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

Producing a "Beep"

Introduction

Beep sounds (do, re, mi, fa, sol, la, ti, do) are assigned to eight key switches and a 'beep' is output while a switch is pressed. A 'beep' is produced by converting a PWM output waveform by an external D/A converter and is output from a speaker.

Target Device

H8/38024

Contents

1. Specifications

1. Eight key switches are assigned a beep sound to be output while a switch is being pressed.
2. A PWM output waveform is converted by an external D/A converter circuit to output a beep sound from the speaker.
3. The high-level width of the PWM waveform is varied. A sine wave data table is stored in the ROM in advance to output signal in accordance with the variations of that width from the PWM2 pin at constant intervals.
4. Figure 1.1 shows the connection of beep sound output.

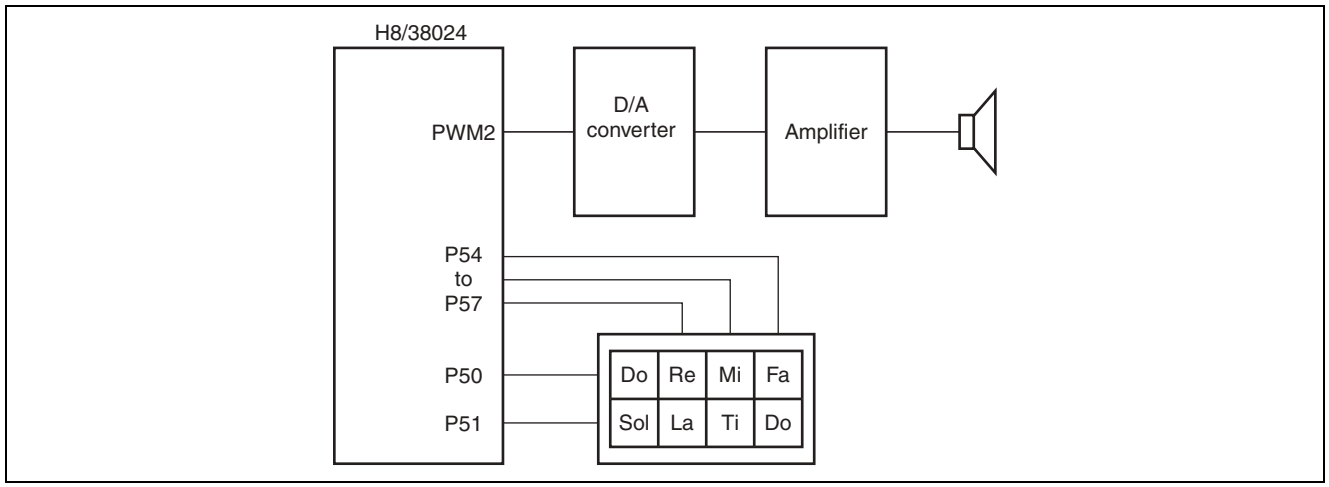


Figure 1.1 Configuration for the Sample Task

2. Concepts

1. The high-level width of the PWM waveform is changed every PWM conversion cycle.
2. An external circuit is connected which changes the voltage value according to variation in the high-level width of the PWM waveform.
3. A PWM high-level width data table is stored in the ROM in advance and, for each musical note, one data value from that table is output every PWM conversion cycle.
4. One PWM conversion cycle is determined through a compare-match interrupt generated by timer FH.

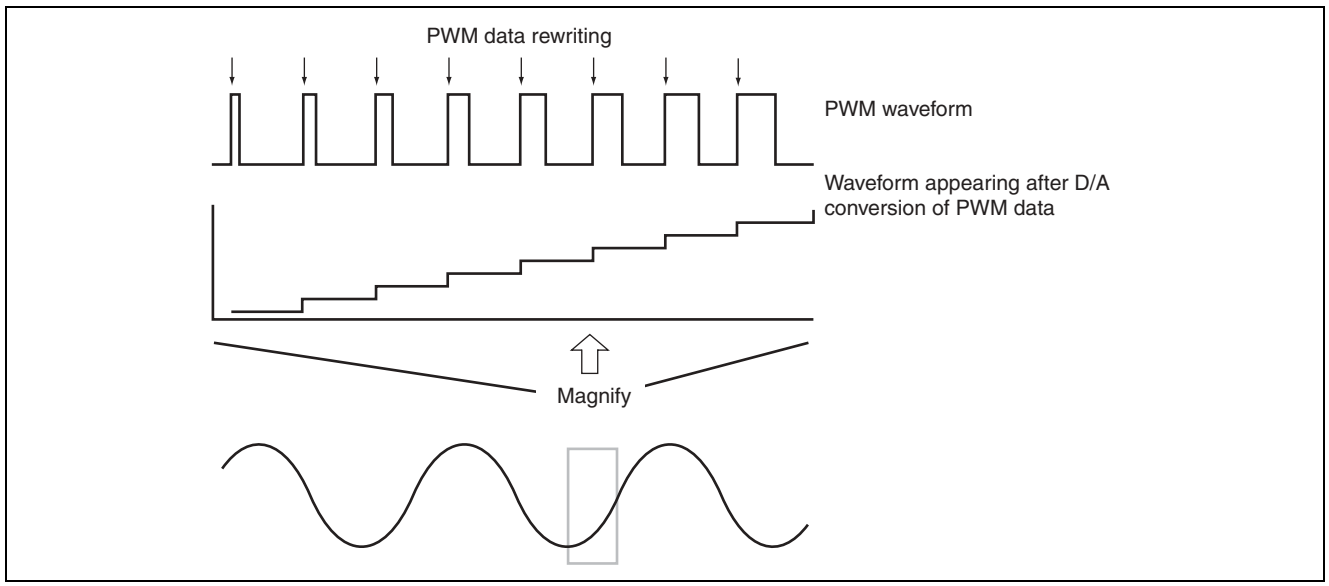


Figure 2.1 Sample Beep Sound Output Waveforms

3. Description of Functions

1. The following explains the H8/38024's functions used to output beep sounds. Figure 3.1 shows the block diagram of the functions involved in this sample task.

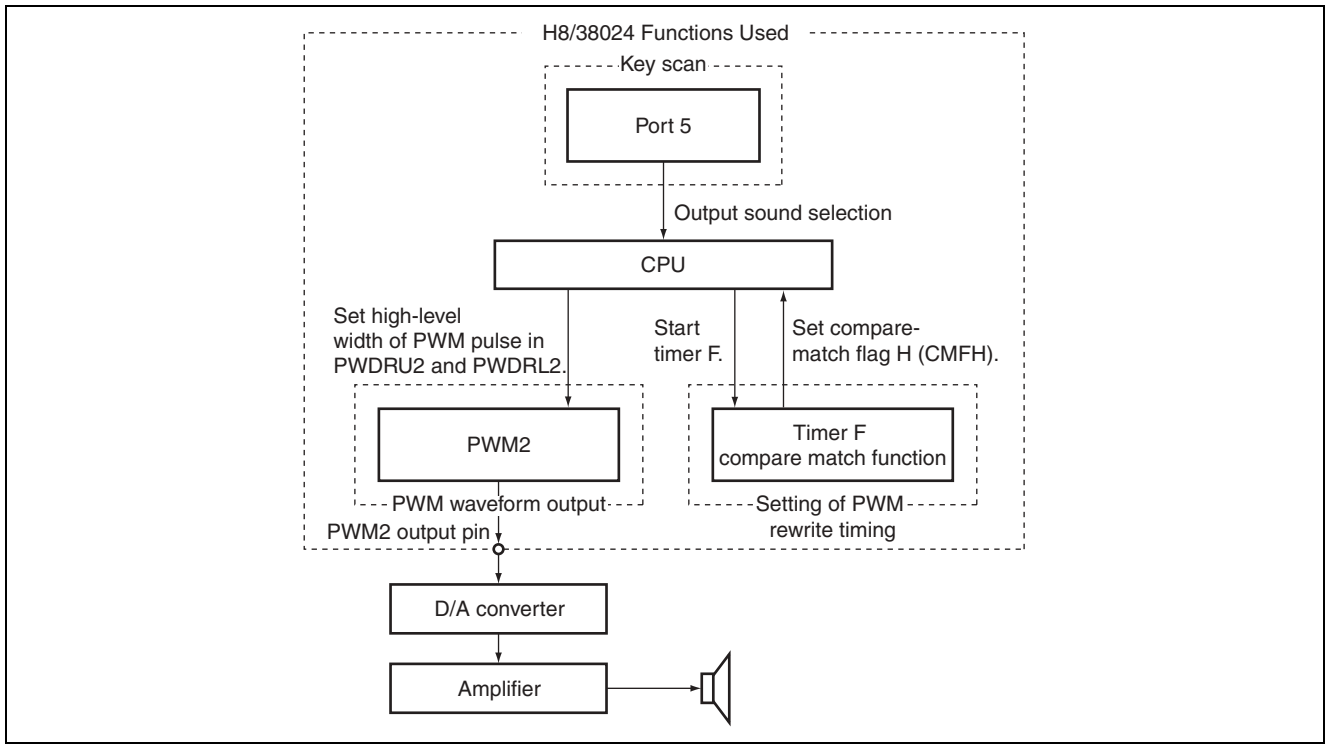


Figure 3.1 H8/38024 Functions Used

- Port 5
  This port is connected to the key switch and selects a beep sound.
- PWM2
  Outputs a PWM waveform for generating a sine wave to be output to the speaker.
- Timer F compare-match function
  Sets a flag at predetermined intervals to change the high-level width of the PWM pulse every PWM conversion cycle.

3. This section describes the functions of the PWM2.
   A. Figure 3.2 shows the block diagram of the 10-bit PWM function. The following explains the block diagram of this function.
      - The system clock (ϕ), which is rated at 5 MHz, is used as a reference clock to operate the CPU and its peripheral functions.
      - The PWM2 control register (PWCR2) is an 8-bit write-only register that selects an input clock.
      - A pulse-division method is used to reduce ripples.
      - The PWM2 data registers U and L (PWDRU2 and PWDRL2) are 10-bit write-only registers. The upper two bits provide PWDRU2, while the lower eight bits provide PWDRL2. The contents written to PWDRU2 and PWDRL2 correspond to the total of high level width over one PWM waveform cycle. Writing 10-bit data to PWDRU2 and PWDRL2 moves their contents to the PWM waveform generator, and updates the data for PWM waveform generation. To set 10-bit data, the lower eight bits must be written to PWDRL2 first, and then the upper two bits to PWDRU2.
      - The port mode register 9 (PMR9) is an 8-bit readable/writable register that controls selection of the pin functions of port 9. Use the P90/PWM2 pin function select bit (PWM2), PMR9's bit 0, to set the P91/PWM2 pin as the PWM2 output pin.
      - The PWM2 output pin (PWM2) outputs a PWM waveform based on the pulse division method.

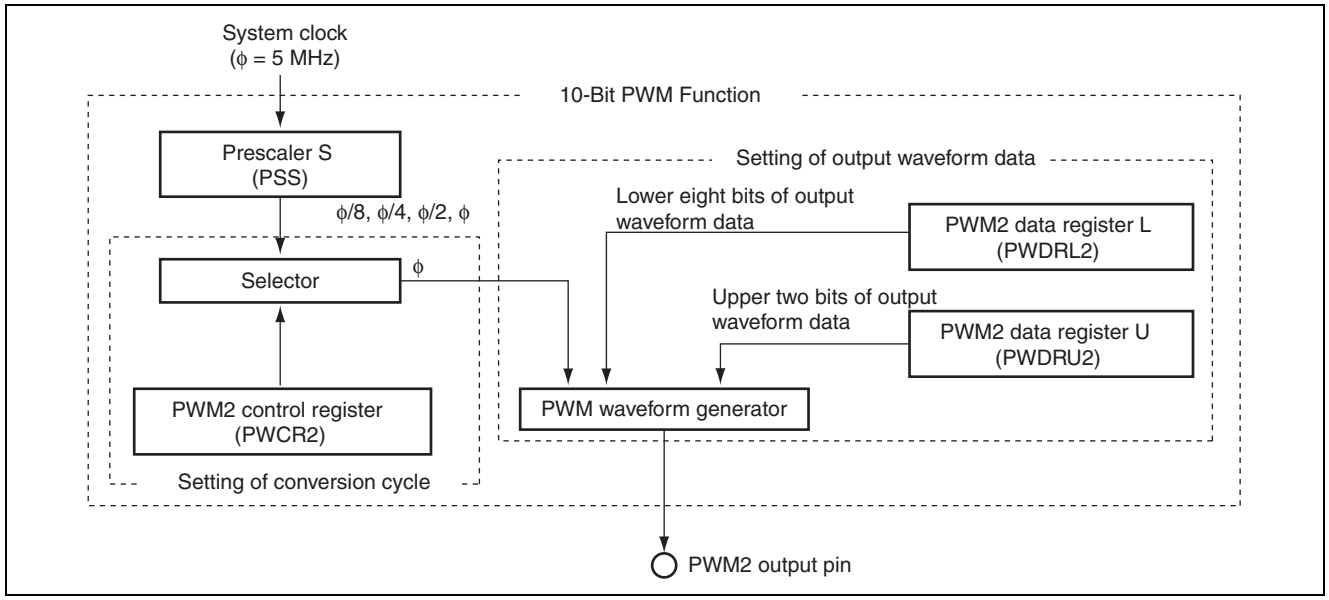


Figure 3.2 Block Diagram of PWM2

3. This section explains port 5 functions.
   A. Figure 3.3 shows the block diagram of the key input circuit connected to port 5.

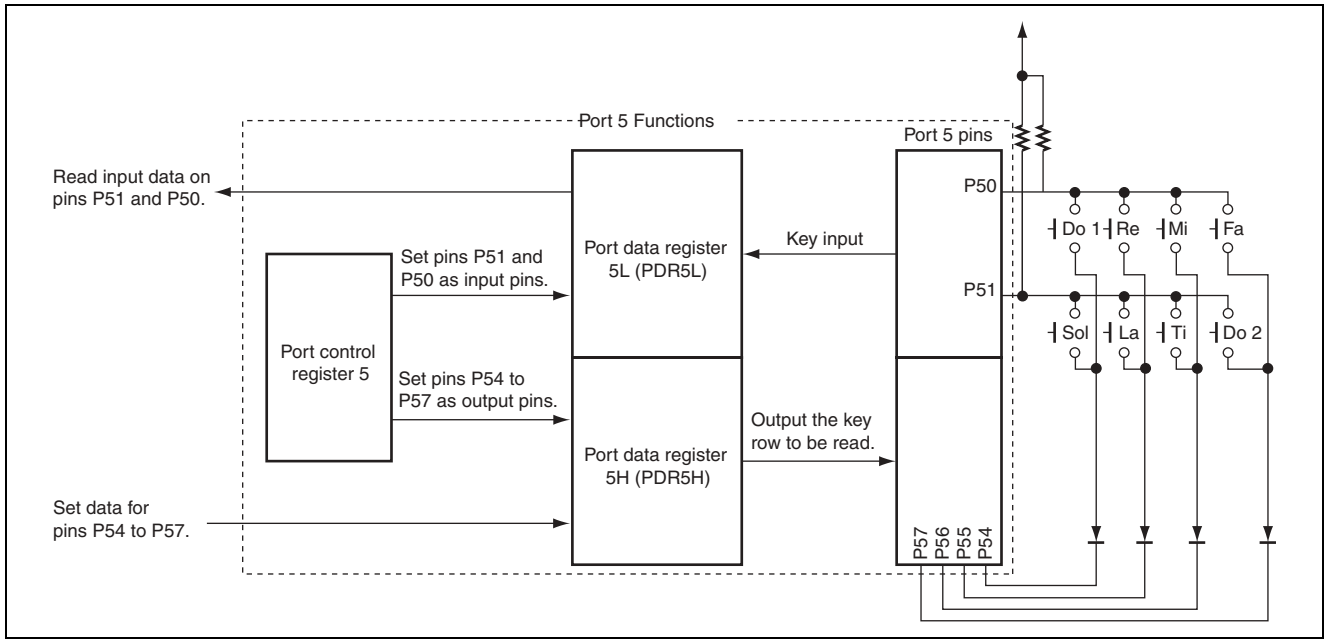


Figure 3.3 Block Diagram of Key Input Circuit on Port 5

B. In the sample task, the port 5 is used for key input. The following explains the block diagram of the key input circuit.
   - Port control register 5 (PCR5)
     Sets input/output for the port 5. When PCR5 = 0xF0, pins P54 to P57 are set as the output ports, while pins P50 to P53 are set as the input ports.
   - Port data register 5H (PDR5H)
     The upper four bits of PDR5 provide PDR5H. This register is used to set data to be stored into output ports P54 to P57. The state of the key row, connected to one of P54 to P57 pins, set at the low level is reflected on pins P51 and P50.
   - Port data register 5L (PDR5L)
     The lower four bits of PDR5 provide PDR5L. The values of input ports P50 to P53 are reflected in PDR5L. The value stored in PDR5L indicates the state of the key row selected by PDR5H. The sample task uses only pins P51 and P50.

4. This section describes timer F functions.
   A. Figure 3.4 shows the block diagram of the timer F compare-match function.

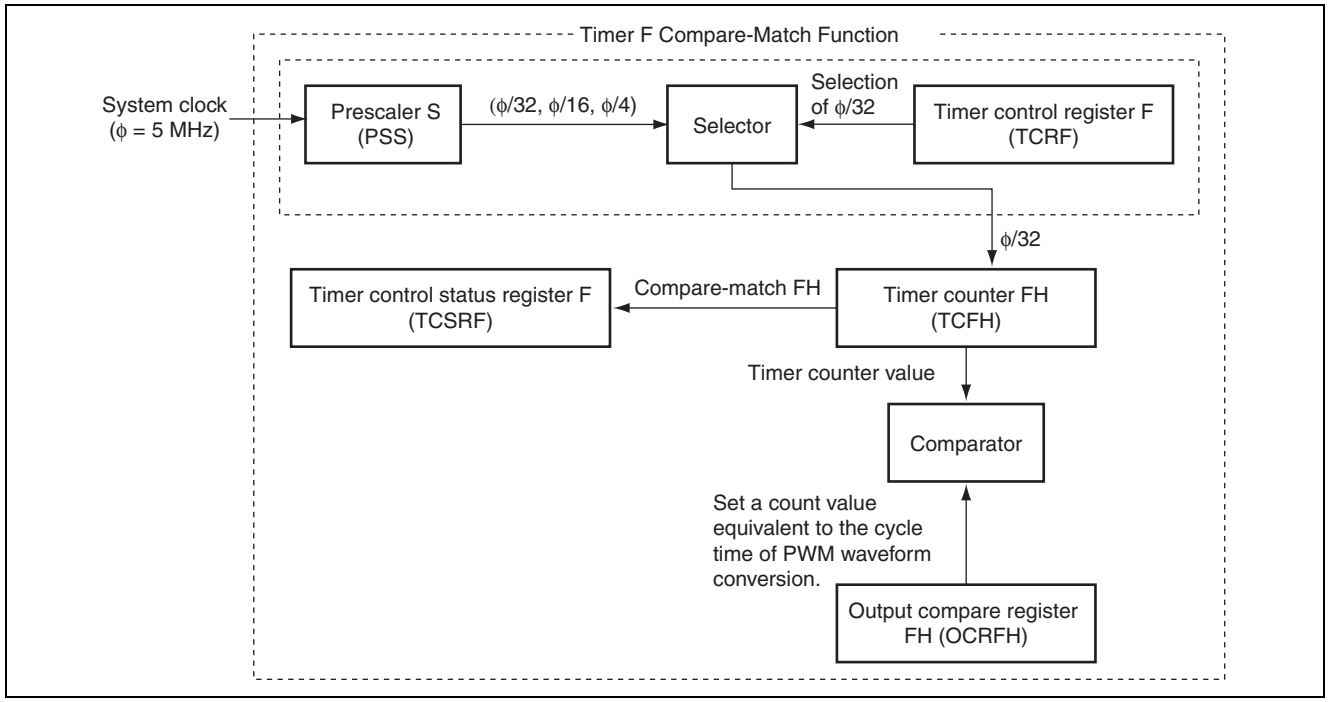


Figure 3.4 Block Diagram of Timer F Compare-match Function

B. The sample task uses timer F's compare-match function to generate an interrupt at intervals of 102.4 ms, which is the PWM conversion cycle time, to change the high-level width of the PWM pulse.
   - System clock (ϕ)
     A 5 MHz clock, which is used as the reference clock to operate the CPU and peripheral functions.
   - Prescaler S (PSS)
     This is a 13-bit counter that receives ϕ as the input and is incremented every cycle.
   - Timer counter F (TCF)
     This is a 16-bit readable/writable up-counter that is incremented by the input clock, which can be selected from among ϕ/4, ϕ/16, ϕ/32, and ϕ_w/4 (watch clock divided by 4). In this sample task, ϕ/4 is used as the input clock.
   - Timer control register F (TCRF)
     This is an 8-bit readable/writable register that selects 16-bit or 8-bit mode, the input clock from among four internal clocks, and use of external events, and sets the output levels of pins TMOFH and TMOFL.
   - Timer control status register F (TCSRF)
     This is an 8-bit register that selects counter clearing by compare-match, sets an overflow flag and a compare-match flag, and controls whether to enable an interrupt request due to an overflow.

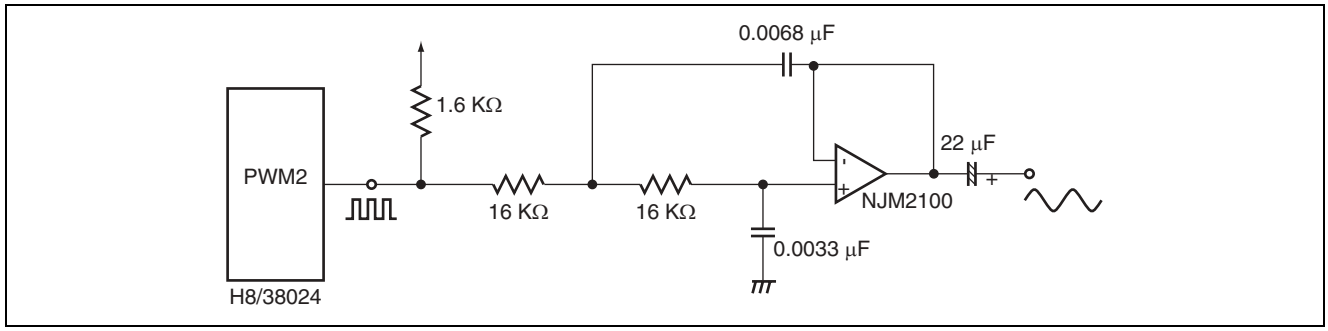
5. Table 3.1 lists the functions assigned for the sample task. With these function assignments, beep sounds are produced using the PWM waveform.

Table 3.1 Function Assignments

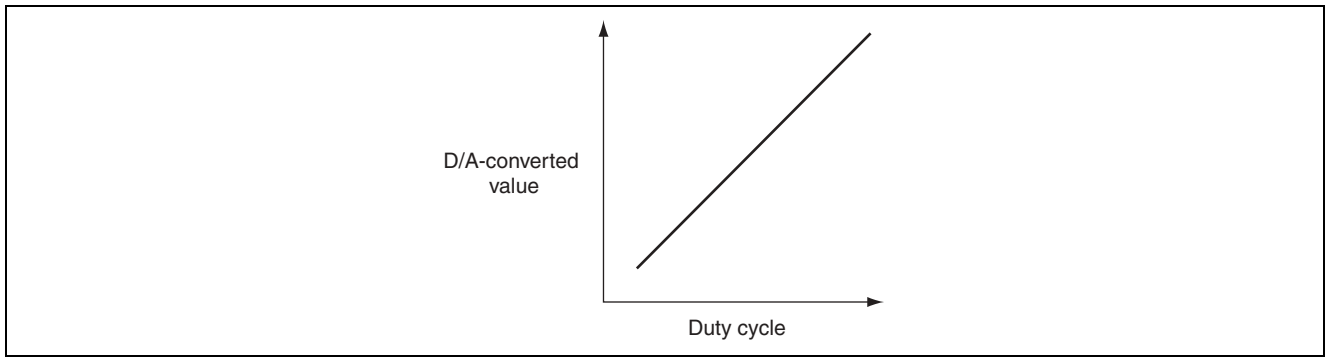
Functions	Function Allocation
PSS	13-bit up-counter that receives the system clock (5 MHz) as the input
TCRF	Selects a clock to be supplied to timer F.
TCSRF	Selects clearing of TCFH upon compare-match (every 102.4 μs) and sets the flag.
TCFH	Timer F counter
OCRFH	Should be set so that a compare-match occurs at intervals of 102.4 μs, which is the PWM conversion cycle time.
PWCR2	Selects a clock to be supplied to the 10-bit PWM.
PWDRU2	Sets the upper two bits of PWM output waveform data.
PWDRL2	Sets the lower eight bits of PWM output waveform data.
PWM2	PWM waveform output pin
PCR5	Sets the port 5 input/output function.
PDR5H, PDR5L	2 × 4 key scanning
PMR5	Sets the port 5 input/output pin.
PUCR5	Sets the port 5 MOS pull-up to off.

4. Principle of Operation

1. This section describes the operation for producing beep sound output.
   A. The figure below shows the PWM-D/A converter circuit for converting a PWM waveform into an analog waveform.



- In the sample task, the minimum variation width of the PWM waveform is 25.6 µs. The same PWM waveform is output four times.
- In this circuit, a lower duty cycle produces a lower D/A conversion value, and a higher duty cycle produces a higher D/A conversion value.



B. Table 4.1 lists the output pulse frequencies for the musical notes.

Table 4.1 Output Pulse Frequencies for Musical Notes

Item \ Musical Notes	Do 1	Re	Mi	Fa	Sol	La	Ti	Do 2
Period [ms]	1.95	1.74	1.54	1.43	1.23	1.13	1.02	0.92
Frequency [Hz]	514	574	651	698	814	888	977	1085
Number of samples in a conversion cycle	19	17	15	14	12	11	10	9

- Methods to calculate the period and frequency of the musical notes
  - — PWM pulse conversion cycle time in this sample task
    512/ϕ = 512/5 MHz = 102.4 μs
  - — Period of each musical note
    Number of samples in a conversion cycle for the musical note $\times$102.4 μs
  - — Frequency of each musical note
    1/period of the musical note

Example:
- — Period of scale Do 1
  Number of samples in a conversion cycle $\times$ 102.4 μs = 19 $\times$ 102.4 μs = 1.945 ms = 1.95 ms
- — Frequency of scale Do 1
  1/period of the musical note = 1/1.945 ms = 514 Hz

C. Data table

Table 4.2 lists settings of PWDRU2 and PWDRL2 for each musical note.

Table 4.2 Settings of PWDRU2 and PWDRL2 for Each Musical note

Musical Note / No.	Do 1	Re	Mi	Fa	Sol	La	Ti	Do 2
1	527	527	527	527	527	527	527	527
2	627	639	653	661	682	695	709	725
3	716	735	757	769	795	809	821	832
4	785	804	821	829	835	830	822	797
5	827	836	835	829	796	761	711	638
6	836	826	796	770	684	614	530	427
7	812	776	711	662	530	440	348	262
8	758	692	594	528	375	293	233	220
9	679	587	466	393	260	220	231	320
10	584	473	348	285	220	245	341	
11	482	367	260	225	256	359		
12	386	282	220	224	367			
13	304	230	231	284				
14	247	220	293	391				
15	220	247	396					
16	224	314						
17	262	410						
18	328							
19	415							

5. Description of Software

5.1 Modules

Table 5.1 lists the modules of the sample task.

Table 5.1 Description of Modules

Module	Label	Functions
Main routine	main	Sets global variables and PWM2.
Beep sound output	beep	Routine that sets a duty cycle for PWM2.
Port 5L reading	keyread	Returns the contents of P50 to P53.

5.2 Arguments

Table 5.2 describes the arguments used in this sample task.

Table 5.2 Description of Arguments

Argument	Functions	Label	Data Length	Input/ Output
beep_no	Selects a beep sound.	beep	One byte	Input

5.3 Internal Registers

Table 5.3 lists the available internal registers used for the sample task.

Table 5.3 Description of Internal Registers

Register		Functions	Address	Setting
TCRF	CKSH2 CKSH1 CKSH0	Timer control register F (clock select H) When CKSH2 = 1, CKSH1 = 1, and CKSH0 = 0, TCFH is incremented by internal clock ϕ/4.	0xFFB6 Bit 6 Bit 5 Bit 4	CKSH2 = 1 CKSH1 = 1 CKSH0 = 0
TCSRF	OVFH	Timer control status register F (timer overflow flag H) When OVFH = 0, the TCF has not overflowed. When OVFH = 1, the TCF has overflowed.	0xFFB7 Bit 7	0
TCSRF	CMFH	Timer control status register F (compare-match flag H) When CMFH = 0, a compare-match F has not occurred. When CMFH = 1, a compare-match F has occurred.	0xFFB7 Bit 6	0
	OVIEH	Timer control status register F (timer overflow interrupt enable H) When OVIEH = 0, disables an interrupt request by TCF overflow. When OVIEH = 1, enables an interrupt request by TCF overflow.	0xFFB7 Bit 5	0
	CCLRH	Timer control status register F (counter clear H) When CCLRH = 0, disables TCFH clearing by compare-match. When CCLRH = 1, enables TCFH clearing by compare-match.	0xFFB7 Bit 4	1
TCFH		8-bit timer counter FH 8-bit up-counter receiving internal clock ϕ/4 as the input.	0xFFB8	0x00

Register		Functions	Address	Setting
OCRFH		Output compare register FH Compared with TCFH. When the OCRFH value matches the TCFH value, the CMFH bit in TCSRF is set to 1.	0xFFBA	0x80
PMR5		Port mode register 5 (P5n/$\overline{\text{WKPn}}$/SEGn+1 pin function selection) Pin function when the pin is not specified as SEGn+1 When PMR5 = 0x00, pin P5n/$\overline{\text{WKPn}}$/SEGn+1 functions as the P5n input/output pin.	0xFFCC	0x00
PWCR2	PWCR20 PWCR21	PWM2 control register (clock select 1 and 0) When PWCR20 = 1 and PWCR21 = 1, sets ϕ as the clock supplied to the 10-bit PWM.	0xFFCD Bit 1 Bit 0	PWCR10 = 1 PWCR11 = 1
PWDRU2		PWM2 data register U Sets the upper two bits of PWM output waveform data.	0xFFCE	0x00
PWDRL2		PWM2 data register L Sets the lower eight bits of PWM output waveform data.	0xFFCF	0x00
PDR5	PDR5H	Port data register 5 (P54 to P57) The upper four bits of PDR5 make up PDR5H, and sets data to be stored into output ports P54 to P57. Out of the data in P54 to P57, the status of the key row set at the low level is reflected in P50 to P53. When PDR5 = 0xE0, selects P54 key row (0, 4). When PDR5 = 0xD0, selects P55 key row (1, 5). When PDR5 = 0xB0, selects P56 key row (2, 6). When PDR5 = 0x70, selects P57 key row (3, 7).	0xFFD8 Bits 4 to 7	—
	PDR5L	Port data register 5 (P50 to P53) The lower four bits of PDR5 make up PDR5L. The values on input ports P50 to P53 are reflected here.	0xFFD8 Bits 0 to 3	—
PUCR5		Port pull-up control register 5 When PUCR5 = 0x00, turns off the MOS pull-up.	0xFFE2	0x00
PCR5		Port control register 5 When PCR5 = 0xF0, sets P54 to P57 as the output ports, and P50 to P53 as the input ports.	0xFFE8	0xF0
PMR9	PWM2	Port mode register 9 (P91/PWM2 pin function selection) When PWM2 = 0, sets the P91/PWM2 pin as the P91 output pin. When PWM2 = 1, sets the P91/PWM2 pin as the PWM2 output pin.	0xFFEC Bit 1	1

5.4 Description of RAM

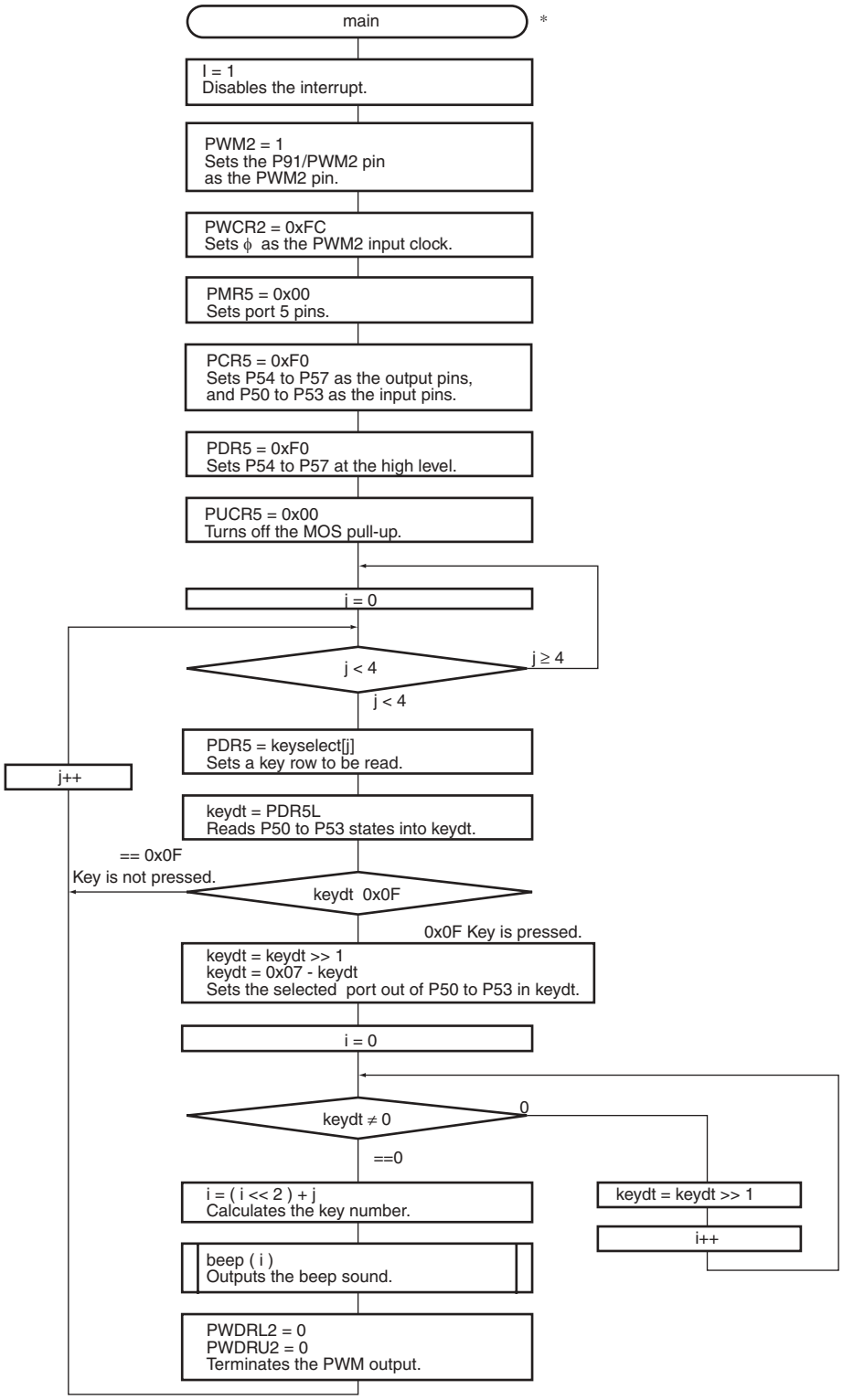
Table 5.4 lists the RAM used for the sample task.

Table 5.4 Description of RAM

Label	Function	Address	Used in
keyselect	Selects a key row to be read.	4 bytes	main

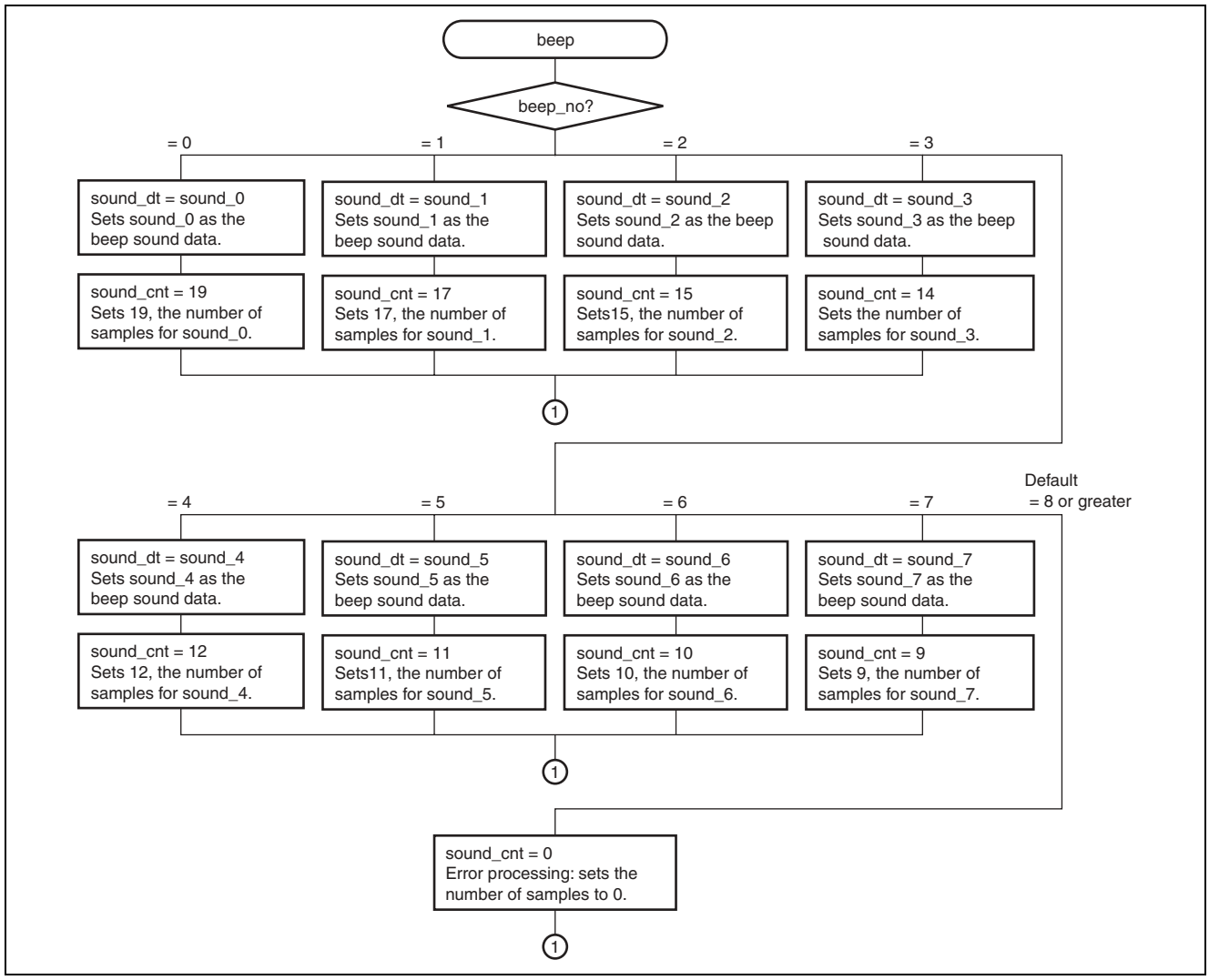
6. Flowchart

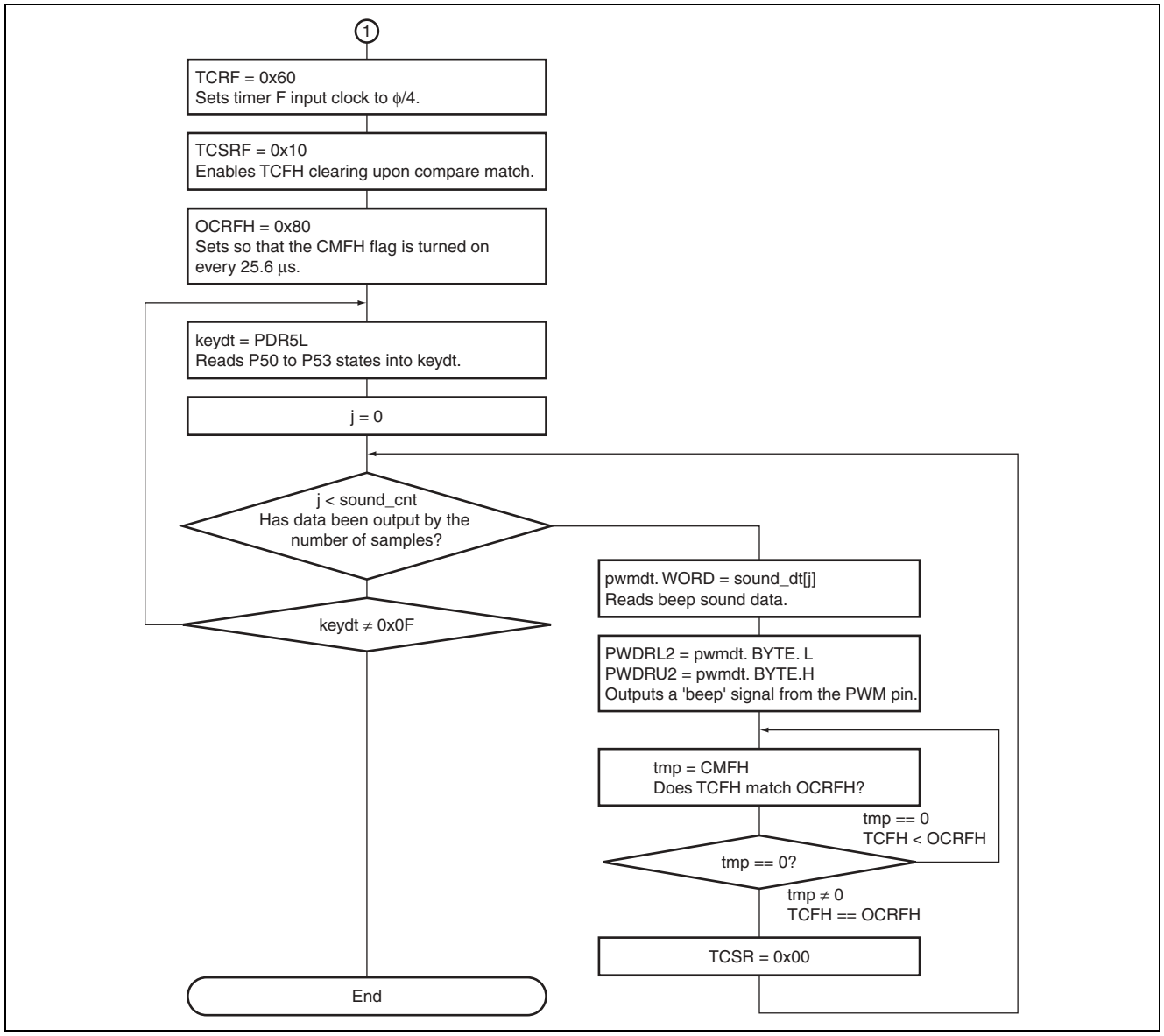
1. Main routine



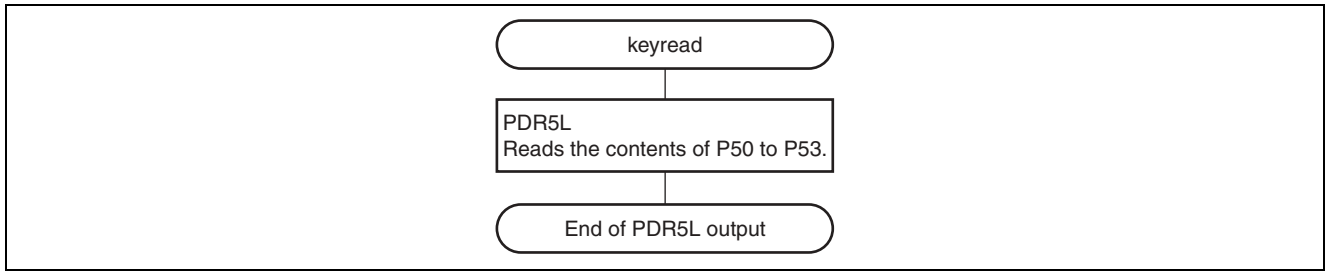
Note: * In the sample task, the stack pointer is set in INIT.SRC (assembly language).

2. Beep sound output





3. Port 5L reading



7. 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                                                                                                    */
/*                                                                                                                      */
/*  'Beep Sound Output by 10-bit PWM Function'                                                                          */
/*                                                                                                                      */
/*  Function                                                                                                            */
/*  : 10bit PWM                                                                                                         */
/*                                                                                                                      */
/*  External Clock : 10MHz                                                                                              */
/*  Internal Clock : 5MHz                                                                                               */
/*  Sub Clock      : 32.768kHz                                                                                          */
/*                                                                                                                      */
/************************************************************************************************************************/

#include    <machine.h>

/************************************************************************************************************************/
/*  Symbol Definition                                                                                                   */
/************************************************************************************************************************/
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 */
};

struct P4BIT {
    unsigned char   H:4;        /* bit7-bit4 */
    unsigned char   L:4;        /* bit3-bit0 */
};

#define     TCRF        *(volatile unsigned char *)0xFFB6       /* Timer Control Register F                */
#define     TCRF_BIT    (*(struct BIT *)0xFFB6)                 /* Timer Control Register F                */
#define     TOLH        TCRF_BIT.b7                             /* Toggle Output Level F                   */
#define     CKSH2       TCRF_BIT.b6                             /* Clock Select H2                         */
#define     CKSH1       TCRF_BIT.b5                             /* Clock Select H1                         */
#define     CKSH0       TCRF_BIT.b4                             /* Clock Select H0                         */
```

```
#define     TCSRF       *(volatile unsigned char *)0xFFB7       /* Timer Control Status Register F          */
#define     TCSRF_BIT   (*(struct BIT *)0xFFB7)                 /* Timer Control Status Register F          */
#define     OVFH        TCSRF_BIT.b7                            /* Timer Overflow Flag H                    */
#define     CMFH        TCSRF_BIT.b6                            /* Compare Match Flag H                     */
#define     OVIEH       TCSRF_BIT.b5                            /* Timer Overflow Interrupt Enable          */
#define     CCLRH       TCSRF_BIT.b4                            /* Output Select 3                          */
#define     TCFH        *(volatile unsigned char *)0xFFB8       /* Timer Counter FH                         */
#define     OCRFH       *(volatile unsigned char *)0xFFBA       /* Output Compare Register FH               */
#define     PMR5        *(volatile unsigned char *)0xFFCC       /* Port mode register 5                     */
#define     PWCR2       *(volatile unsigned char *)0xFFCD       /* PWM2 Control Register                    */
#define     PWCR2_BIT   (*(struct BIT *)0xFFCD)                 /* PWM2 Control Register                    */
#define     PWCR21      PWCR2_BIT.b1                            /* Clock Select                             */
#define     PWCR20      PWCR2_BIT.b0                            /* Clock Select                             */
#define     PWDRU2      *(volatile unsigned char *)0xFFCE       /* PWM Data Registers U                     */
#define     PWDRL2      *(volatile unsigned char *)0xFFCF       /* PWM Data Registers L                     */
#define     PDR5        *(volatile unsigned char *)0xFFD8       /* Port data register 5                     */
#define     PDR5_BIT    (*(struct P4BIT *)0xFFD8)               /* Port mode register 5                     */
#define     PDR5H       PDR5_BIT.H                              /* P57-P54                                  */
#define     PDR5L       PDR5_BIT.L                              /* P53-P50                                  */
#define     PUCR5       *(volatile unsigned char *)0xFFE2       /* Port pull-up control register 5          */
#define     PCR5        *(volatile unsigned char *)0xFFE8       /* Port control register 5                  */
#define     PMR9_BIT    (*(struct BIT *)0xFFEC)                 /* Port Mode Register 9                     */
#define     PWM2        PMR9_BIT.b1                             /* P91/PWM2 pin function switches           */

/************************************************************************************************************/
/*  Function define                                                                                         */
/************************************************************************************************************/
extern void INIT ( void );                                      /* SP Set                                   */
void main( void );
void beep( unsigned char beep_no );
unsigned char keyread( void );

unsigned char keyselect[4] = {
    0xE0,
    0xD0,
    0xB0,
    0x70,
};

/************************************************************************************************************/
/*  Vector Address                                                                                          */
/************************************************************************************************************/
#pragma section     V1                                          /* Vector Section Set                       */
void (*const VEC_TBL1[])(void) = {                              /* 0x0000 - 0x000F                          */
    INIT                                                        /* 0x0000 Reset                             */
};

extern unsigned short   sound_0[19];                            /* Sound Sample Data                        */
extern unsigned short   sound_1[17];                            /* Sound Sample Data                        */
extern unsigned short   sound_2[15];                            /* Sound Sample Data                        */
extern unsigned short   sound_3[14];                            /* Sound Sample Data                        */
extern unsigned short   sound_4[12];                            /* Sound Sample Data                        */
extern unsigned short   sound_5[11];                            /* Sound Sample Data                        */
extern unsigned short   sound_6[10];                            /* Sound Sample Data                        */
extern unsigned short   sound_7[9];                             /* Sound Sample Data                        */

#pragma section                                                 /* P                                        */
```

```
/*****************************************************************************************************************/
/*  Main Program                                                                                                 */
/*****************************************************************************************************************/
void main( void )
{
    unsigned char   i,j,keydt;

    set_imask_ccr(1);                                       /* Interrupt Disable                    */

    PWM2 = 1;                                               /* Pin function Select PWM2             */
    PWCR2 = 0xFC;                                           /* Initialize PWM Input Clock           */

    PMR5 = 0x00;                                            /* Pin function Select Port5            */
    PCR5 = 0xF0;                                            /* P57-54 Output, P53-50 Input Port     */
    PDR5 = 0xF0;                                            /* P57-54 Port "1"set                   */
    PUCR5 = 0x00;                                           /* Port5 pull-up OFF                    */

    while(1){
        for(j = 0; j < 4; j++){
            PDR5 = keyselect[j];                            /* Set Key Select                       */

            keydt = keyread();
            if(keydt ! =  0x0F){                            /* Touch Key?                           */
                keydt = keydt>>1;                           /* What Key?                            */
                keydt = 0x07 - keydt;
                for(i = 0; keydt ! =  0; i++){
                    keydt = keydt>>1;
                }

                PWDRL2 = 0;                                 /* Stop PWM                             */
                PWDRU2 = 0;
            }
        }
    }
}

/*****************************************************************************************************************/
/*  Beep Sound Output                                                                                            */
/*****************************************************************************************************************/
void beep( unsigned char beep_no )
{
    unsigned char   i,j,keydt,sound_cnt,tmp;
    unsigned short  *sound_dt;
    union REG{
        unsigned short  WORD;
        struct {
            unsigned char   H;
            unsigned char   L;
        }BYTE;
    };
    union REG pwmdt;
```

```
switch ( beep_no ){
    case 0:                                     /* Select Beep Sound 0              */
        sound_dt = sound_0;
        sound_cnt = 19;
        break;

    case 1:                                     /* Select Beep Sound 1              */
        sound_dt = sound_1;
        sound_cnt = 17;
        break;

    case 2:                                     /* Select Beep Sound 2              */
        sound_dt = sound_2;
        sound_cnt = 15;
        break;

    case 3:                                     /* Select Beep Sound 3              */
        sound_dt = sound_3;
        sound_cnt = 14;
        break;

    case 4:                                     /* Select Beep Sound 4              */
        sound_dt = sound_4;
        sound_cnt = 12;
        break;

    case 5:                                     /* Select Beep Sound 5              */
        sound_dt = sound_5;
        sound_cnt = 11;
        break;

    case 6:                                     /* Select Beep Sound 6              */
        sound_dt = sound_6;
        sound_cnt = 10;
        break;

    case 7:                                     /* Select Beep Sound 7              */
        sound_dt = sound_7;
        sound_cnt = 9;
        break;

    default:                                    /* Select Beep Sound Error?         */
        sound_cnt = 0;
        break;
}
```

```
    TCRF   = 0x60;                                  /* Initialize Clock Select                */
    TCSRF  = 0x10;                                  /* Initialize Overflow Interrupt          */
    OCRFH  = 0x80;                                  /* Initialize Compare Match FH Value      */

    do{
        keydt = PDR5L;
        for(j = 0; j < sound_cnt; j++){
            pwmdt.WORD = sound_dt[j];
            PWDRL2 = pwmdt.BYTE.L;                  /* Set PWM Output Data                    */
            PWDRU2 = pwmdt.BYTE.H;
            do{
                tmp = CMFH;
            }while( tmp == 0);
            TCSRF  = 0x10;                          /* Initialize Overflow Interrupt          */
        }
    }while(keydt ! = 0x0F);                         /* Touch Key?                             */
}

/*****************************************************************************************************************/
/*  KeyRead                                                                                                      */
/*****************************************************************************************************************/
unsigned char keyread( void )
{
    return(PDR5L);
}


#pragma section PCMDT
unsigned short   sound_0[19] =
{
527,
627,
716,
785,
827,
836,
812,
758,
679,
584,
482,
386,
304,
247,
220,
224,
262,
328,
415,
};
```

```
unsigned short   sound_1[17] =
{
527,
639,
735,
804,
836,
826,
776,
692,
587,
473,
367,
282,
230,
220,
247,
314,
410,
};

unsigned short   sound_2[15] =
{
527,
653,
757,
821,
835,
796,
711,
594,
466,
348,
260,
220,
231,
293,
396,
};

unsigned short   sound_3[14] =
{
527,
661,
769,
829,
829,
770,
662,
528,
393,
285,
225,
224,
284,
391,
};
```

```
unsigned short   sound_4[12] =
{
527,
682,
795,
835,
796,
684,
530,
375,
260,
220,
256,
367,
};


unsigned short   sound_5[11] =
{
527,
695,
809,
830,
761,
614,
440,
293,
220,
245,
359,
};



unsigned short   sound_6[10] =
{
527,
709,
821,
822,
711,
530,
348,
233,
231,
341,
};
```

```
unsigned short   sound_7[9] =
{
527,
725,
832,
797,
638,
427,
262,
220,
320,
};
```

Link address specifications

Section Name	Address
CV1	0x0000
P, C,D,DPCMDT	0x0100

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.