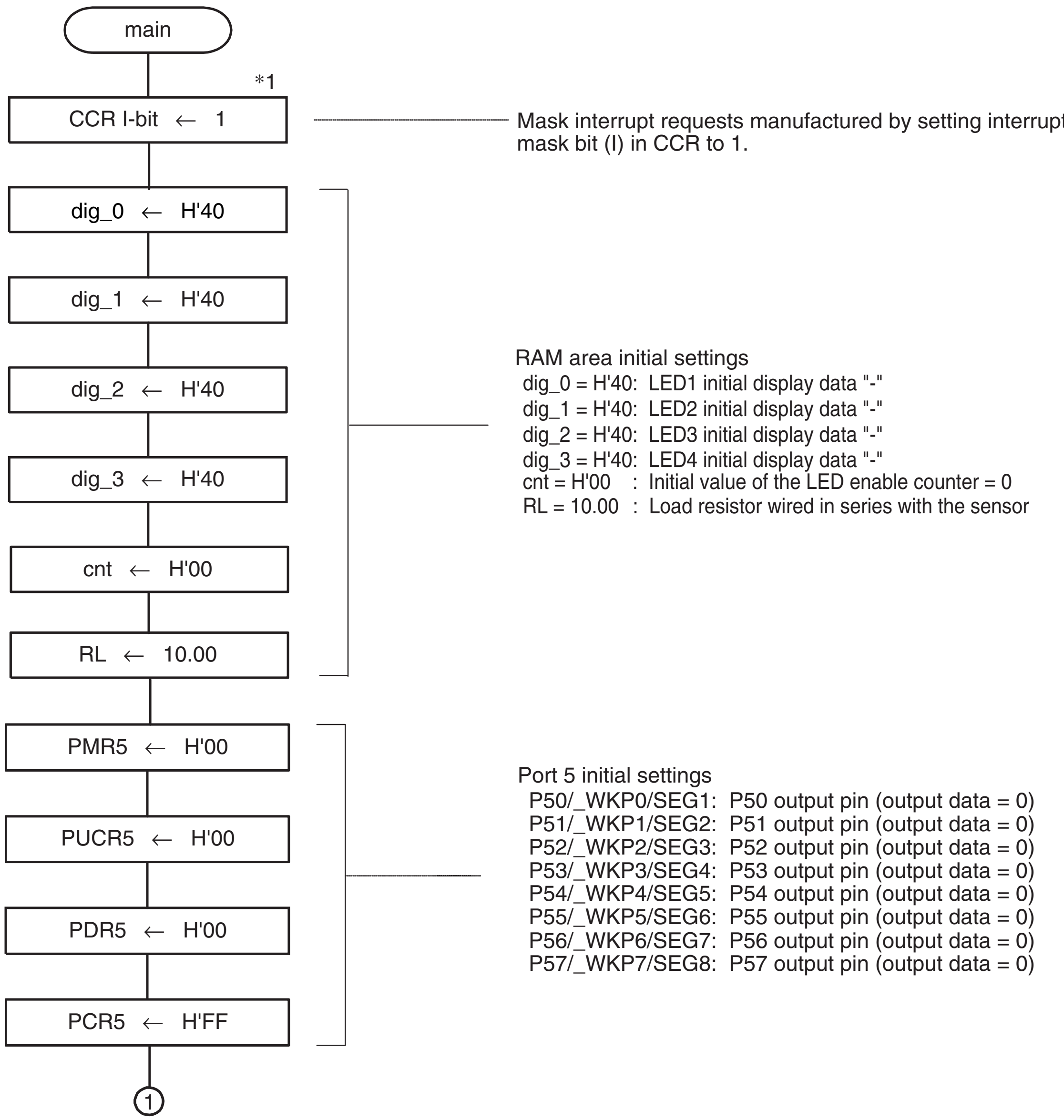
5. Description of data tables
   In this sample task, the display data for the 7-segment LEDs is stored in ROM as a 1-dimensional array (data table). Table 7 describes the data table for the 7-segment LED display (dsp_data [ ]).

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

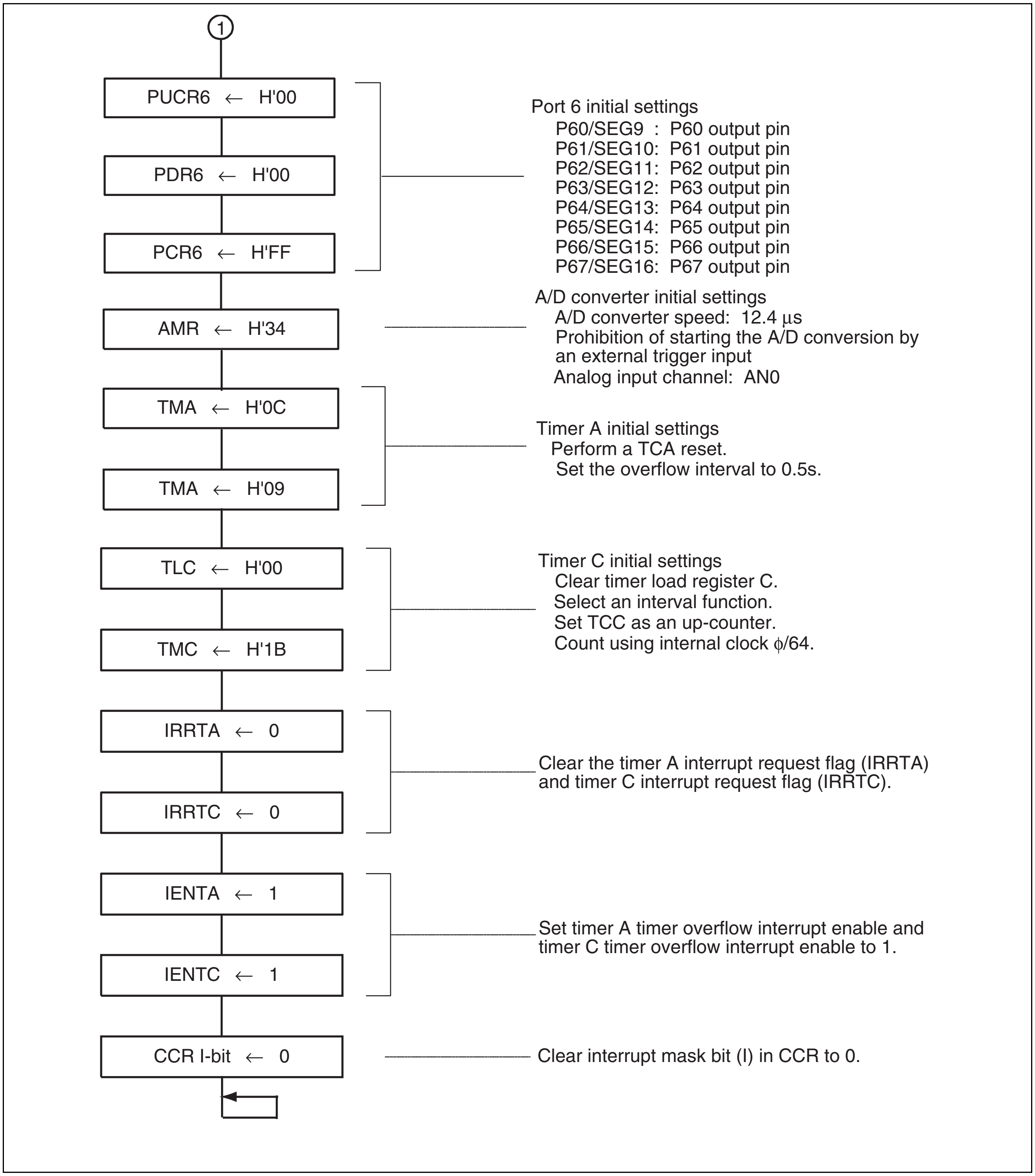
Element	Data	Description	Data size	Address
dsp_data[0]	H'3F	Port 5 output data for displaying "0" on a LED	1 byte	H'FF0
dsp_data[1]	H'06	Port 5 output data for displaying "1" on a LED	1 byte	H'FF1
dsp_data[2]	H'5B	Port 5 output data for displaying "2" on a LED	1 byte	H'FF2
dsp_data[3]	H'4F	Port 5 output data for displaying "3" on a LED	1 byte	H'FF3
dsp_data[4]	H'66	Port 5 output data for displaying "4" on a LED	1 byte	H'FF4
dsp_data[5]	H'6D	Port 5 output data for displaying "5" on a LED	1 byte	H'FF5
dsp_data[6]	H'7D	Port 5 output data for displaying "6" on a LED	1 byte	H'FF6
dsp_data[7]	H'27	Port 5 output data for displaying "7" on a LED	1 byte	H'FF7
dsp_data[8]	H'7F	Port 5 output data for displaying "8" on a LED	1 byte	H'FF8
dsp_data[9]	H'6F	Port 5 output data for displaying "9" on a LED	1 byte	H'FF9

5. Flowchart

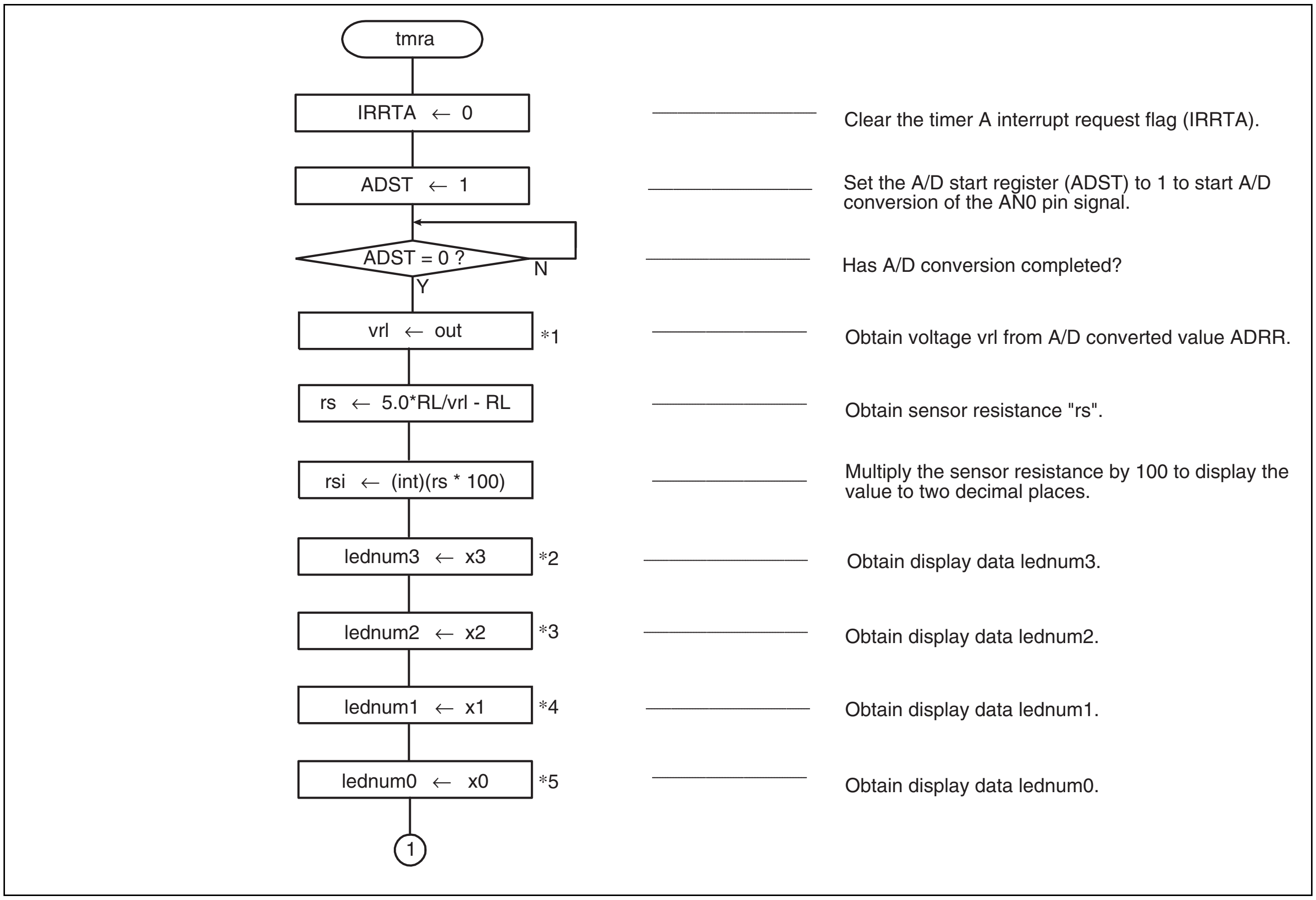
1. Main Routine (main)

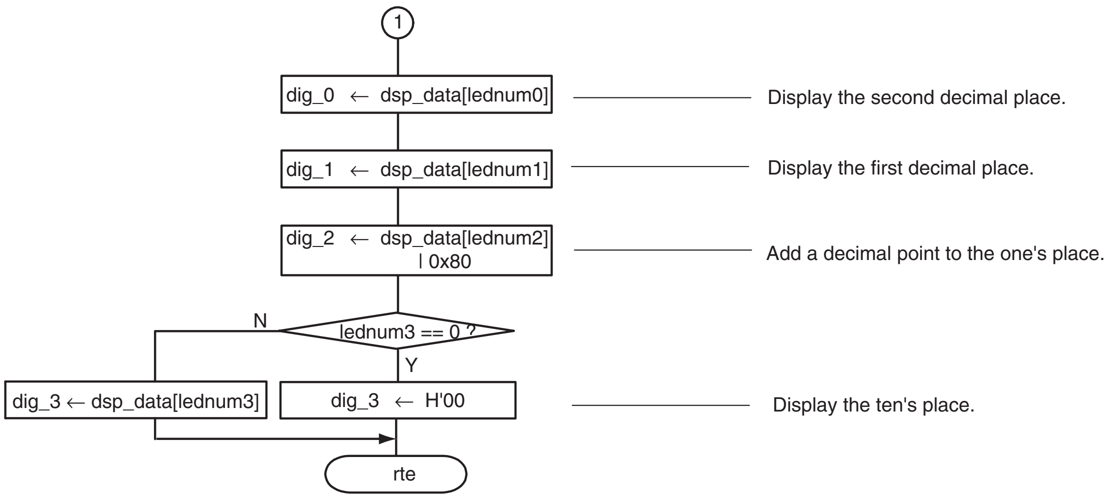


*1 In this sample task, INIT.SRC (assembly language) is used to set the stack pointers.


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 0.5s.
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.

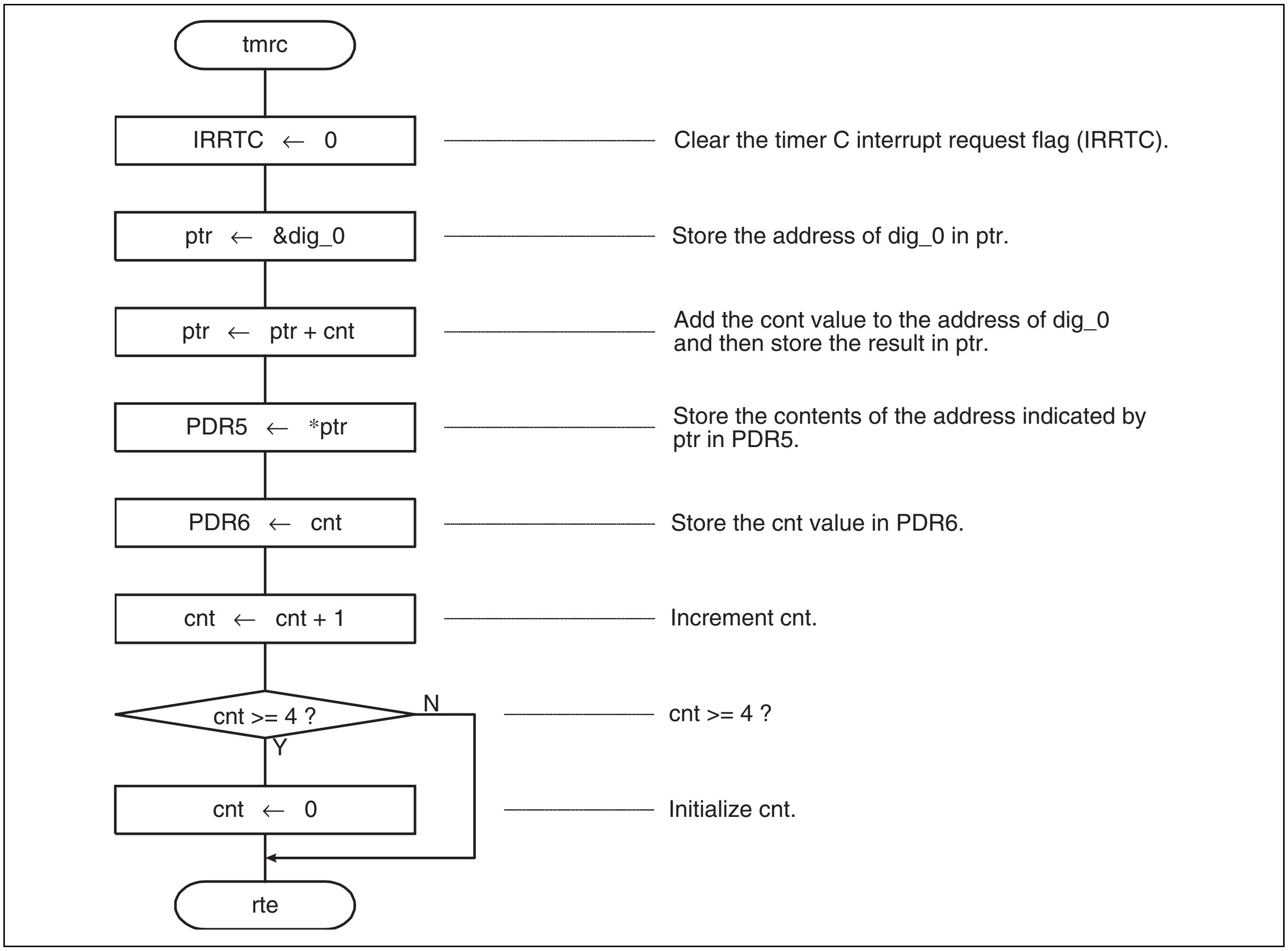
2. Timer A Interrupt Processing Routine (tmra)





*1 : out = (ADRR >> 6 & 0x03ff) * 3.3 / 1023.0
*2 : x3 = (unsigned char)(rsi/ 1000)
*3 : x2 = (unsigned char)(rsi% 1000 / 100)
*4 : x1 = (unsigned char)(rsi% 100 / 10)
*5 : x0 = (unsigned char)(rsi % 10)

3. Timer C Interrupt Processing Routine (tmrc)



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
```

```
/* Super Low Power Series -H8/38024- Application note */
/* Application example */
/* Example of connecting a gas sensor for detecting air contaminants*/

#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)
#define ADST     ADSR_BIT.b7                         /* A/D start */
#define IRR1    *(volatile unsigned char *)0xFFF6    /* Interrupt request register 1 */
#define IRR1_BIT (*(struct BIT *)0xFFF6)
#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)
#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)
#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)
#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 */
float vrl;                                           /* output voltage(RL) */
float rs;                                            /* sensor resistance(unit:kO) */
float RL;                                            /* Used resistance(unit: kO) */
int   rsi;                                           /* 100 times sensor resistance */
unsigned char cnt;                                   /* LED enable counter */
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 */
unsigned char lednum3;                          /* Dig-3 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) = {
    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 */
    cnt   = 0x00;                               /* Used RAM area initialize */
    RL    = 10.00;                              /* Used resistance(unit:kO) */

    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 to 0 */
    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 ? */
    vrl = (ADRR >> 6 & 0x03ff) * 3.3 / 1023.0;      /* output voltage(RL) */
    rs  = 5.0 * RL / vrl - RL;                      /* sensor resistance(unit:kO) */
    rsi = (int)(rs * 100);                          /* 100 times sensor resistance */
    lednum3 = (unsigned char)(rsi / 1000);          /* compute Dig-3 LED display data */
    lednum2 = (unsigned char)(rsi % 1000 / 100);    /* compute Dig-2 LED display data */
    lednum1 = (unsigned char)(rsi % 100 / 10);      /* compute Dig-1 LED display data */
    lednum0 = (unsigned char)(rsi % 10);            /* compute Dig-0 LED display data */
    dig_0 = dsp_data[lednum0];                      /* Dig-0 LED display data set */
    dig_1 = dsp_data[lednum1];                      /* Dig-1 LED display data set */
    dig_2 = dsp_data[lednum2] | 0x80;               /* Dig-2 LED display data set */
    if (lednum3 == 0){
        dig_3 = 0x00;                               /* Dig-3 LED display data set */
    }else{
        dig_3 = dsp_data[lednum3];                  /* Dig-3 LED display data set */
    }
}

/************************************************************/
/* Timer C Interrupt                                        */
/************************************************************/
void tmrc(void)
{
    IRRTC = 0;                                      /* Clear IRRTC to 0 */

    ptr   = &dig_0;                                 /* LED display 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

Rev.	Date	Description	
		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.