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

H8/300H Tiny Series

Ohm Meter Implementation Example

Introduction

In this example an ohm meter is implemented using an inverting op-amp. A reference resistor and the A/D converter built into the H8/36014 are used to measure the resistance value.

Target Device

H8/300H Tiny Series H8/36014

Contents

1. Specifications

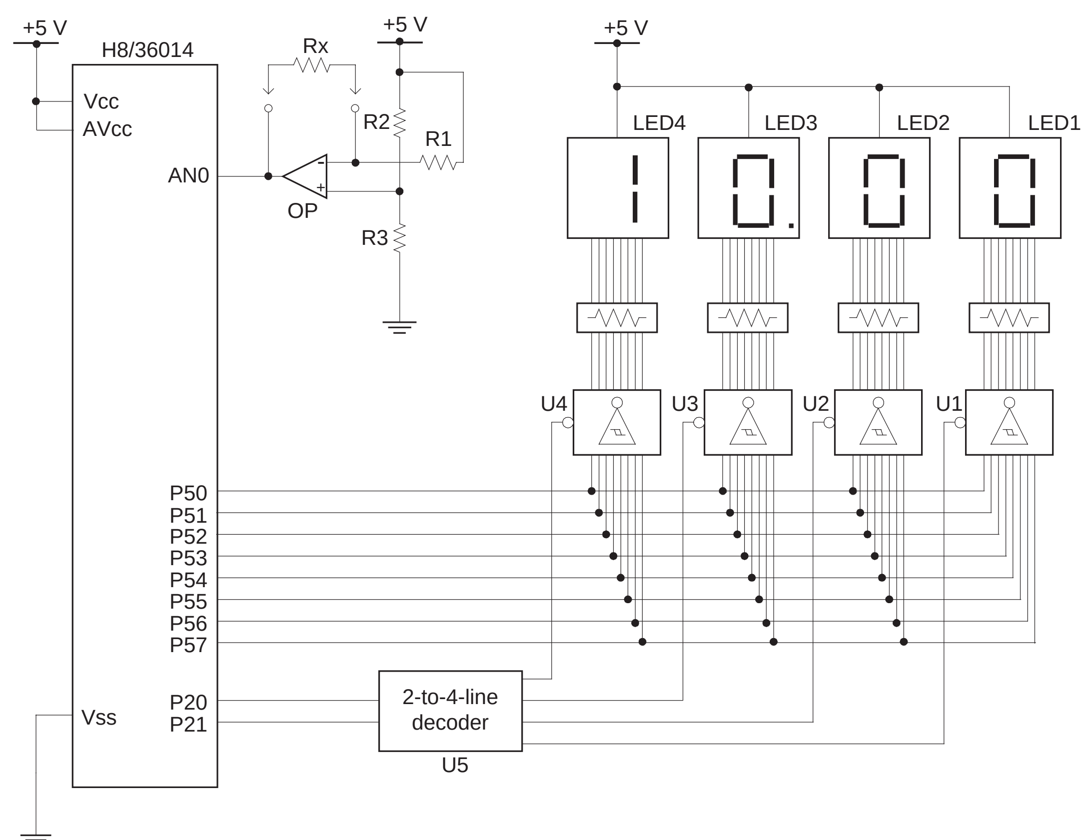
(1) The hardware configuration used to measure resistance values is shown in figure 1.

(2) In this sample task the ohm meter comprises the reference resistor R1, an inverting op-amp, and the A/D converter built into the H8/36014.

(3) The output voltage from the op-amp is input to analog input pin 0 (the AN0 pin) of the A/D converter, which performs A/D conversion.

(4) The resistance value is obtained from the A/D conversion result by calculation and displayed as a decimal number on a 7-segment LED connected to the I/O ports.

(5) In this sample task the operating voltage (Vcc) and analog power supply voltage (AVcc) of the H8/36014 are 5 V, and the OSC clock frequency is 20 MHz, supplied as an external clock by a crystal oscillator.



Legend:

OP: Op-amp (TLV2464CN, TI)

R1 to R3: Resistors (note: R1 = 10 kΩ, R2 = 10 kΩ, R3 = 10 kΩ)

LED1 to LED4: Anode common 7-segment LEDs

U1 to U4: 3-state output inverter drivers (HD74LS240)

U5: 2-to-4-line decoder (HD74HC139)

Figure 1 Hardware Configuration

(6) In this sample task the R1 resistance value has been determined so as to allow accurate measurement of resistance values up to 10 kΩ. The design procedure is described below.

(a) C-MOS rail-to-rail (some manufacturers use the term “full swing”) operational amplifier is a type of op-amp used as an inverting amplifier. It can produce an output amplitude up to the power supply voltage. Due to the relationship of the op-amp and inverting amplifier circuitry, the following relationship exists.

$$Es/R1 = –Ex/Rx$$

(Es: reference voltage, R1: reference resistance, Ex: op-amp output voltage,
Rx: resistance being measured)

(b) Based on this relationship, the following equation can be used to calculate the resistance being measured.

$$Rx = –R1/Es \times Ex$$

(c) To simplify the circuit configuration a common power supply voltage is used in this sample task for the op-amp and the H8/36014. Therefore, bias voltage Eb is applied to the non-inverted input pin (+) so that a signal is output from the negative side. Taking bias voltage Eb into account, the equation for calculating Rx changes to the following.

$$Rx = –R1/(Es – Eb) \times (Ex – Eb)$$

(d) To obtain a measuring range up to 10 kΩ a value of 10 kΩ is assigned for reference resistor R1, 5 V for reference voltage Es, and 2.5 V for bias voltage Eb, resulting in the following.

$$\begin{aligned} Rx &= –10000/(5 – 2.5) \times (Ex – 2.5) \\ &= 10000 – 4000 \cdot Ex \end{aligned}$$

(Ex: op-amp output voltage)

(e) The measuring range can be changed easily by changing reference resistor R1.

Example: Increasing R1 10 times or 100 times also increases the Rx measuring range 10 times or 100 times, respectively.

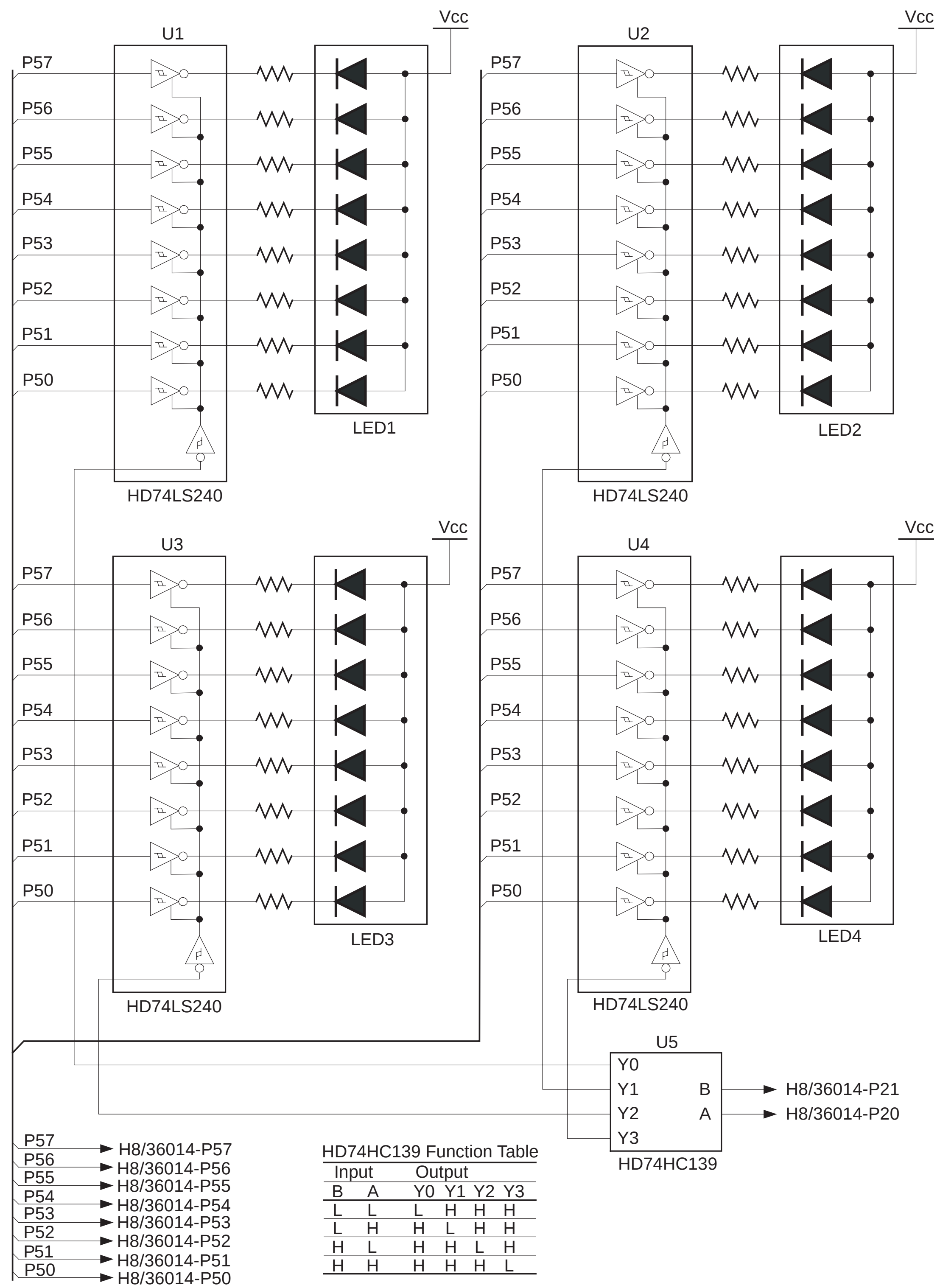
(7) The operations in this sample task are as follows.

(a) Connect the resistors to be measured (using probes, or the like) to Rx as shown in figure 1.

(b) When, for example, 10 kΩ (unknown) resistors are connected, the op-amp output will be 0 V due to the relationship between the inverting amplifier and the bias voltage. This output voltage is input to analog input pin 0 (the AN0 pin) of the A/D converter.

(c) Using the equation shown in step (d) above, the resistance being measured (Rx) is calculated as 10 kΩ.

(d) The resistance value calculated from the measurement results from the A/D converter is displayed using the 7-segment LEDs. A value of 10 kΩ is displayed as the decimal number 10.00.



HD74HC139 Function Table

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 2 7-Segment LED Control Method

(8) In this sample task the A/D conversion result is displayed using the 7-segment LEDs as a decimal number (00.00–99.99). The LED display for the A/D conversion result is shown in figure 3.

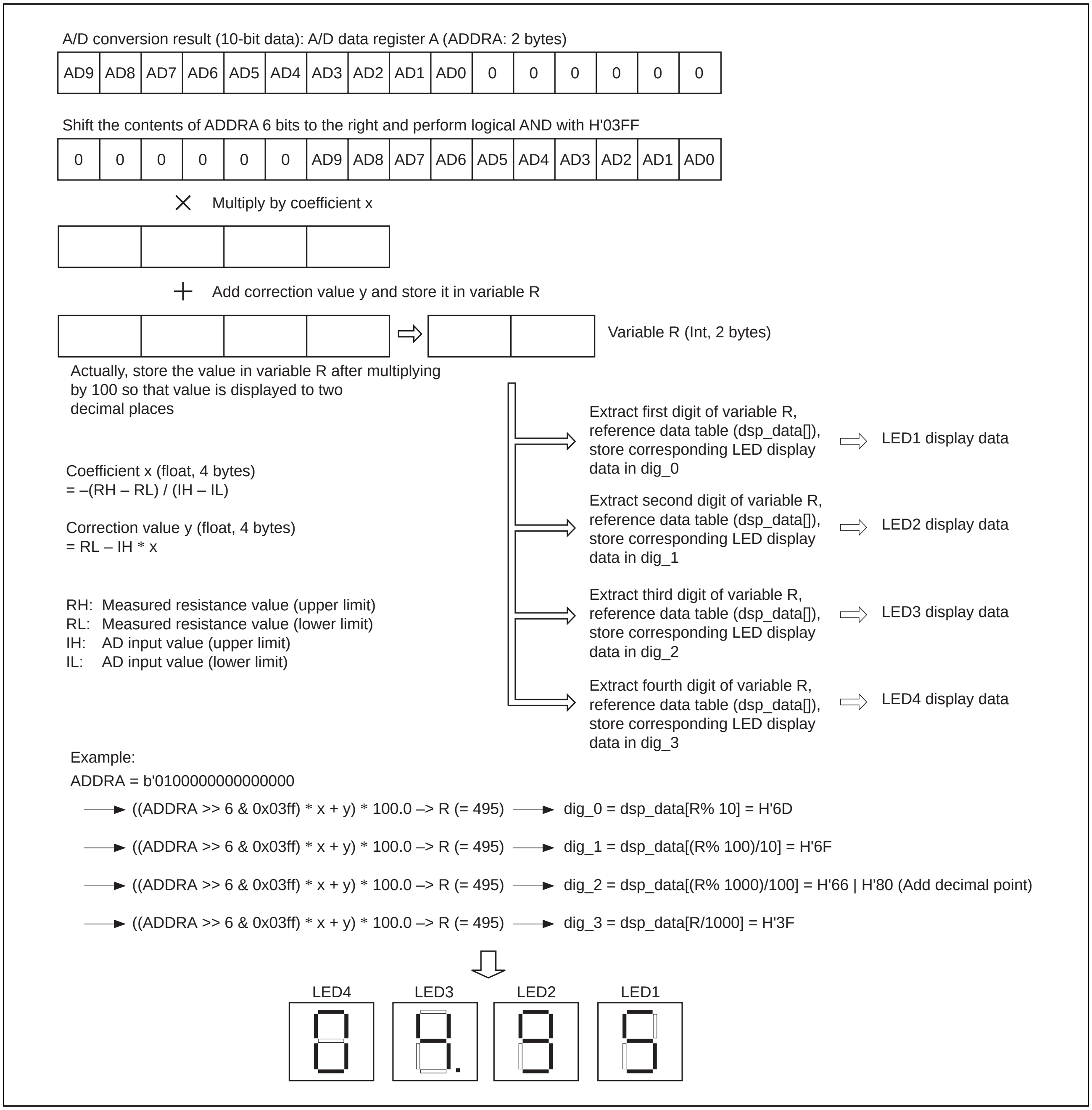


Figure 3 LED Display for A/D Conversion Result

2. Functions Used

(1) A block diagram of the H8/36014 functions used in this sample task is shown in figure 4, and the assignment of functions is shown in table 1.

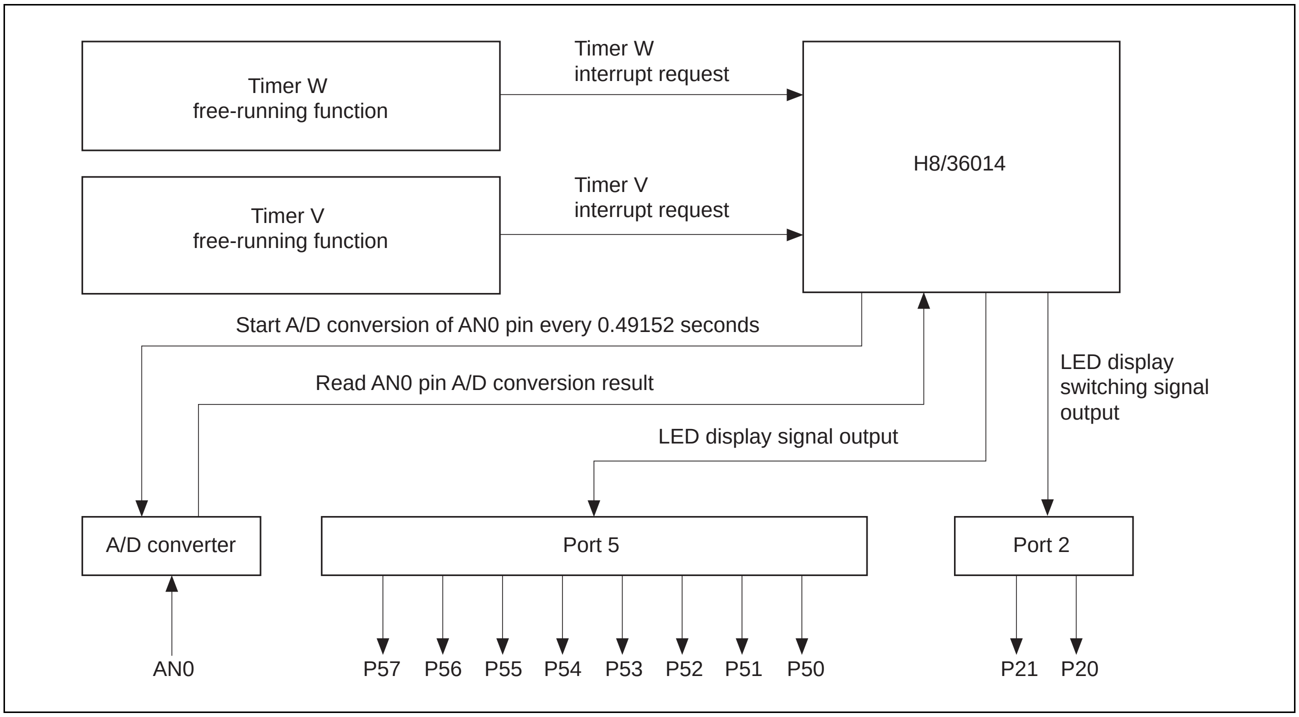


Figure 4 Block Diagram of Functions Used

Table 1 Assignment of Functions

Elements	Description
Timer W	The free-running function of timer W is used to perform A/D conversion of the input to analog input pin 0 (AN0). The A/D conversion cycle is measured using the timer W overflow cycle of 32.768 ms.
Timer V	The free-running function of timer V is used to control switching of the 7-segment LEDs. Dynamic display is implemented by illuminating the four 7-segment LEDs sequentially each timer V overflow cycle of 2.048 ms.
A/D converter	The A/D converter performs A/D conversion of the output voltage of the inverting op-amp, and the resistance value is calculated from the detected voltage value.
Port 2	Switching of the four 7-segment LEDs is performed based on the output from port 2 output pins P20 and P21. Output pins P20 and P21 are connected to the input pins of the 2-to-4-line decoder.
Port 5	The display of the 7-segment LEDs is based on the output from port 5 output pins P50 to P57. The 10-bit data obtained by A/D conversion of the input to AN0 is converted to 3-digit hexadecimal data and output to the LEDs.

(2) A connection diagram for the 7-segment LEDs used is shown in figure 5. As shown in figure 5, a high-level signal output from port 5 causes the corresponding LED segment to light. Table 2 shows the correspondence between port 5 output and 7-segment LED display data.

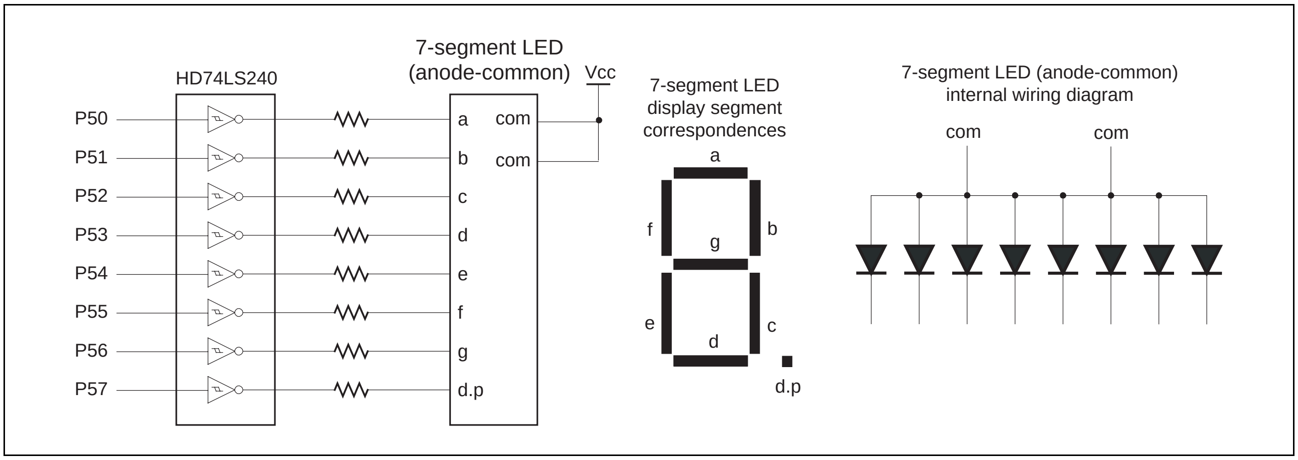
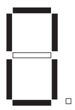
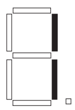
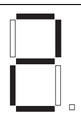
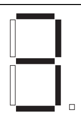
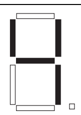
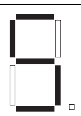
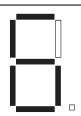
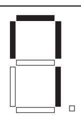
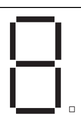
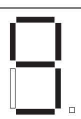
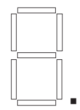


Figure 5 7-Segment LED Connection Diagram and Internal Wiring Diagram

Table 2 Correspondence Between Port 5 Output and 7-Segment LED Display 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

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

3. Principles of Operation

(1) The principles of operation whereby A/D conversion of the input to the AN0 pin is performed using timer W are illustrated in figure 6. As shown in figure 6, in this sample task measurement of the A/D conversion cycle (0.49152 seconds) using the timer W overflow flag and determination of the end of A/D conversion in the tmrw routine are accomplished without using A/D converter interrupts.

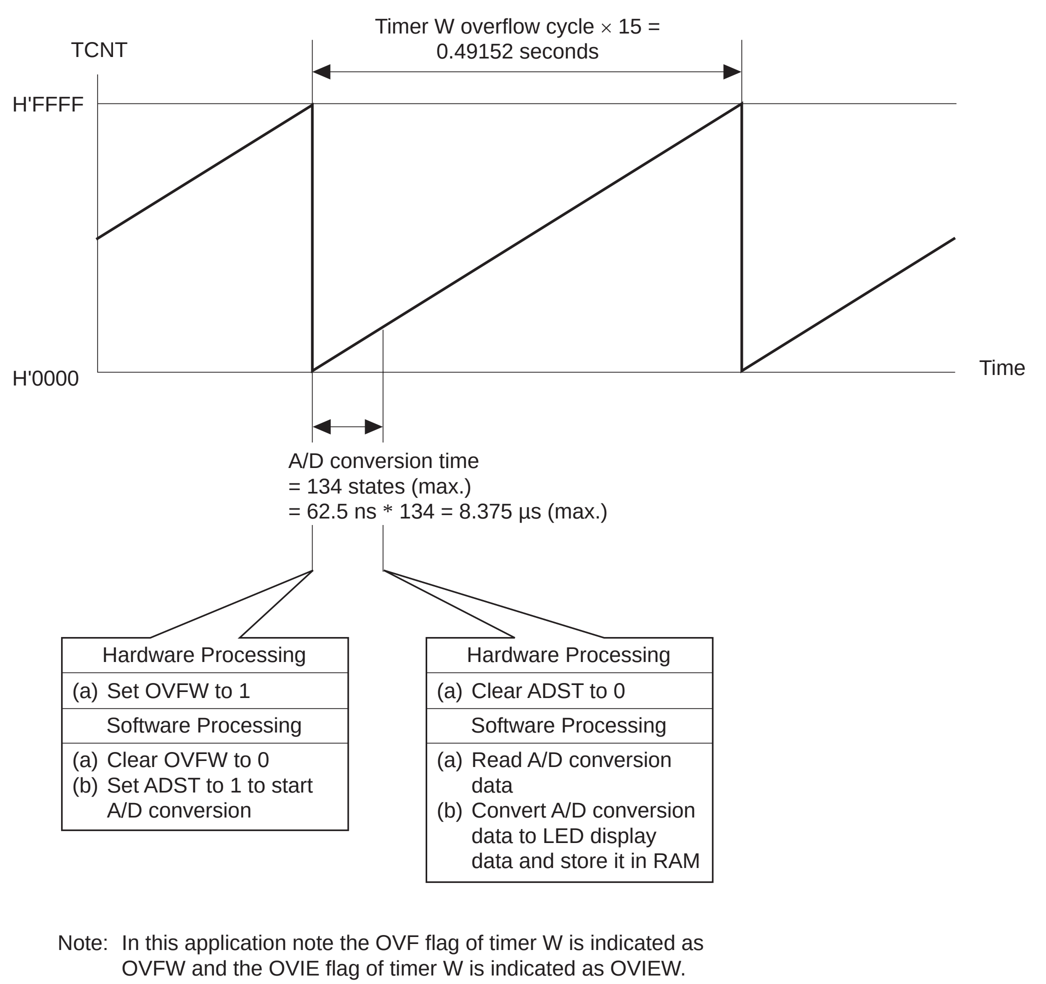


Figure 6 Operating Principle of Remote Control Reception Using Timer W

(2) The operation principles of 7-segment LED display control are shown below. Displaying “3210” using LED4 to LED1 is illustrated in figure 7. As shown in figure 7, dynamic 7-segment LED display is implemented by illuminating LED4 to LED1 sequentially each timer V overflow cycle.

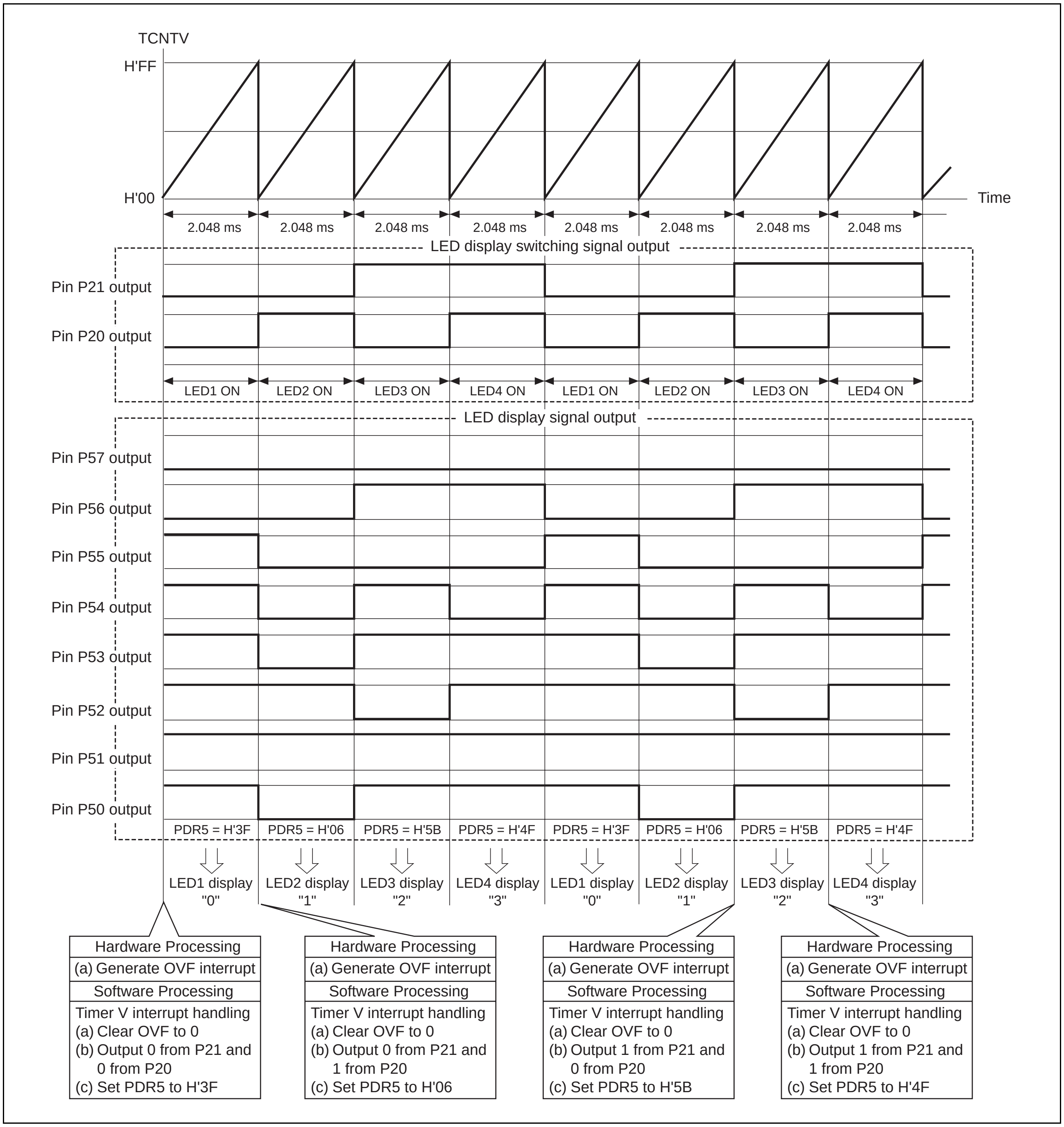


Figure 7 Operating Principle of 7-Segment LED Display Control

4. Description of Software

4.1 Modules

Table 3 shows the modules used in this sample task.

Table 3 Modules

Module Name	Label	Description
Main routine	main	Performs initial settings, enables interrupts
Timer W interrupt handling routine	tmrw	Clears interrupt flag, converts A/D conversion data to LED display data and stores it in RAM
Timer V interrupt handling routine	tmrv	Clears interrupt flag, controls LED display data output and LED display switching

4.2 Arguments

No arguments are used in this sample task.

4.3 Internal Registers Used

The internal registers used in this sample task are shown in table 4.

Table 4 Internal Registers Used

Register		Description	Address	Set Value
TCRV0		Timer control register V0 Selects the input clock signals of TCNTV, specifies the clearing conditions of TCNTV, and controls each interrupt request.	H'FFA0	H'03 (initial setting)
	CMIEB	Compare match interrupt enable B When this bit is cleared to 0, interrupt requests indicated by the CMFB bit in TCSRV are disabled.	Bit 7	0
	CMIEA	Compare match interrupt enable A When this bit is set to 0, interrupt requests indicated by the CMFA bit in TCSRV are disabled.	Bit 6	0
	OVIE	Timer overflow interrupt enable When this bit is cleared to 0, interrupt requests indicated by the OVF bit in TCSRV are disabled. When this bit is set to 1, interrupt requests indicated by the OVF bit in TCSRV are enabled.	Bit 5	0/1

Register		Description	Address	Set Value
TCRV0	CCLR1	Counter clear 1 and 0	Bit 4	0
	CCLR0	These bits specify the clearing conditions of TCNTV. Clearing is disabled when CCLR1 = 0 and CCLR0 = 0.	Bit 3	0
	CKS2	Clock select 2 to 0	Bit 2	0
	CKS1	These bits select the clock signals input to TCNTV and the counting condition in combination with ICKS0 in TCRV1. When CKS2 = 0, CKS1 = 1, CKS0 = 1, and ICKS0 = 1, TCNTV counts on internal clock ϕ/128, falling edge.	Bit 1	1
	CKS0		Bit 0	1
TCSRV		Timer control/status register V Indicates the status flag and controls output using compare match.	H'FFA1	H'10
	CMFB	Compare match flag B This bit is set to 1 when the TCNTV and TCORB values match.	Bit 7	0
	CMFA	Compare match flag A This bit is set to 1 when the TCNTV and TCORA values match.	Bit 6	0
	OVF	Timer overflow flag This bit is set to 1 when the TCNTV value overflows. It is cleared to 0 when OVF is read as 1 and then cleared by writing 0 to OVF.	Bit 5	0
	OS3	Output select 3 and 2	Bit 3	0
	OS2	These bits set the output level of the TMOV pin by compare match B. No change when OS3 = 0 and OS2 = 0.	Bit 2	0
	OS1	Output select 3 and 2	Bit 1	0
	OS0	These bits set the output level of the TMOV pin by compare match A. No change when OS1 = 0 and OS0 = 0.	Bit 0	0
TCRV1		Timer control register V1 Selects the edge at the TRGV pin, enables TRGV input, and selects the clock input to TCNTV.	H'FFA5	H'E3
	TVEG1	TRGV input edge select 1 and 0	Bit 4	0
	TVEG0	These bits select the TRGV input edge. TRGV trigger input is disabled when TREG1 = 0 and TREG0 = 0.	Bit 3	0
	TRGE	TRGV input enable Enables or disables TCNTV counting-up by the input of the TRGV pin. When TREG = 0, starting counting-up TCNTV by the input of the TRGV pin, and halting counting-up TCNTV when TCNTV is cleared by a compare match, are disabled.	Bit 2	0

Register		Description	Address	Set Value
TCRV1	ICKS0	Internal clock select 0 This bit selects the clock signals input to TCNTV and the counting condition in combination with CKS2 to CKS0 in TCRV0. When CKS2 = 0, CKS1 = 1, CKS0 = 1, and ICKS0 = 1, TCNTV counts on internal clock ϕ/128, falling edge.	Bit 0	1
TMRW		Timer mode register W Selects the general register functions and the timer output mode.	H'FF80	H'C8
	CTS	Counter start When CTS = 1, counter operation starts. When CTS = 0, counter operation stops.	Bit 7	1
TCRW		Timer control register W Selects the counter clock source, selects a clearing condition, and specifies the timer output levels.	H'FF81	H'30
	CKS2	Clock select	Bit 6	0
	CKS1	When CKS2 = 0, CKS1 = 1, and CKS0 = 1, the TCNT input clock is set as 1/8 the system clock.	Bit 5	1
	CKS0		Bit 4	1
TIERW		Timer interrupt enable register W Controls timer W interrupt requests.	H'FF82	H'00 (initial setting)
	OVIE	Timer overflow interrupt enable When OVIE = 0, interrupt requests by the OVF flag are disabled. When OVIE = 1, interrupt requests by the OVF flag are enabled.	Bit 7	0/1
TSRW		Timer status register W Shows the status of interrupt requests.	H'FF83	H'00
	OVF	Timer overflow OVF = 0 when TCNT has not overflowed. OVF = 1 when TCNT overflows.	Bit 7	0
TCNT		Timer counter 16-bit up-counter that accepts a source 1/8 the system clock as input.	H'FF86	H'00
ADCSR		A/D control/status register Consists of the control bits and conversion status bits of the A/D converter.	H'FFB8	H'00 (initial setting)
	ADF	A/D end flag Set to 1 when A/D conversion ends in single mode. Cleared to 0 when 0 is written after reading ADF = 1.	Bit 7	0
	ADIE	A/D interrupt enable When this bit is cleared to 0, A/D conversion end interrupt requests are disabled.	Bit 6	0

Register		Description	Address	Set Value
ADCSR	ADST	A/D start Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when A/D conversion is complete.	Bit 5	0/1
	SCAN	Scan mode When this bit is cleared to 0, the single mode is selected for A/D conversion.	Bit 4	0
	CKS	Clock select When this bit is cleared to 0, the A/D conversion time = 134 states (max.).	Bit 3	0
	CH2	Channel select	Bit 2	0
	CH1	Selects the analog input channels.	Bit 1	0
	CH0	AN0 is selected when CH2 = 0, CH1 = 0, and CH0 = 0 (when SCAN = 0).	Bit 0	0
ADCR		A/D control register Enables A/D conversion started by an external trigger signal.	H'FFB9	H'7E
	TRGE	Trigger enable When this bit is cleared to 0, A/D conversion starting at the falling edge or the rising edge of the external trigger signal ($\overline{\text{ADTRG}}$) is disabled.	Bit 7	0
ADDRC		A/D data register C Stores A/D conversion results from AN0.	H'FFB4	—
PCR2		Port control register 2 Selects input or output status in bit units for pins to be used as general I/O ports of port 2. When PCR2 = H'03, pin P22 functions as a general input pin and pins P21 and P20 function as general output pins.	H'FFE5	H'03
PDR2		Port data register 2 General I/O port data register of port 2.	H'FFD5	H'F8
PMR5		Port mode register 5 Sets the functions of pins in port 5.	H'FFE1	H'00
	POF57	P57 pin function switch When this bit is cleared to 0, P57 functions as a general I/O port.	Bit 7	0
	POF56	P56 pin function switch When this bit is cleared to 0, P56 functions as a general I/O port.	Bit 6	0
	WKP5	P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$ pin function switch When this bit is cleared to 0, P55 functions as a general I/O port.	Bit 5	0
	WKP4	P54/$\overline{\text{WKP4}}$ pin function switch When this bit is cleared to 0, P54 functions as a general I/O port.	Bit 4	0

Register		Description	Address	Set Value
PMR5	WKP3	P53/$\overline{\text{WKP3}}$ pin function switch When this bit is cleared to 0, P53 functions as a general I/O port.	Bit 3	0
	WKP2	P52/$\overline{\text{WKP2}}$ pin function switch When this bit is cleared to 0, P52 functions as a general I/O port.	Bit 2	0
	WKP1	P51/$\overline{\text{WKP1}}$ pin function switch When this bit is cleared to 0, P51 functions as a general I/O port.	Bit 1	0
	WKP0	P50/$\overline{\text{WKP0}}$ pin function switch When this bit is cleared to 0, P50 functions as a general I/O port.	Bit 0	0
PUCR5		Port pull-up control register 5 Controls the pull-up MOS in bit units of the port 5 pins set as the input ports. When PUCR2 = H'00, pull-up MOS is off for pins P57 to P50.	H'FFD1	H'00
PDR5		Port data register 5 General I/O port data register of port 5.	H'FFD8	H'00
PCR5		Port control register 5 Selects input or output status in bit units for pins to be used as general I/O ports of port 5. When PCR5 = H'FF, pins P57 to P50 function as general output pins.	H'FFE8	H'FF

4.4 RAM Usage

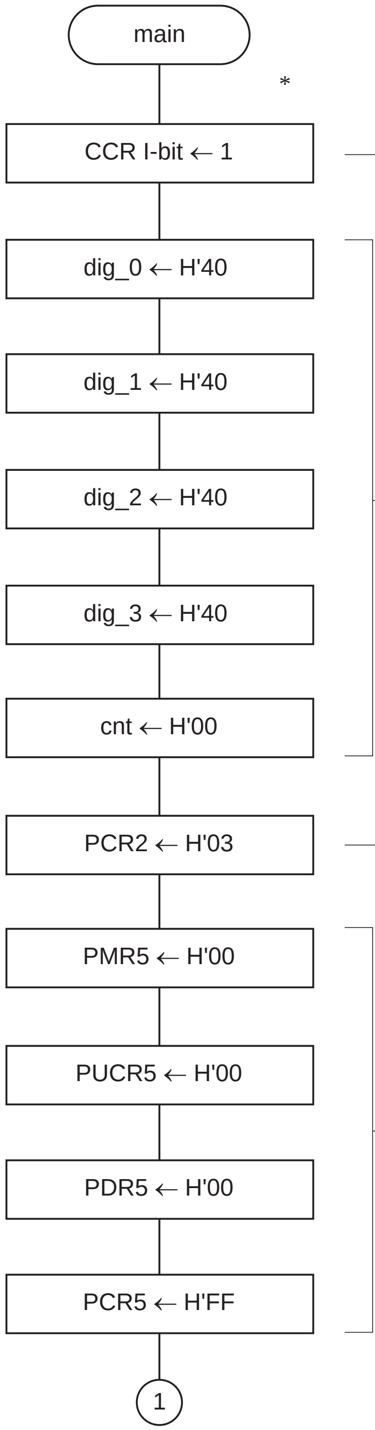
The RAM usage in this sample task is shown in table 5.

Table 5 RAM Usage

Label	Description	Address	Used in
dig_0	Stores LED1 display data (1 byte)	H'FB8A	main, tmrw
dig_1	Stores LED2 display data (1 byte)	H'FB8B	main, tmrw
dig_3	Stores LED3 display data (1 byte)	H'FB8C	main, tmrw
dig_4	Stores LED4 display data (1 byte)	H'FB8D	main, tmrw
cnt	8-bit counter for display switching of LED1 to LED4 (1 byte)	H'FB8E	main, tmrv
counter_sub	8-bit counter for adjusting the A/D acquisition interval (1 byte)	H'FB8F	main, tmrw
R	For displaying the resistance value (2 bytes)	H'FB80	tmrw
x	Coefficient (4 bytes)	H'FB82	main, tmrw
y	Correction value (4 bytes)	H'FB86	main, tmrw

5. Flowcharts

5.1 Main Routine (main)



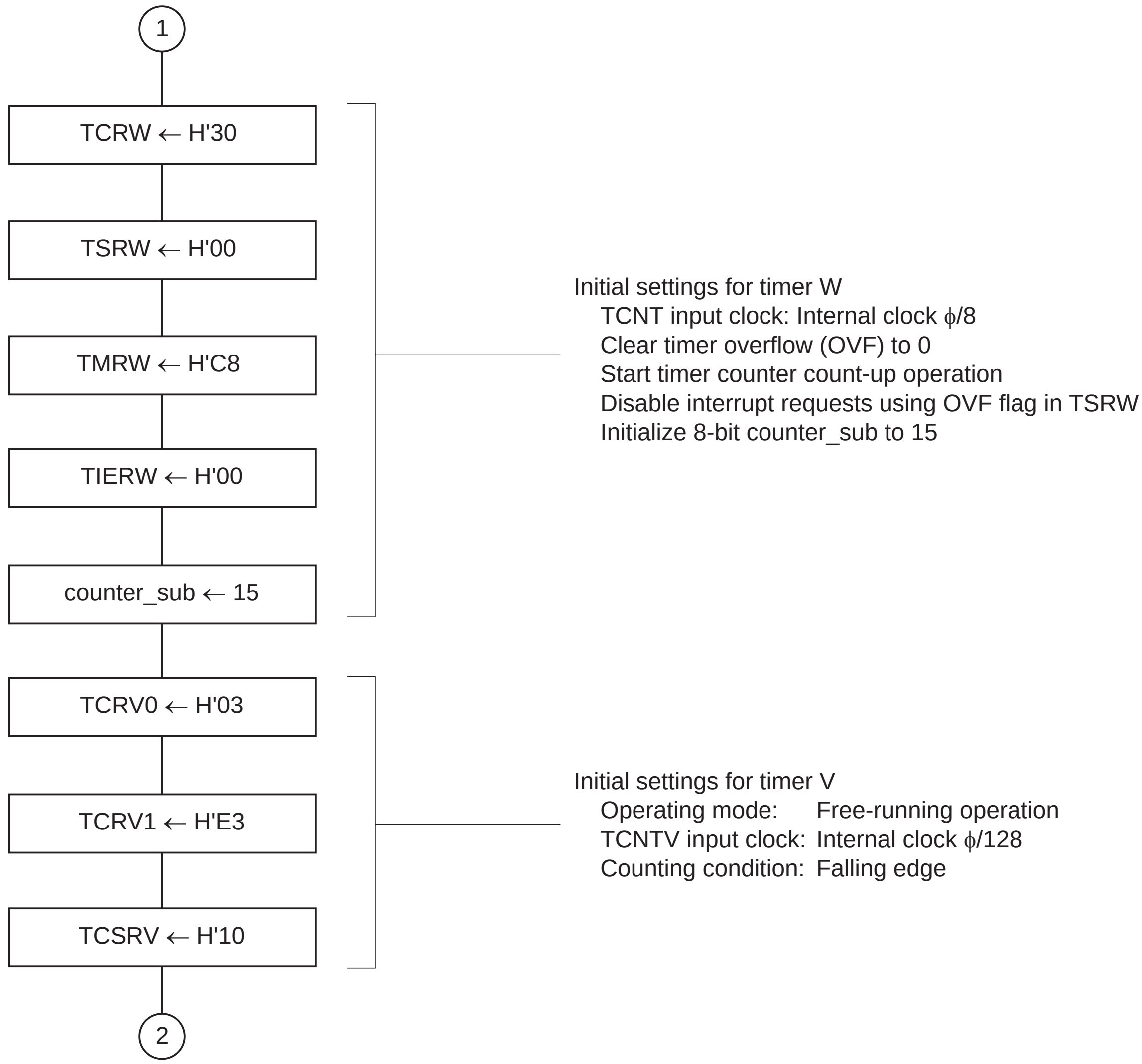
Set CCR interrupt mask bit (I) to 1 to mask interrupt requests

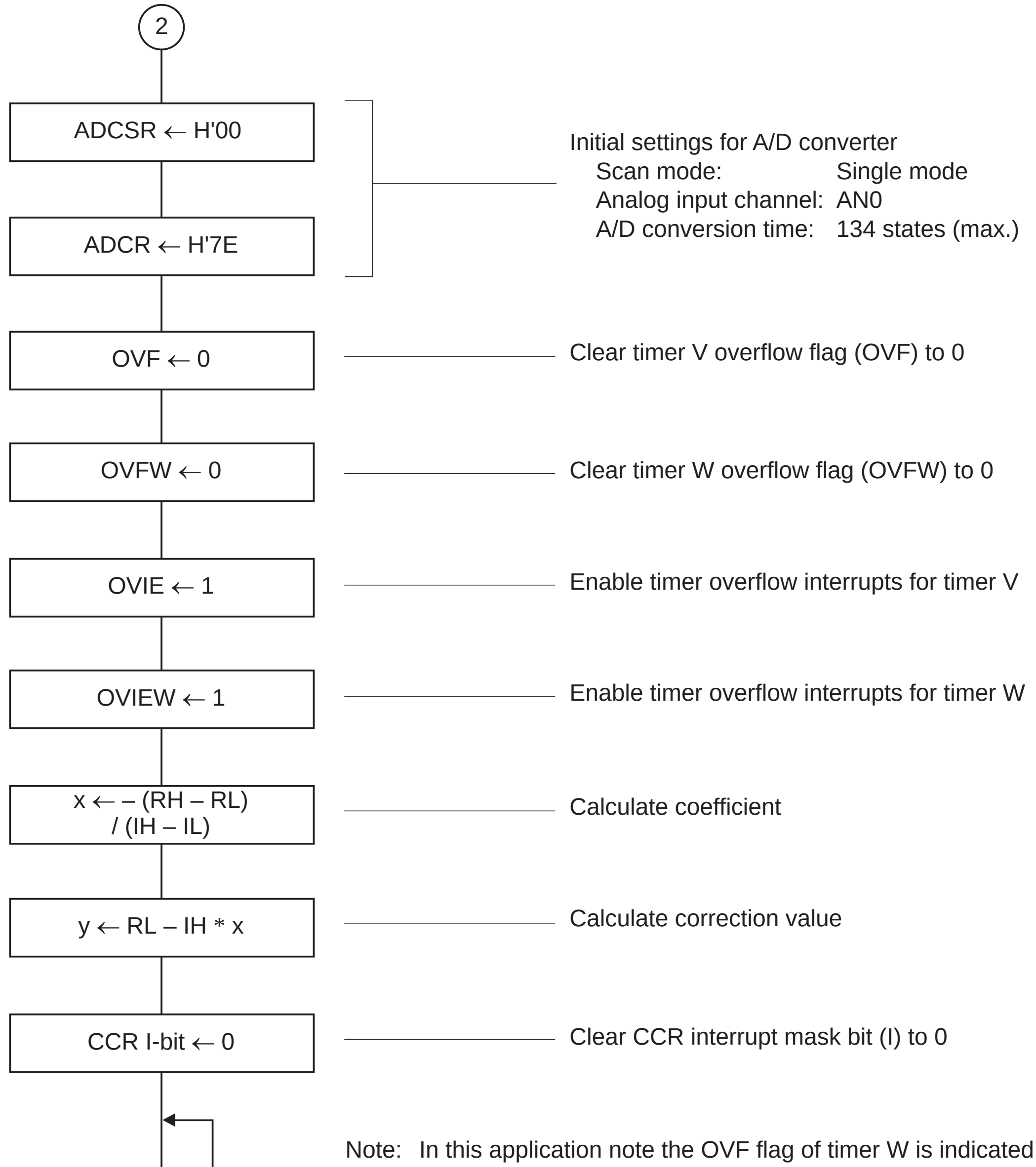
Initial settings for RAM area to be used
- dig_0 = H'40 = LED1 initial display data "_"
- dig_1 = H'40 = LED2 initial display data "_"
- dig_2 = H'40 = LED3 initial display data "_"
- dig_3 = H'40 = LED4 initial display data "_"
- cnt = H'00 = LED enable counter initial value = 0

Initial settings for port 2
- P20/SCK3: P20 output pin (output data = 0)
- P21/RXD: P21 output pin (output data = 0)
- P22/TXD: P22 input pin

Initial settings for port 5
- P50/$\overline{\text{WKP0}}$: P50 output pin (output data = 0)
- P51/$\overline{\text{WKP1}}$: P51 output pin (output data = 0)
- P52/$\overline{\text{WKP2}}$: P52 output pin (output data = 0)
- P53/$\overline{\text{WKP3}}$: P53 output pin (output data = 0)
- P54/$\overline{\text{WKP4}}$: P54 output pin (output data = 0)
- P55/$\overline{\text{WKP5}}$/$\overline{\text{ADTRG}}$: P55 output pin (output data = 0)
- P56: P56 output pin (output data = 0)
- P57: P57 output pin (output data = 0)

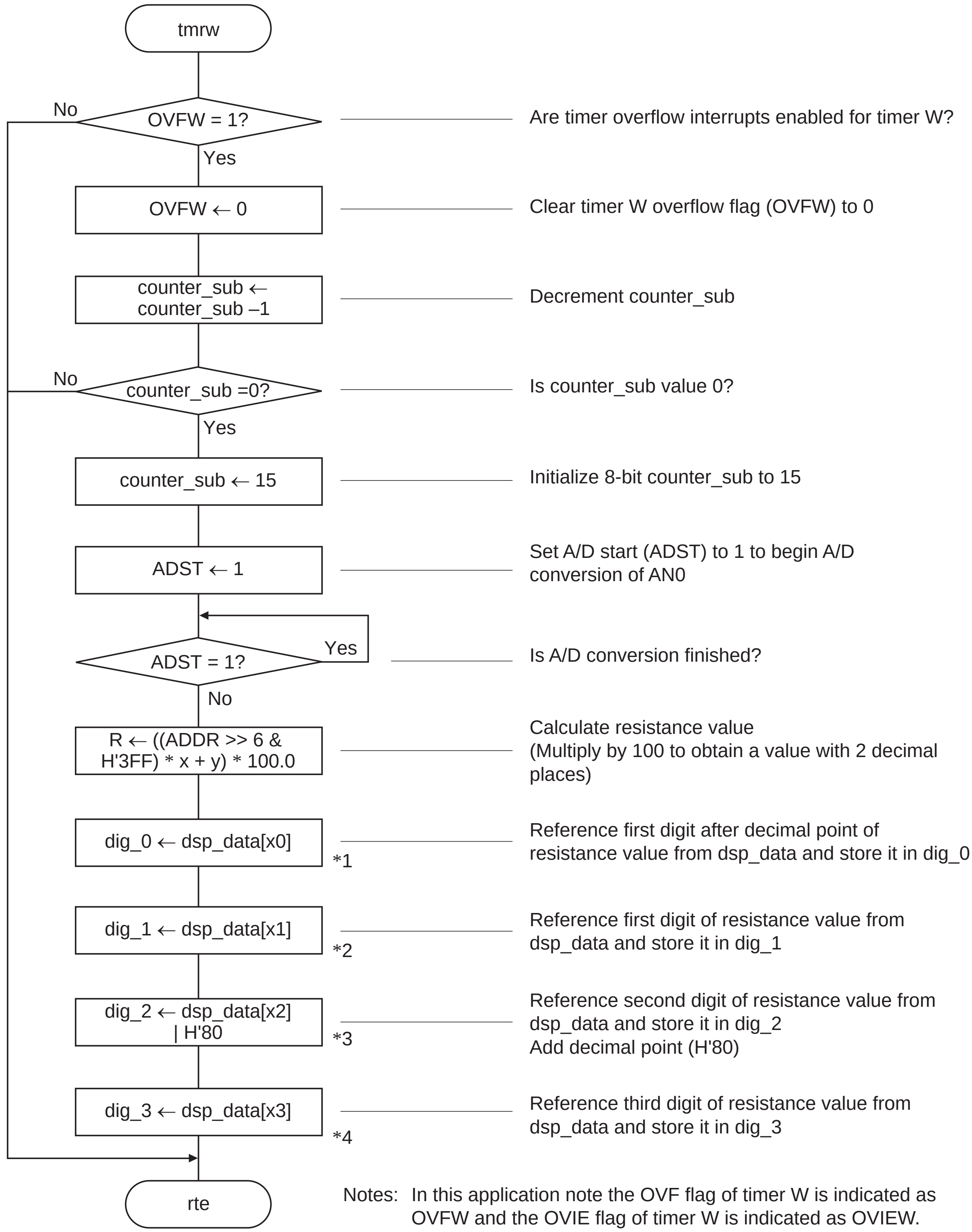
Note: * In this task example task pointer settings are performed by INIT.SCR (assembly language).


1
TCRW ← H'30
TSRW ← H'00
TMRW ← H'C8
TIERW ← H'00
counter_sub ← 15
Initial settings for timer W
TCNT input clock: Internal clock φ/8
Clear timer overflow (OVF) to 0
Start timer counter count-up operation
Disable interrupt requests using OVF flag in TSRW
Initialize 8-bit counter_sub to 15
TCRV0 ← H'03
TCRV1 ← H'E3
TCSRV ← H'10
Initial settings for timer V
Operating mode: Free-running operation
TCNTV input clock: Internal clock φ/128
Counting condition: Falling edge
2



Note: In this application note the OVF flag of timer W is indicated as OVFW and the OVIE flag of timer W is indicated as OVIEW.

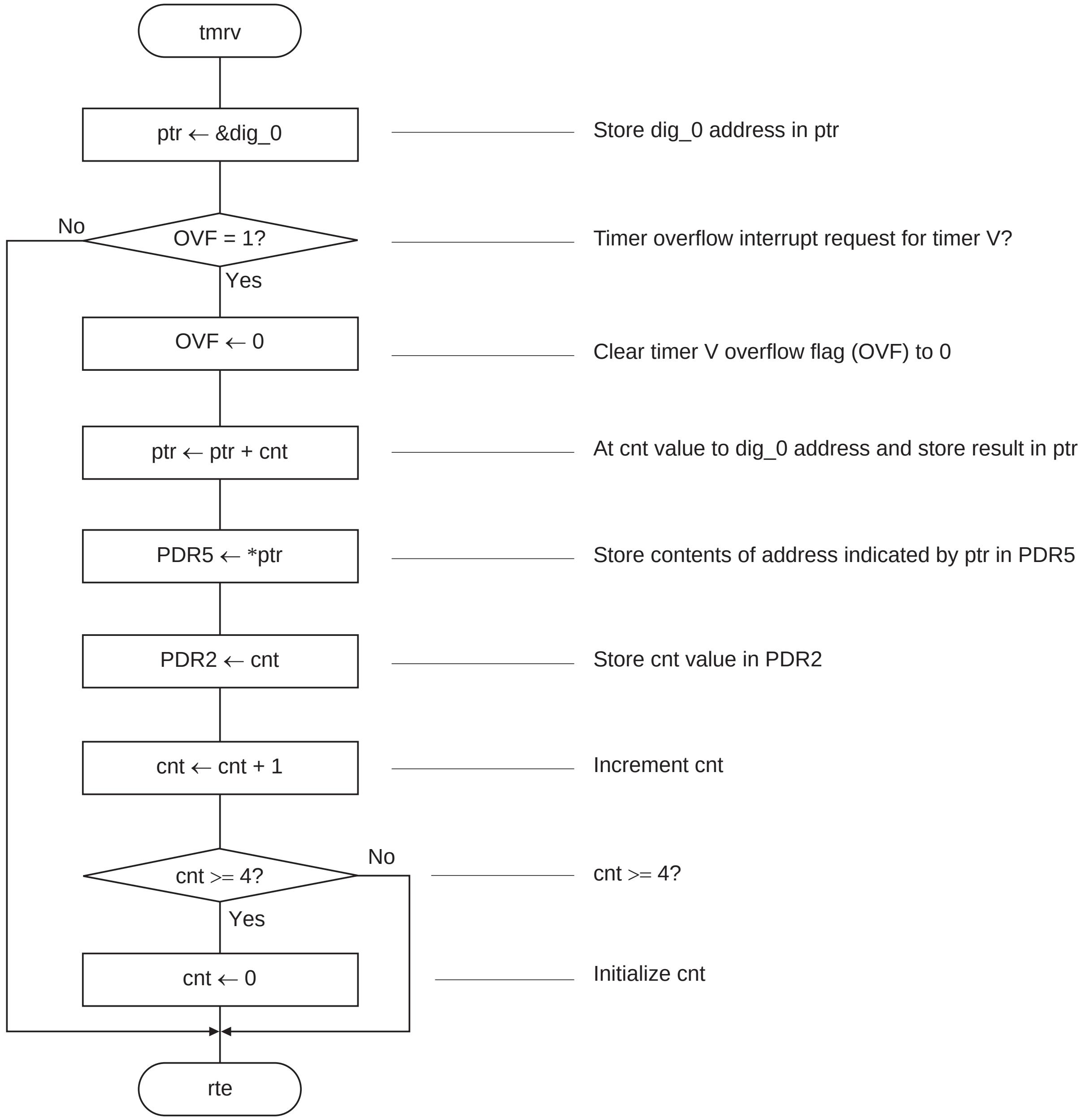
5.2 Timer W Interrupt Handling Routine (tmrw)



Notes: In this application note the OVF flag of timer W is indicated as OVFW and the OVIE flag of timer W is indicated as OVIEW.

1. ×0 = R % 10
2. ×1 = (R % 100) / 10
3. ×2 = (R % 1,000) / 100
4. ×3 = (R / 1,000)

5.3 Timer V Interrupt Handling Routine (tmrv)



6. Program Listing

Note: In this application note the OVF flag of timer W is indicated as OVFW and the OVIE flag of timer W is indicated as OVIEW.

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
```

```
/* H8/300H tiny Series -H8/36014- Application note */
/* Application */
/* Resistance value measurement example */

#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 PDR2  *(volatile unsigned char *)0xFFD5        /* Port data register 2 */
#define PCR2  *(volatile unsigned char *)0xFFE5        /* Port control register 2 */

#define PMR5  *(volatile unsigned char *)0xFFE1        /* Port mode register 5 */
#define PUCR5  *(volatile unsigned char *)0xFFD1       /* Port pull-up control register 5 */
#define PDR5  *(volatile unsigned char *)0xFFD8        /* Port data register 5 */
#define PCR5  *(volatile unsigned char *)0xFFE8        /* Port control register 5 */

#define TMRW  *(volatile unsigned char *)0xFF80        /* Timer mode register W */
#define TCRW  *(volatile unsigned char *)0xFF81        /* Timer control register W */
#define TCRW_BIT  (*(struct BIT *)0xFF81)              /* Timer Control Register W */
#define TIERW     *(volatile unsigned char *)0xFF82    /* Timer interrupt enable register W */
#define TIERW_BIT (*(struct BIT *)0xFF82)              /* Timer Interrupt Enable Register */
#define OVIEW     TIERW_BIT.b7                         /* Timer Overflow Interrupt Enable W */
#define TSRW  *(volatile unsigned char *)0xFF83        /* Timer status register W */
#define TSRW_BIT  (*(struct BIT *)0xFF83)              /* Timer Status Register W */
```

```
#define OVFW  TSRW_BIT.b7                                /* Timer Over flow W */

#define TCRV0  *(volatile unsigned char *)0xFFA0         /* Timer control register V0 */
#define TCRV0_BIT  (*(struct BIT *)0xFFA0)
#define OVIE  TCRV0_BIT.b5                               /* Timer overflow interrupt enable */
#define TCSRV  *(volatile unsigned char *)0xFFA1         /* Timer control/status register V */
#define TCSRV_BIT  (*(struct BIT *)0xFFA1)
#define OVF  TCSRV_BIT.b5                                /* Timer overflow flag */
#define TCRV1  *(volatile unsigned char *)0xFFA5         /* Timer control register V1 */

#define ADCSR  *(volatile unsigned char *)0xFFB8         /* A/D control/status register */
#define ADCSR_BIT (*(struct BIT *)0xFFB8)
#define ADST  ADCSR_BIT.b5                               /* A/D start */
#define ADCR  *(volatile unsigned char *)0xFFB9          /* A/D control register */
#define ADDRA *(volatile unsigned int *)0xFFB0           /* A/D data register A */
#define ADDRB *(volatile unsigned int *)0xFFB2           /* A/D data register A */
#define ADDRC *(volatile unsigned int *)0xFFB4           /* A/D data register A */
#define ADDRD *(volatile unsigned int *)0xFFB6           /* A/D data register A */

#define IH   513.0                                       /* INPUT HIGH */
#define IL   28.0                                        /* INPUT LOW */
#define RH   9.347                                       /* Resistance HIGH */
#define RL   0.002                                       /* Resistance LOW */

#pragma interrupt (tmrw)
#pragma interrupt (tmrv)

/* Function */
extern void INIT(void);                                  /* Stack pointer set */
void main(void);                                         /* main routine */
void tmrw(void);                                         /* Timer W interrupt routine */
void tmrv(void);                                         /* Timer V interrupt routine */

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

```
/* RAM define */
unsigned char dig_0;                                    /* Dig-0 LED display data store */
unsigned char dig_1;                                    /* Dig-1 LED display data store */
unsigned char dig_2;                                    /* Dig-2 LED display data store */
unsigned char dig_3;                                    /* Dig-3 LED display data store */
unsigned char cnt;                                      /* LED enable counter */
unsigned char counter_sub;                              /* sub-counter */
int R;                                                  /* resistance(for display) */
float x;                                                /* coefficient */
float y;                                                /* correction */

/* Vector address */
#pragma section V1                                      /* Vector section set */
void (*const VEC_TBL1[])(void) = {
      INIT                                              /* H'0000 Reset vector */
};
#pragma section V2                                      /* Vector section set */
void (*const VEC_TBL2[])(void) = {
      tmrw                                              /* H'002a Timer W interrupt vector */
};
#pragma section V3                                      /* Vector section set */
void (*const VEC_TBL3[])(void) = {
      tmrv                                              /* H'002c Timer V interrupt vector */
};
#pragma section                                         /* P */

/*******************************************************/
/* Main program                                         */
/*******************************************************/
void main(void)
{
     set_imask_ccr(1);                                  /* CCR I-bit = 1 */

  dig_0 = 0x40;                                         /* Used RAM area initialize */
  dig_1 = 0x40;                                         /* Used RAM area initialize */
  dig_2 = 0x40;                                         /* Used RAM area initialize */
  dig_3 = 0x40;                                         /* Used RAM area initialize */
  cnt   = 0x00;                                         /* Used RAM area initialize */

  PCR2  = 0x03;                                         /* Port 2 initialize */

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

                                                        /* Timer W initialize */
  TCRW  = 0x30;                                         /* Clock Select */
  TSRW  = 0x00;                                         /* Clear OVF */
  TMRW  = 0xC8;                                         /* 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 */

  x = - (RH - RL) / (IH - IL);                 /* Get coefficient */
  y = RL - IH * x;                             /* Get correction */

  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 ? */
               R = ((ADDRA >> 6 & 0x03ff) * x + y) * 100.0;
               dig_0 = dsp_data[R % 10];       /* Dig-0 LED display data set */
               dig_1 = dsp_data[(R % 100)/10]; /* Dig-1 LED display data set */
               dig_2 = dsp_data[(R % 1000)/100] | 0x80; /* Dig-2 LED display data set */
               dig_3 = dsp_data[R / 1000];     /* Dig-3 LED display data set */
         }
    }
}

/*******************************************************/
/* Timer V Interrupt                                   */
/*******************************************************/
void tmrv(void)
{
unsigned char *ptr;                            /* Pointer set */

    ptr = &dig_0;                              /* LED display data store address set */

    while(OVF == 1){                           /* OVF = 1 ? */
        OVF  = 0;                              /* Clear OVF to 0 */
        ptr += cnt;                            /* LED display data read */
        PDR5 = *ptr;                           /* LED display data output */
```

```
        PDR2 = cnt;                                 /* LED enable data output */
        cnt++;                                      /* "cnt" increment */
        if (cnt >= 4){                              /* 4 times end ? */
            cnt = 0;                                /* "cnt" initialize */
        }
    }
}
```

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Dec.20.04	—	First edition issued