To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

H8/300L SLP Series

Connecting a Thermistor

Introduction

A resistor and thermistor are connected to the analog input pin, and the results of A/D conversion are displayed on seven-segment LEDs.

Target Device

H8/38024

Contents

1. Specifications

1. Figure 1.1 shows the hardware configuration for an example of a thermistor connection. A resistor and NTC (negative temperature coefficient) thermistor are connected to the analog input pin 2 (AN2 pin) as shown in the figure.
2. The signal on the AN2 pin is A/D converted, after which the results of A/D conversion are displayed on the 7-segment LEDs connected to the I/O port.
3. The 7-segment LED display shows the 10-bit result of A/D conversion as a hexadecimal value in the range from H'000 to H'3FF.
4. A/D conversion is performed at 0.5-s intervals.

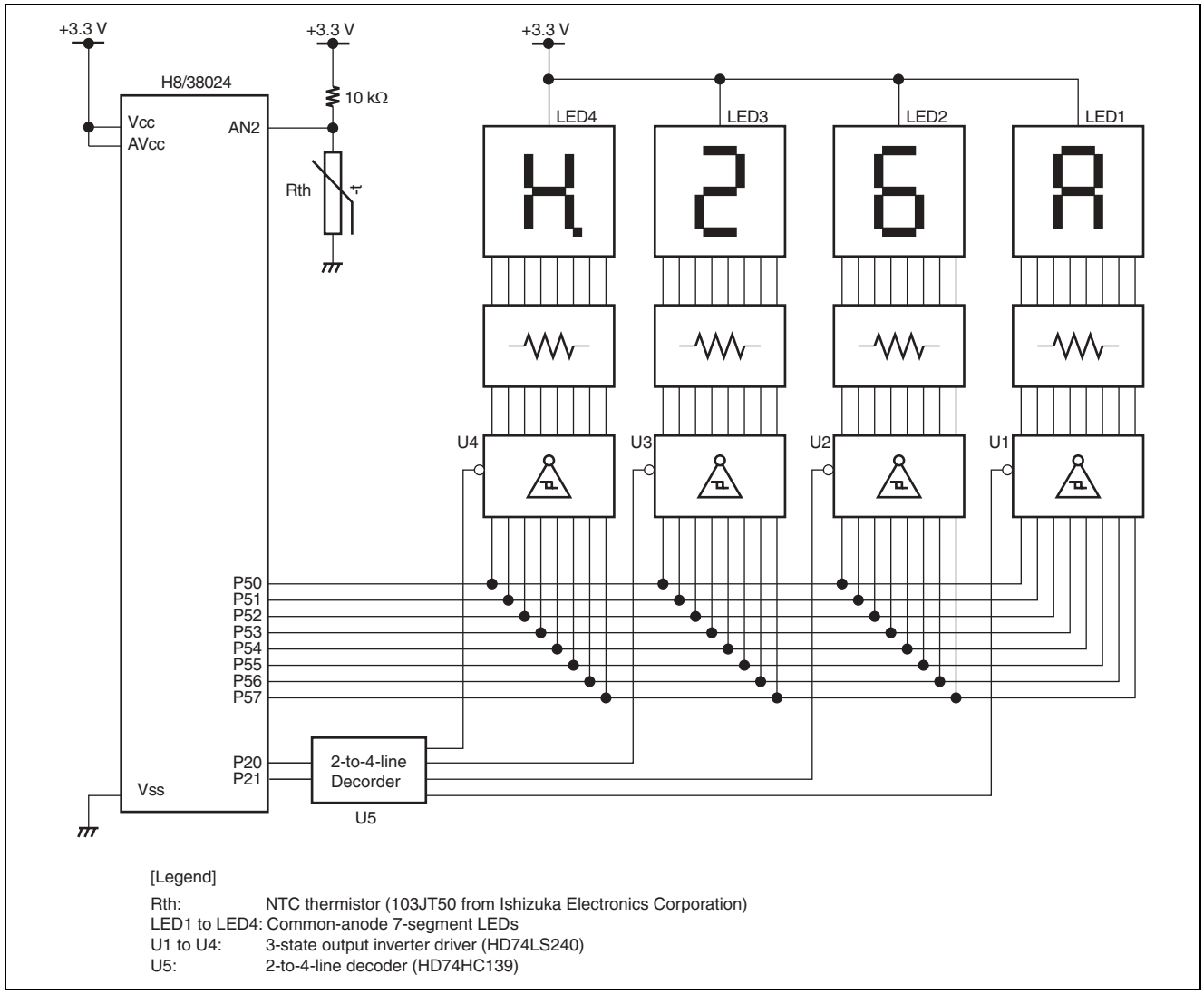


Figure 1.1 Hardware Configuration

5. In this sample task, the H8/38024's operating voltage (Vcc) and the analog power supply voltage (AVcc) are 3.3 V, the oscillation frequency of the system clock is 10 MHz and that of the sub-clock is 32.768 kHz.

6. The NTC thermistor used in this sample task is a high-precision ultra-thin model from Ishizuka Electronics Corporation (model 103JT-50). The specifications of the NTC thermistor are given below.
   A. Table 1.1 lists the ratings of the NTC thermistor (130JT-50).

Table 1.1 Ratings

R_{25}*1	Tolerance	B Constant*2	Thermal Dissipation Constant	Thermal Time Constant*3	Maximum Allowable Power	Temperature Range
10.0 kΩ	±1%	3435K±1%	Approx. 0.7 mW/°C	Approx. 5 s	3.5 mW at 25°C	–50 to +90°C

Notes: 1. Nominal zero load resistance value at 25°C
2. Calculated from the zero load resistance values at 25°C and 85°C
3. Measured in still air

   B. Table 1.2 shows the resistance vs. temperature characteristic data (reference values) of the NTC thermistor (103JT-50).

Table 1.2 Resistance vs. Temperature Characteristic Data (Reference Values)

Temperature (°C)	Resistance (kΩ)	Temperature (°C)	Resistance (kΩ)
–50	367.7	25	10.00
–45	272.6	30	8.301
–40	204.7	35	6.925
–35	154.9	40	5.811
–30	118.5	45	4.896
–25	91.27	50	4.147
–20	71.02	55	3.525
–15	55.43	60	3.011
–10	43.67	65	2.582
–5	34.63	70	2.224
0	27.70	75	1.922
5	22.29	80	1.668
10	18.07	85	1.451
15	14.74	90	1.267
20	12.11	—	—

7. In this sample task, display on the 7-segment LED is handled by attaching port outputs to the inputs to the tri-state-output inverter drivers (HD74LS240), and the driver outputs are in turn connected to the cathodes of the 7-segment LEDs. The port outputs are connected to each of the four 7-segment LEDs to control the display on the LEDs. The enable pins of the tri-state inverter driver control switching of display on the 7-segment LEDs. The signals used to switch the display are generated by the 2-to-4-line decoder (HD74HC139), which is controlled by two port-pin outputs. Figure 1.2 shows how the 7-segment LEDs are controlled.

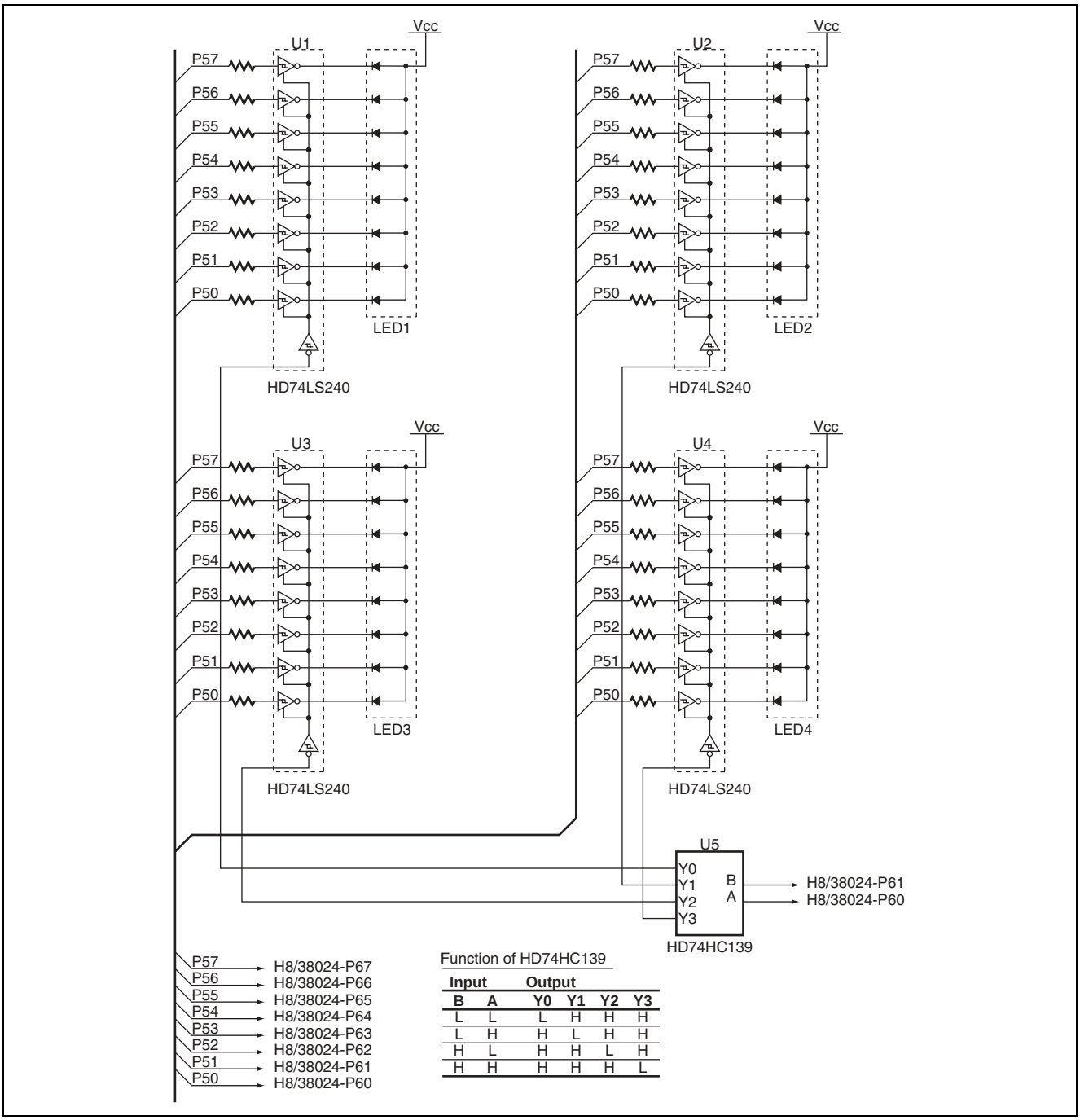


Figure 1.2 7-Segment LED Control

8. In this sample task, the results of A/D conversion are displayed in hexadecimal format (H'3FF to H'000) on the 7-segment LEDs. Figure 1.3 shows how this is done.

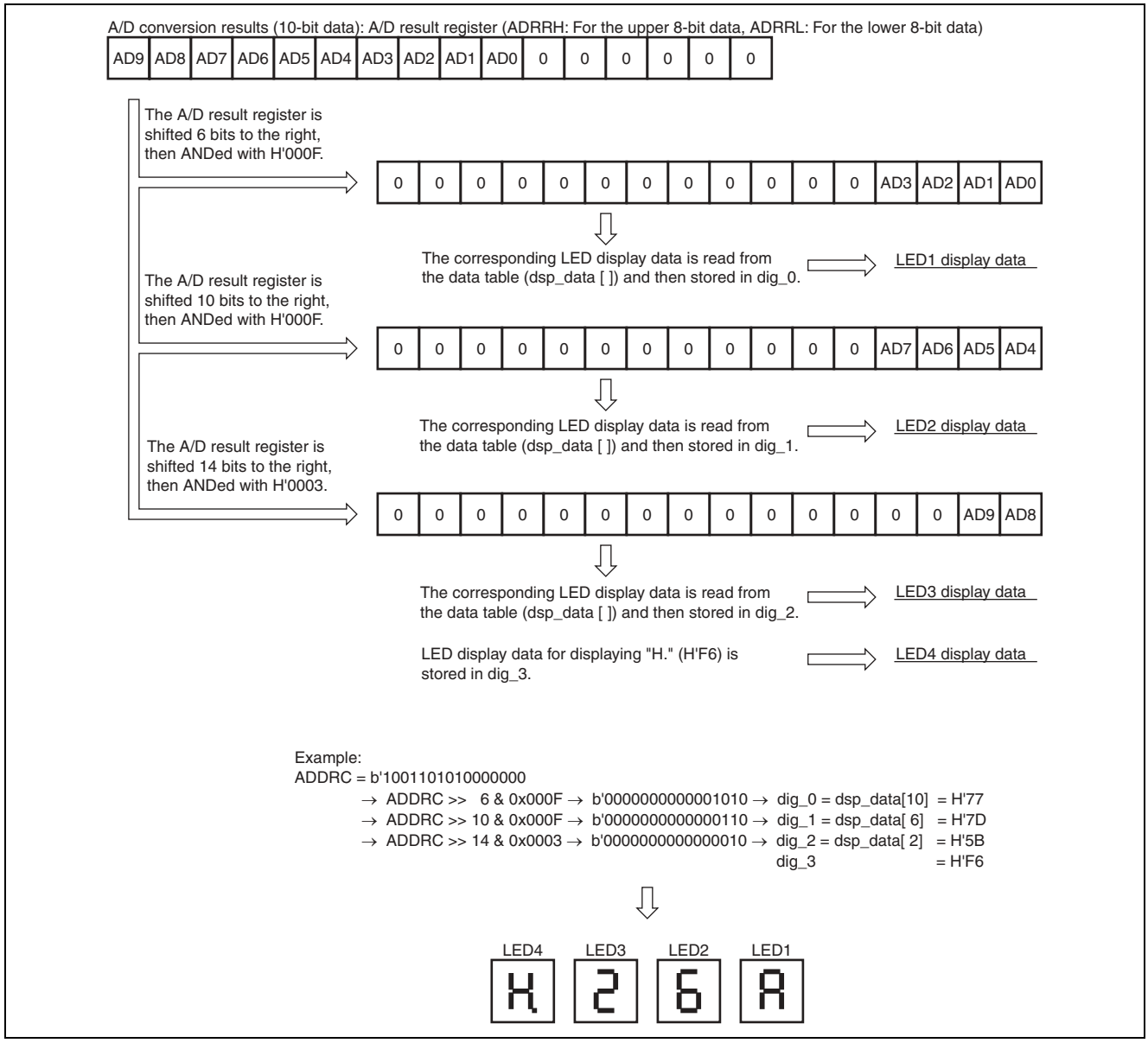


Figure 1.3 How A/D Conversion Results are Displayed on the LEDs

2. Description of Functions

1. Figure 2.1 is a block diagram of the H8/38024 functions used in this sample task. Table 2.1 shows function allocations.

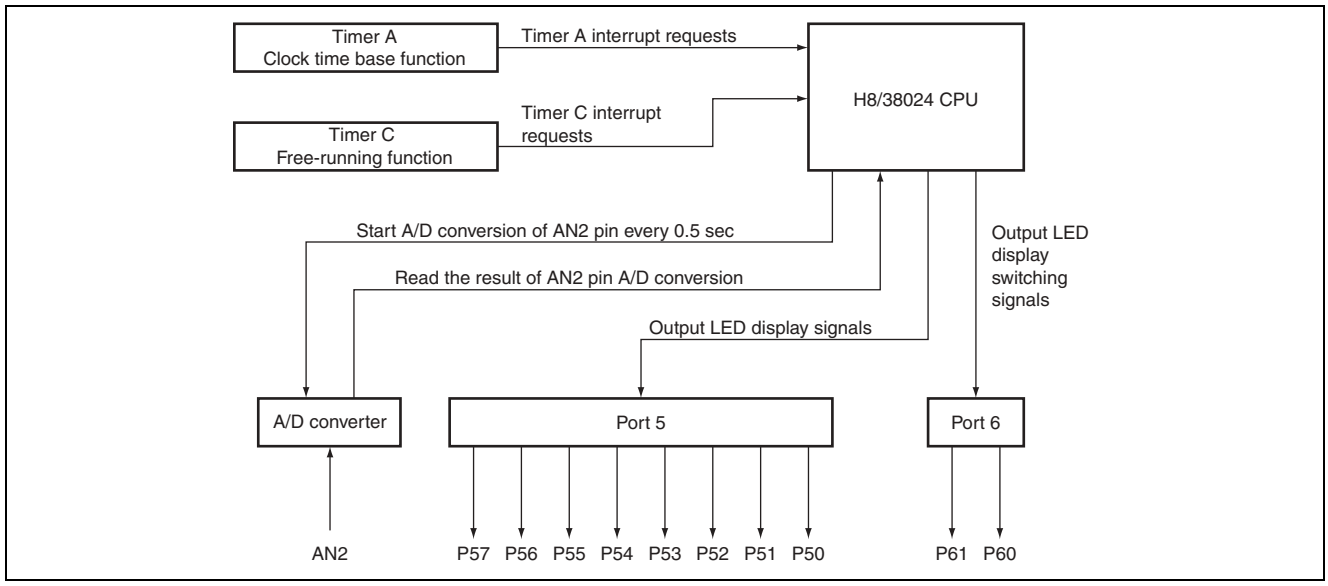


Figure 2.1 Block Diagram of Functions Used

Table 2.1 Function Allocation

Function	Function Allocation
Timer A	The timer A's clock time base function is used to measure 0.5 s, which is the period for A/D conversion of the signal on the analog input pin 2 (AN2). The timer A interrupt is used for each A/D conversion period.
Timer C	Timer C's free-running function is used to control switching of the 7-segment LED display. Each of the four 7-segment LEDs is lit in sequence at an interval of 3.2768 ms, which is the time taken for timer C to overflow. This obtains dynamic illumination from the LEDs.
A/D converter	This unit A/D-converts the result of dividing the voltage between the AVcc and GND planes by the thermally-variable resistance of the thermistor and the fixed resistance connected to analog input pin 2 (AN2) of the A/D converter.
Port 6	The four 7-segment LED display is switched by the P60 and P61 output pins of port 6. These pins are connected to the input/output pins of the 2-to-4-line decoder.
Port 5	The 7-segment LEDs are displayed by the P50 to P57 output pins of port 5. The 10 bits of data produced by A/D conversion of the value on the AN2 pin are converted to 3 digits of hexadecimal data for display, this is then output to the LED.

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

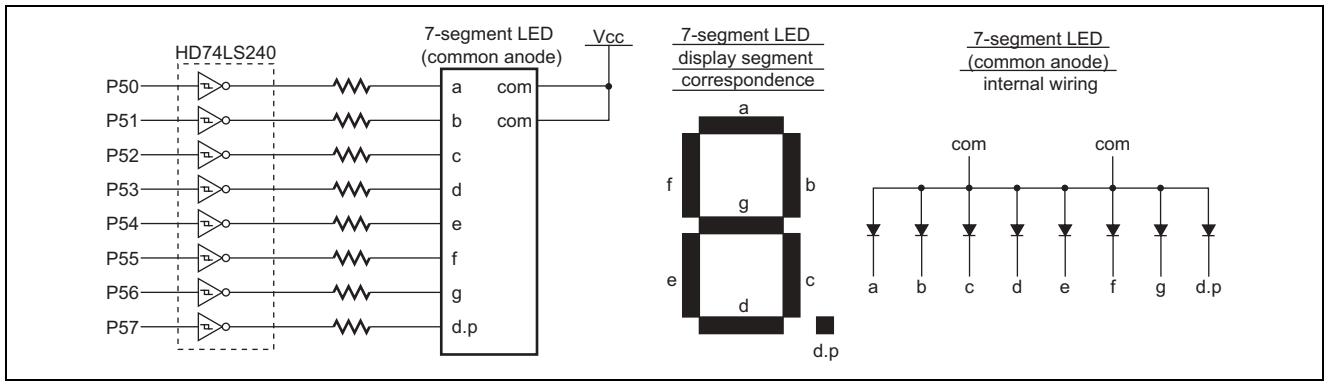
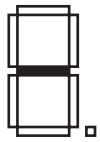


Figure 2.2 7-Segment LED Connections and Internal Wiring

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

LED Display	LED DisplayPort 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

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

3. Principle of Operation

1. Figure 3.1 shows the principle of operation in the use of timer A and A/D conversion carried out on the AN2 pin. The A/D conversion interrupt is not used in this sample task. Instead, the completion of A/D conversion is detected in the tmra interrupt processing routine.

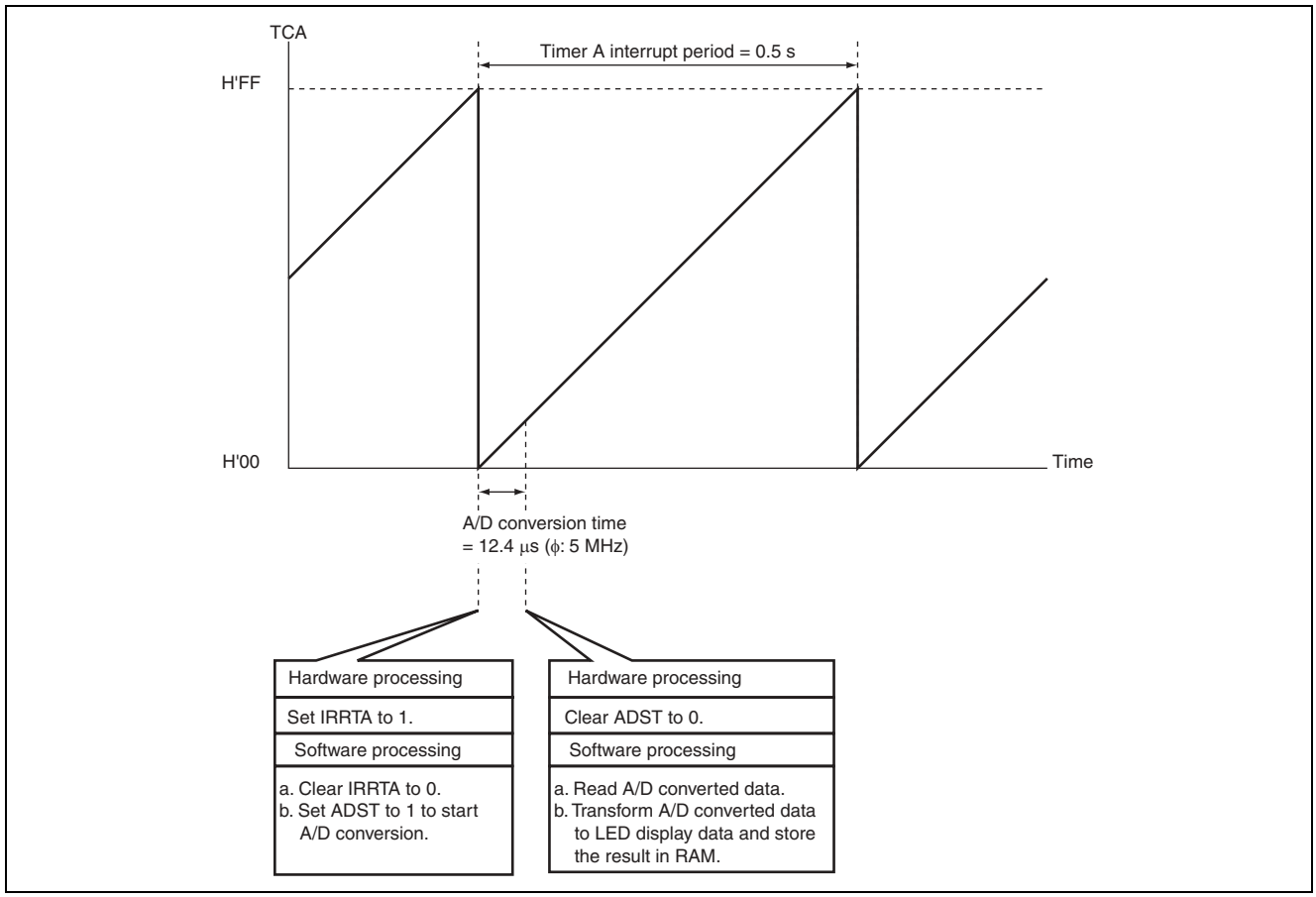


Figure 3.1 Operation Principle of A/D Conversion of AN2-Pin Signal Using Timer A

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

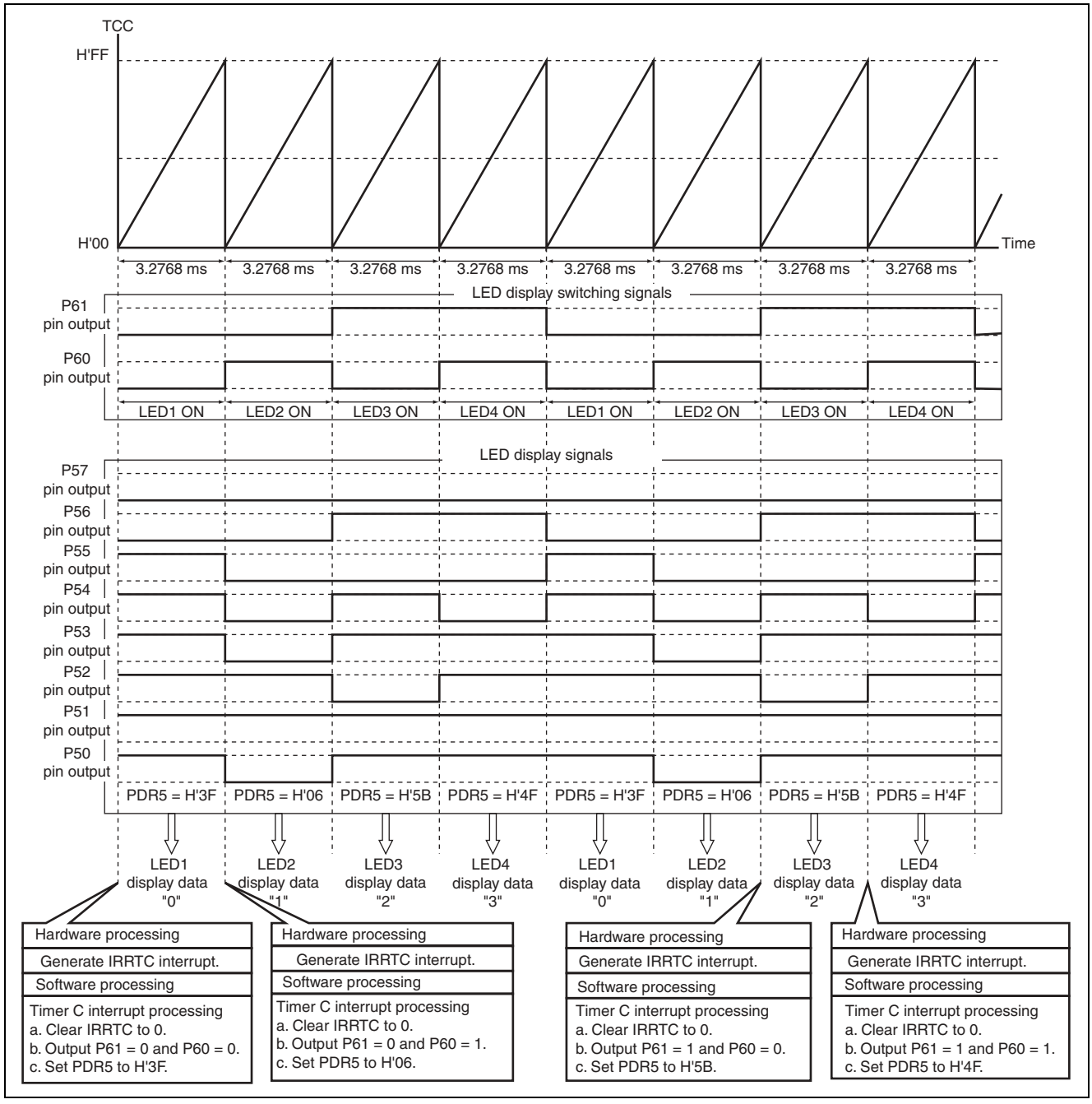


Figure 3.2 Operation Principle of 7-Segment LED Display Control

4. Description of Software

4.1 Modules

Table 4.1 describes the modules used in this sample task.

Table 4.1 Description of Modules

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

4.2 Arguments

No arguments are used in this sample task.

4.3 Internal Registers

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

Table 4.2 Description of Internal Registers

Register		Function	Address	Setting
TMA		Timer Mode Register A Sets the prescaler and input clock.	H'FFB0	H'0C (initial setting)
	TMA3	Internal Clock Select 3 Selects the operating mode for timer A. When TMA3 = 1, timer A functions as the clock time base by counting the output from prescaler W.	Bit 3	1
	TMA2	Internal Clock Select 2 to 0 When TMA3 = 1, clock time base (32.768 kHz) operation is selected. When TMA2 = 1, TMA1 = 0 and TMA0 = 0, TCA is reset. When TMA2 = 0, TMA1 = 0 and TMA0 = 1, TCA overflow period is 0.5 s.	Bit 2	0/1
	TMA 1		Bit 1	0
	TMA 0		Bit 0	0/1
TMC		Timer Mode Register C Selects the automatic reloading function, controls counting-upward/downward of the counter, and controls the input clock.	H'FFB4	H'1B
	TMC7	Automatic Reloading Select When TMC7 = 0, the interval timer function is selected.	Bit 7	0
	TMC6	Counter Upward/Downward Control When TMC6 = 0 and TMC5 = 0, TCC is an up-counter.	Bit 6	0
	TMC5		Bit 5	0
	TMC2	Clock Select When TMC2 = 0, TMC1 = 1 and TMC0 = 1, TCC counts on the internal clock ϕ/64.	Bit 2	0
	TMC1		Bit 1	1
	TMC0		Bit 0	1
TLC		Timer Load Register C Sets TCC reload value.	H'FFB5	H'00

Register		Function	Address	Setting
AMR		A/D Mode Register Sets A/D conversion speed, selects use of external trigger, and specifies analog input pin.	H'FFC6	H'36
	CKS	A/D Conversion Speed Setting When ϕ = 5 MHz: CKS = 0 selects 12.4 μs.	Bit 7	0
	TRGE	Trigger Enable When TRGE = 0, starting of A/D conversion in response to an external trigger input is disabled.	Bit 6	0
	CH3	Channel Select Bits 3 to 0 When CH3 = 0, CH2 = 1, CH1 = 1 and CH0 = 0, AN2 is selected.	Bit 3	0
	CH2		Bit 2	1
	CH1		Bit 1	1
	CH0		Bit 0	0
ADSR		A/D Start Register Sets to start or stop A/D conversion.	H'FFC7	—
	ADSF	A/D Conversion Start/Completion Check When read: ADSF = 0 indicates that A/D conversion is complete. ADSF = 1 indicates that A/D conversion is in progress. When written: Writing ADSF = 0 forcibly terminates A/D. Writing ADSF = 1 starts A/D conversion.	Bit 7	0/1
ADRRH		A/D Result Register Stores the upper 8 bits of the results of A/D conversion.	H'FFC4	—
ADRRL		A/D Result Register Stores the lower two bits of the results of A/D conversion in bits 7 and 6.	H'FFC5	—
PUCR6		Port Pull-Up Control Register 6 Provides bit-by-bit control of the MOS pull-up for the pins of port 6 that have been set as inputs. When PUCR6 = H'00, the MOS pull-up for the P67 to P60 pins are turned off.	H'FFE3	H'00
PDR6		Port Data Register 6 General-purpose I/O port data register for port 6	H'FFD9	H'00
PCR6		Port Control Register 6 Provides bit-by-bit control of input/output selection for the pins of port 6 that have been set as general-purpose I/O pins. When PCR6 = H'FF, the pins P67 to P60 function as general-purpose output pins.	H'FFE9	H'FF
PMR5		Port Mode Register 5 Sets the port 5 pin functions	H'FFCC	H'00
	WKP7	P57/$\overline{\text{WKP7}}$/SEG7 Pin Function Switching WKP7 = 0 selects the general-purpose I/O port function for P57.	Bit 7	0
	WKP6	P56/$\overline{\text{WKP6}}$/SEG6 Pin Function Switching WKP6 = 0 selects the general-purpose I/O port function for P56.	Bit 6	0

Register		Function	Address	Setting
PMR5	WKP5	P55/$\overline{\text{WKP5}}$/ADTRG Pin Function Switching WKP5 = 0 selects the general-purpose I/O port function for P55.	Bit 5	0
	WKP4	P54/$\overline{\text{WKP4}}$ Pin Function Switching WKP4 = 0 selects the general-purpose I/O port function for P54.	Bit 4	0
	WKP3	P53/$\overline{\text{WKP3}}$ Pin Function Switching WKP3 = 0 selects the general-purpose I/O port function for P53.	Bit 3	0
	WKP2	P52/$\overline{\text{WKP2}}$ Pin Function Switching WKP2 = 0 selects the general-purpose I/O port function for P52.	Bit 2	0
	WKP1	P51/$\overline{\text{WKP1}}$ Pin Function Switching WKP1 = 0 selects the general-purpose I/O port function for P51.	Bit 1	0
	WKP0	P50/$\overline{\text{WKP0}}$ Pin Function Switching WKP0 = 0 selects the general-purpose I/O port function for P50.	Bit 0	0
PUCR5		Port Pull-Up Control Register 5 Provides bit-by-bit control of the MOS pull-up for the pins of port 5 that have been set as inputs. When PUCR5 = H'00, the MOS pull-up for the P57 to P50 pins are turned off.	H'FFE2	H'00
PDR5		Port Data Register 5 General-purpose I/O port data register for port 5	H'FFD8	H'00
PCR5		Port Control Register 5 Provides bit-by-bit control of input/output selection for the pins of port 5 that have been set as general-purpose I/O pins. When PCR5 = H'FF, the pins P57 to P50 function as general-purpose output pins.	H'FFE8	H'FF
IENR1		Interrupt Enable Register 1 Enables/disables interrupt requests.	H'FFF3	—
	IENTA	Timer A Interrupt Request Enable When IENTA = 1, timer A overflow interrupt requests are enabled.	Bit 5	1
IRR1		Interrupt Request Register 1 If an interrupt request is generated by timer A, IRQ4, IRQ3, IRQAEC, IRQ1 or IRQ0, the corresponding flag is set to 1.	H'FFF6	—
	IRRTA	Timer A Interrupt Request Flag This is set to 1 when the timer A counter has overflowed (H'FF → H'00). This is cleared to 0 when 0 is written to.	Bit 7	0/1
IENR2		Interrupt Enable Register 2 Enables/disables interrupt requests.	H'FFF4	—
	IENTC	Timer C Interrupt Request Enable: When IENTC = 1, timer A overflow/underflow interrupt requests are enabled.	Bit 1	1

Register		Function	Address	Setting
IRR2		Interrupt Request Register 2 If an interrupt request is generated by a direct transition, A/D converter, timer G, timer FH, timer FL, timer C or asynchronous event counter, the corresponding flag is set to 1.	H'FFF7	—
	IRRTC	Timer C Interrupt Request Flag This is set to 1 when the timer C counter has overflowed (H'FF → H'00) or underflowed (H'00 → H'FF). This is cleared to 0 when 0 is written to.	Bit 7	0/1

4.4 Description of RAM

Table 4.3 describes the RAM used in this sample task.

Table 4.3 Description of RAM

Label	Function	Address	Used in
dig_0	Stores LED1 display data. (1 byte)	H'FB80	main, tmra
dig_1	Stores LED2 display data. (1 byte)	H'FB81	main, tmra
dig_2	Stores LED3 display data. (1 byte)	H'FB82	main, tmra
dig_3	Stores LED4 display data. (1 byte)	H'FB83	main, tmra
cnt	8-bit counter used in switching display on LED1 to LED4. (1 byte)	H'FB84	main, tmrc

4.5 Description of Data Table

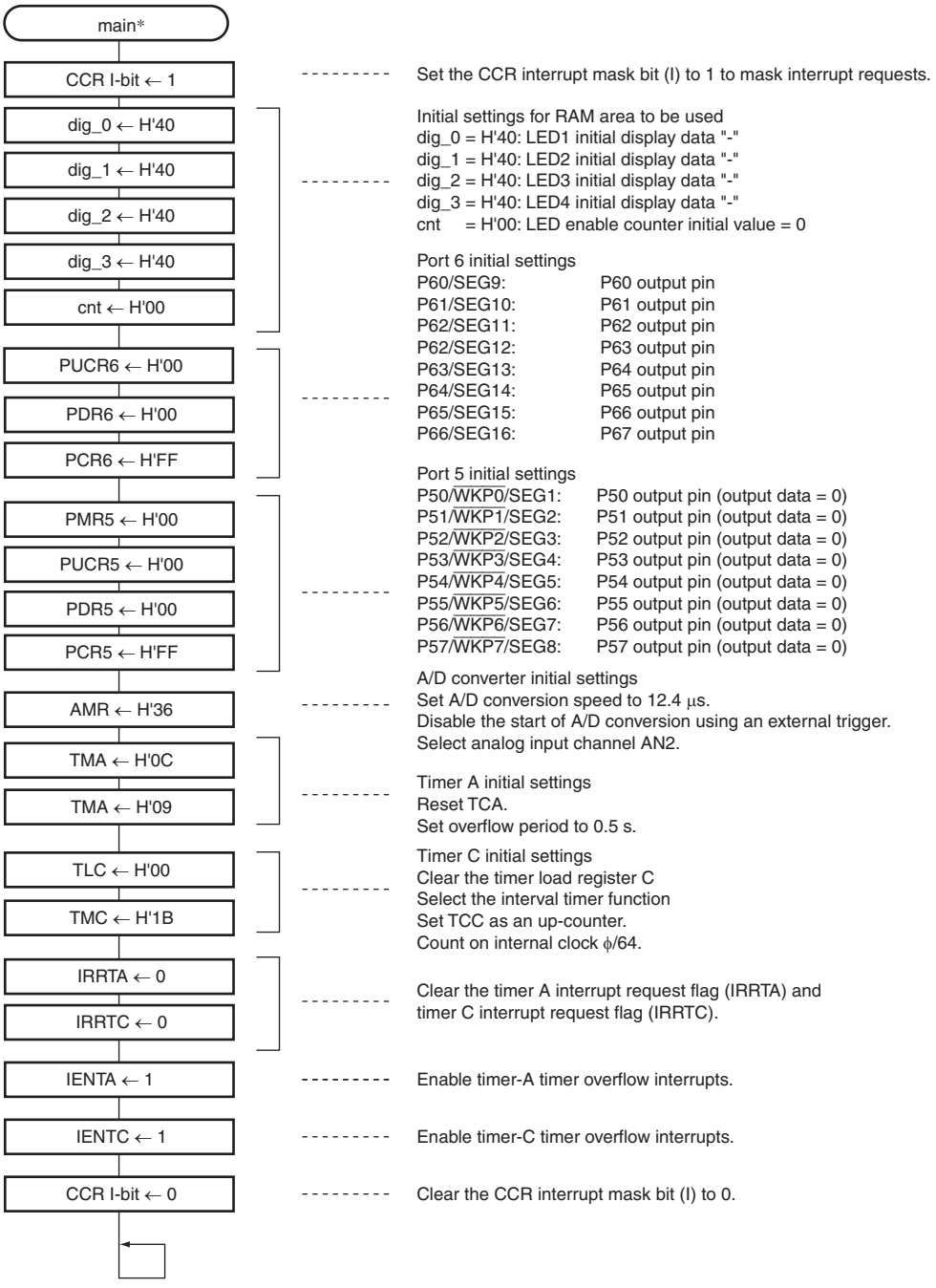
In this sample task, display data for the 7-segment LED displays are stored in the ROM as a 1-dimensional array (data table). Table 4.4 describes the table of display data (dsp_data [ ]).

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

Array Name	Data	Data Description	Data Size	Address
dsp_data[0]	H'3F	Data output from port 5 to display "0"	1 byte	H'01EC
dsp_data[1]	H'06	Data output from port 5 to display "1"	1 byte	H'01ED
dsp_data[2]	H'5B	Data output from port 5 to display "2"	1 byte	H'01EE
dsp_data[3]	H'4F	Data output from port 5 to display "3"	1 byte	H'01EF
dsp_data[4]	H'66	Data output from port 5 to display "4"	1 byte	H'01F0
dsp_data[5]	H'6D	Data output from port 5 to display "5"	1 byte	H'01F1
dsp_data[6]	H'7D	Data output from port 5 to display "6"	1 byte	H'01F2
dsp_data[7]	H'27	Data output from port 5 to display "7"	1 byte	H'01F3
dsp_data[8]	H'7F	Data output from port 5 to display "8"	1 byte	H'01F4
dsp_data[9]	H'6F	Data output from port 5 to display "9"	1 byte	H'01F5
dsp_data[10]	H'77	Data output from port 5 to display "A"	1 byte	H'01F6
dsp_data[11]	H'7C	Data output from port 5 to display "b"	1 byte	H'01F7
dsp_data[12]	H'39	Data output from port 5 to display "C"	1 byte	H'01F8
dsp_data[13]	H'5E	Data output from port 5 to display "d"	1 byte	H'01F9
dsp_data[14]	H'79	Data output from port 5 to display "E"	1 byte	H'01FA
dsp_data[15]	H'71	Data output from port 5 to display "F"	1 byte	H'01FB

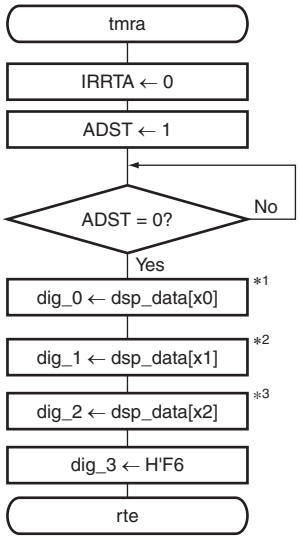
5. Flowchart

1. Main routine (main)



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

2. Timer A interrupt processing routine (tmra)



Clear the timer-A interrupt request flag (IRRTA).

Set the A/D start bit (ADST) to 1 to start A/D conversion on AN2.

A/D conversion completed?

Shift ADRR 6 bits to the right then perform AND operation with H'000F, refer to dsp_data to read data corresponding to the ANDed value, and store the data in dig_0.

Shift ADRR 10 bits to the right then perform AND operation with H'000F, refer to dsp_data to read data corresponding to the ANDed value, and store the data in dig_1.

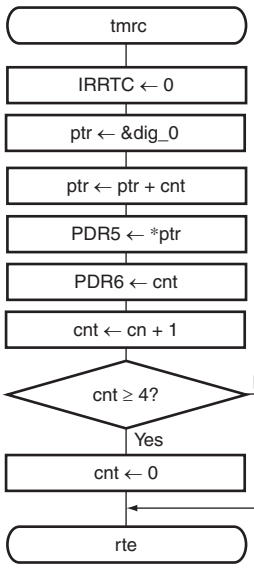
Shift ADDR 14 bits to the right then perform AND operation with H'0003, refer to dsp_data to read data corresponding to the ANDed value, and store the data in dig_2.

Store "H" display data (H'F6) in dig_3.

Notes:
1. x0 = ADRR >> 6 & H'000F
2. x1 = ADRR >> 10 & H'000F
3. x2 = ADRR >> 14 & H'0003

ADRR is a name used to treat ADRRH and ADRRL together as a 16-bit data.

3. Timer C interrupt processing routine (tmrc)



Clear the timer-C interrupt request flag (IRRTC).

Store dig_0 address in ptr.

Store dig_0 address + cnt value in ptr.

Store contents of the address indicated by ptr in PDR5.

Store cnt value in PDR6.

Increment cnt.

cnt ≥ 4?

Initialize cnt.

6. Program Listing

INIT.SRC (Program listing)

```
    .EXPORT   _INIT
    .IMPORT   _main
;
    .SECTION  P,CODE
_INIT:
    MOV.W     #H'FF80,R7
    LDC.B     #B'10000000,CCR
    JMP       @_main
;
    .END
```

```
/************************************************************************************************************************/
/*                                                                                                                      */
/*  H8/300L Super Low Power Series                                                                                      */
/*      -H8/38024 Series-                                                                                               */
/*  Application Note                                                                                                    */
/*  ' Application example '                                                                                             */
/*  ' Connecting a thermistor '                                                                                         */
/*                                                                                                                      */
/*  Function                                                                                                            */
/*  : Connecting a thermistor                                                                                           */
/*                                                                                                                      */
/*  External Clock : 10MHz                                                                                              */
/*  Internal Clock : 5MHz                                                                                               */
/*  Sub Clock      : 32.768kHz                                                                                          */
/*                                                                                                                      */
/************************************************************************************************************************/

#include <machine.h>

/************************************************************************************************************************/
/*  Symbol Defnition                                                                                                    */
/************************************************************************************************************************/
struct BIT {
    unsigned char   b7:1;       /* bit7 */
    unsigned char   b6:1;       /* bit6 */
    unsigned char   b5:1;       /* bit5 */
    unsigned char   b4:1;       /* bit4 */
    unsigned char   b3:1;       /* bit3 */
    unsigned char   b2:1;       /* bit2 */
    unsigned char   b1:1;       /* bit1 */
    unsigned char   b0:1;       /* bit0 */
};

#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    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    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[16] =
{
    0x3f,                                                   /* LED display data = "0"                  */
    0x06,                                                   /* LED display data = "1"                  */
    0x5b,                                                   /* LED display data = "2"                  */
    0x4f,                                                   /* LED display data = "3"                  */
    0x66,                                                   /* LED display data = "4"                  */
    0x6d,                                                   /* LED display data = "5"                  */
    0x7d,                                                   /* LED display data = "6"                  */
    0x27,                                                   /* LED display data = "7"                  */
    0x7f,                                                   /* LED display data = "8"                  */
    0x6f,                                                   /* LED display data = "9"                  */
    0x77,                                                   /* LED display data = "A"                  */
    0x7c,                                                   /* LED display data = "B"                  */
    0x39,                                                   /* LED display data = "C"                  */
    0x5e,                                                   /* LED display data = "D"                  */
    0x79,                                                   /* LED display data = "E"                  */
    0x71                                                    /* LED display data = "F"                  */
};

/*****************************************************************************************************************/
/*  RAM Define                                                                                                   */
/*****************************************************************************************************************/
unsigned char dig_0;                                        /* Dig-0 LED display data store            */
unsigned char dig_1;                                        /* Dig-1 LED display data store            */
unsigned char dig_2;                                        /* Dig-2 LED display data store            */
unsigned char dig_3;                                        /* Dig-3 LED display data store            */
unsigned char cnt;                                          /* LED enable counter                      */

/*****************************************************************************************************************/
/*  Vector Address                                                                                               */
/*****************************************************************************************************************/
#pragma section V1                                          /* Vector section set                      */
void (*const VEC_TBL1[])(void) = {
    INIT                                                    /* 0x0000 Reset vector                     */
};
#pragma section V2                                          /* Vector section set                      */
void (*const VEC_TBL2[])(void) = {
    tmra                                                    /* 0x0016 Timer A interrupt vector         */
};
#pragma section V3                                          /* Vector section set                      */
void (*const VEC_TBL3[])(void) = {
    tmrc                                                    /* 0x001A 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             */

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

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

    AMR   = 0x36;                                       /* A/D converter initialize (AN2)       */

    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 ?                  */
    dig_0 = dsp_data[ADRR >>  6 & 0x000f];              /* Dig-0 LED display data set           */
    dig_1 = dsp_data[ADRR >> 10 & 0x000f];              /* Dig-1 LED display data set           */
    dig_2 = dsp_data[ADRR >> 14 & 0x0003];              /* Dig-2 LED display data set           */
    dig_3 = 0xf6;                                       /* Dig-3 LED display data set           */
}
```

```
/*****************************************************************************************************************/
/*  Timer C Interrupt                                                                                            */
/*****************************************************************************************************************/
void tmrc(void)
{
    unsigned char *ptr;                                 /* Pointer set                          */

    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                     */
    }
}
```

Link address specifications

Section Name	Address
CV1	H'0000
CV2	H'0016
CV3	H'001A
P	H'0100
B	H'FB80

Revision Record

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