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 Musical Notes

Introduction

A keyboard is connected to the parallel port, and the musical notes corresponding to pressed keys are output from a speaker.

Target Device

H8/38024

Contents

1. Specifications

1. Figure 1.1 shows the hardware configuration for an example of connecting musical-note output circuitry. As shown in the figure, a keyboard is connected to the I/O port (pins P70 to P76).
2. A square wave of the frequency corresponding to the key pressed on the keyboard is output from the speaker connected to the I/O port (pin P32).

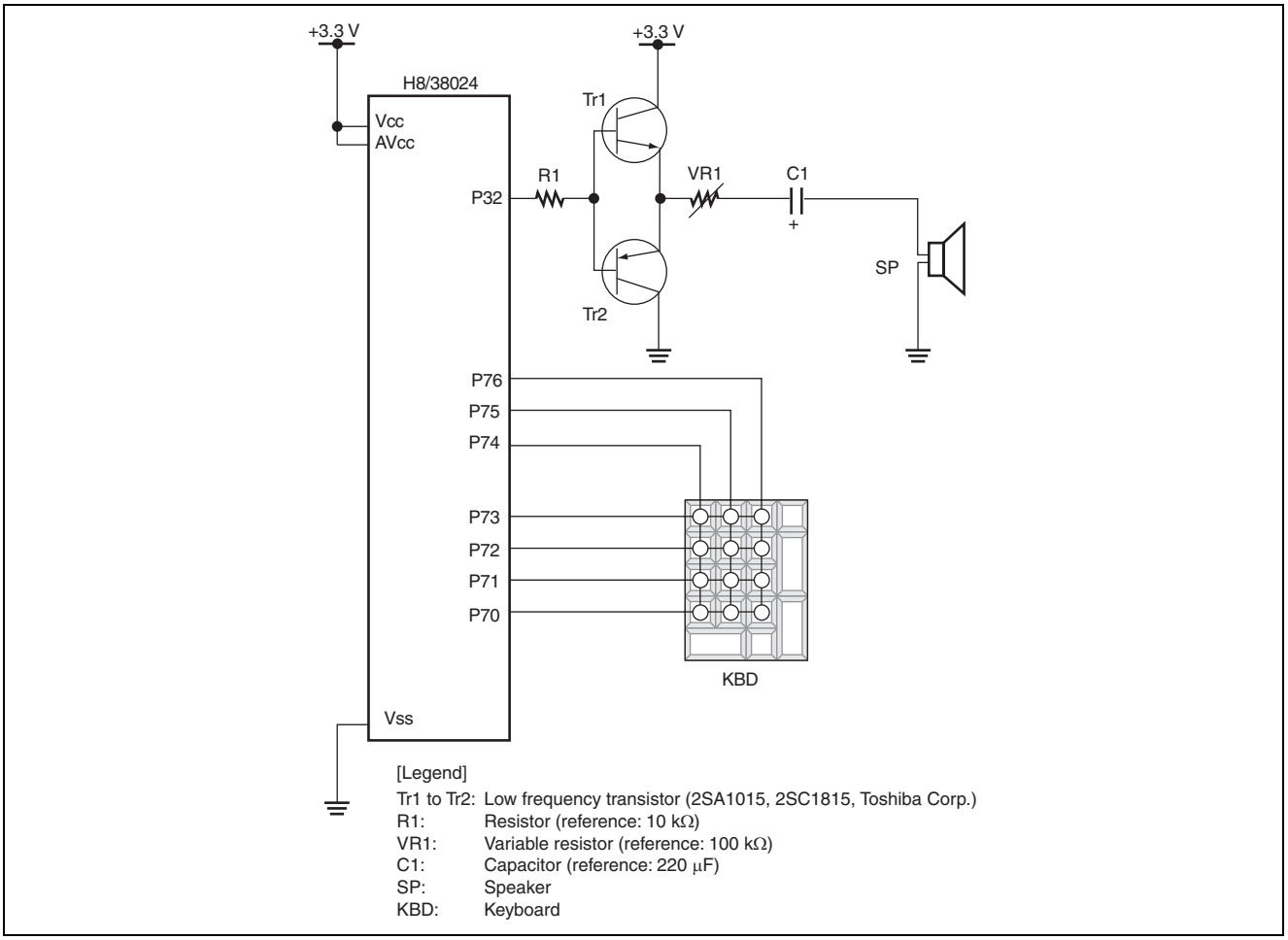


Figure 1.1 Hardware Configuration

3. In this sample task, the H8/38024's operating voltage (Vcc) and analog power supply voltage (AVcc) are 3.3 V, the OSC clock frequency is 10 MHz, and the watch clock is 32.768 kHz.
4. The keyboard used in this sample task is a numerical keypad used for input to a personal computer. Normally a 1-bit serial signal is output, but in this example the keyboard is modified to enable parallel I/O.
5. Operation of this sample task is as follows:
   A. Of the seven ports connected to the keyboard, P70 to P73 are sequentially switched to set to H level.
   B. Then the states of the remaining three ports P74 to P76 are monitored, and a pressed key is identified through the ports at which H level is detected.
   C. The frequency corresponding to the pressed key is output to port P32. The waveform is a rectangular waveform at H level.
   D. For example, if "1" on the numerical keypad is pressed, the combination of ports P70 and P74 causes the frequency 261.63 Hz of the note "do" to be output.
   E. The volume is adjusted using the variable resistor.

2. Description of Functions

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

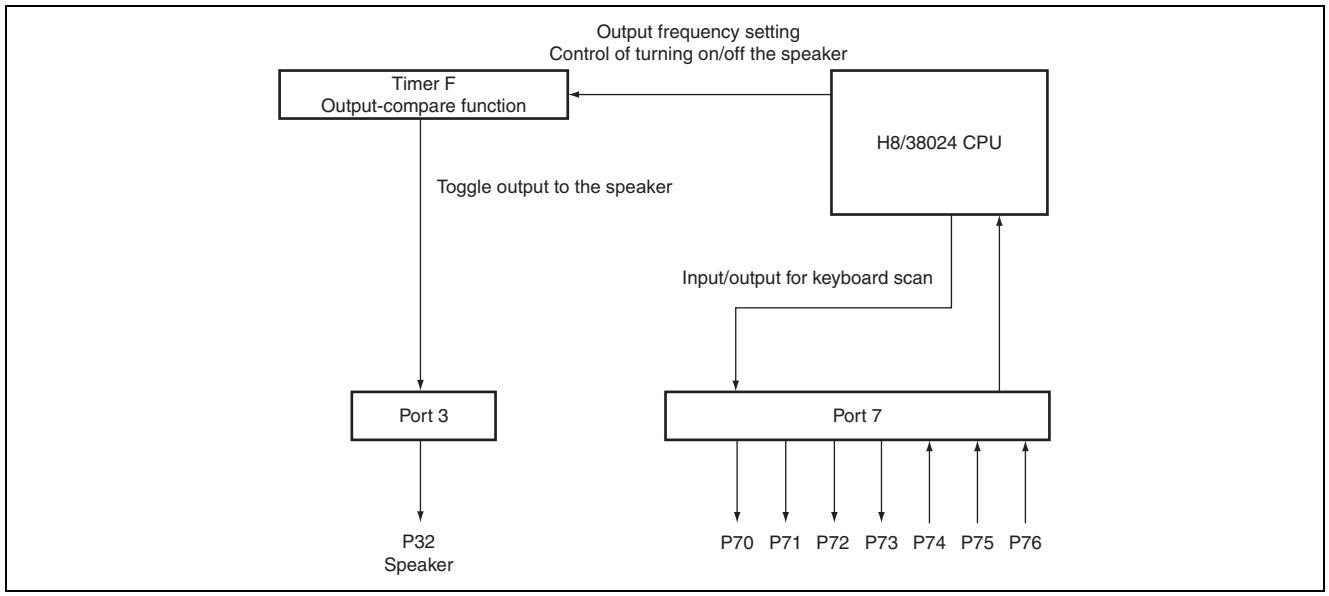


Figure 2.1 Block Diagram of Functions Used

Table 2.1 Function Allocation

Function	Function Allocation
Timer F	A compare-match function is used to toggle output. By setting the value in the 16-bit output-compare register (OCRF), the output frequency is changed.
Port 3	The frequency is output from P32 of port 3.
Port 7	Successive switching to the H level is effected through the port 7 output pins P70 to P73, and the key input is detected with the input pins P74 to P76.

3. Principle of Operation

1. The principle of operation for producing a musical note output using the timer F is shown in figure 3.1. As illustrated in this figure, TMOFH output is toggled in the cycle of compare-match of timer F to output the sound of A (440 Hz).

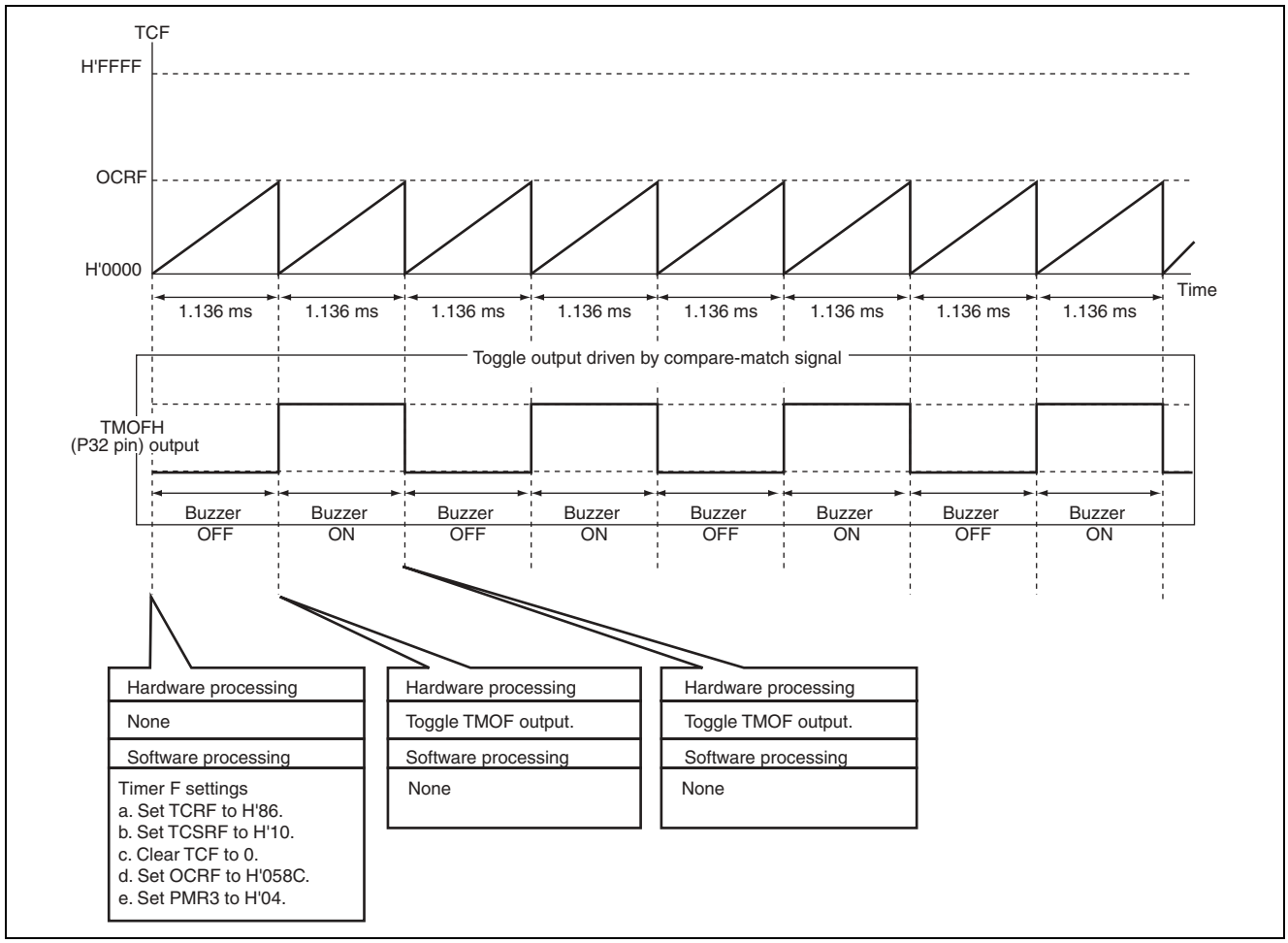


Figure 3.1 Operation Principle for Musical Note Output Using Timer F

2. Timer F settings
   — Internal clock settings
   The system clock is ϕ = 5 MHz, and the internal clock ϕ/4 (= 1250000 Hz) is selected.
   — Counter values set in OCRF

Table 3.1 Description of Counter Values

Note	Counter Value	Counter Value (Hex.)	Output Frequency	Calculated Frequency
C	2389	0955	261.62	261.63
C#	2255	08CF	277.16	277.18
D	2128	0850	293.70	293.66
D#	2009	07D9	311.10	311.13
E	1896	0768	329.64	329.63
F	1790	06FE	349.16	349.23
F#	1689	0699	370.04	369.99
G	1594	063A	392.10	392.00
G#	1505	05E1	415.28	415.30
A	1420	058C	440.14	440.00
A#	1341	053D	466.07	466.16
B	1265	04F1	494.07	493.88

[Legend]
Musical notes: C (do) to B (ti)
Counter value: (operating clock/2)/calculated frequency
Counter value (Hex.): The above counter value in hexadecimal
Output frequency: (operating clock/2)/counter value
Calculated frequency: Calculated frequency of each musical note.
Note: *The operating clock is divided by 2 because the ON/OFF action makes one cycle.

   — Calculated frequency
   The frequency for each note is calculated using the following formula:
   Calculated frequency = 440 × 2^(X/12)
   Example: Calculate to two decimal places.
   For the note of A: 440 × 2^(0/12) = 440 Hz
   For the note of A#: 440 × 2^(1/12) = 466.16 Hz
   For the note of G#: 440 × 2^(-1/12) = 415.30 Hz
   — Counter values
   Calculation of the counter values that are to be set in OCRF.
   Counter value = (operating clock/2)/calculated frequency.
   Example: Calculate as integers.
   For the note of A: 1250000 / 2 / 440 = 1420
   For the note of A#: 1250000 / 2 / 466.16 = 1341
   For the note of G#: 1250000 / 2 / 415.30 = 1505
   — Output frequency
   Calculation of the frequency to be output from the speaker.
   Output frequency = (operating clock/2)/counter value.
   Example: Calculate to two decimal places.
   For the note of A: 1250000 / 2 / 1420 = 440.14
   For the note of A#: 1250000 / 2 / 1341 = 466.07
   For the note of G#: 1250000 / 2 / 1505 = 415.28

3. Keyboard input

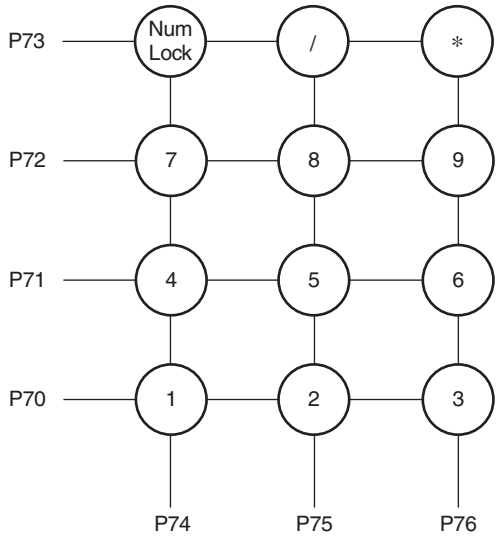


Figure 3.2 Example of Keyboard Connection

Table 3.2 Correspondence Between the Output Musical Note and the Key

Musical Note	Key
C	1
C#	2
D	3
D#	4
E	5
F	6
F#	7
G	8
G#	9
A	Num Lock
A#	/
B	*

P70 to P73 are sequentially switched to be driven to H level, and the states of P74 to P76 are monitored; the pressed key is identified from the ports at which H level is detected. The correspondence between the output musical note and the key pressed is indicated in table 3.2.

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, outputs frequencies.
Keyboard input processing routine	input_key	Scans keyboard and returns the corresponding code. If no key is pressed: −1 If a key is pressed: 0 to 11

4.2 Arguments

This sample task uses no arguments.

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
TCRF		Timer Control Register F Switches between 16-bit mode and 8-bit mode, selects an internal clock from four clocks and the external event, and sets output level for the TMOFH and TMOFL pins.	H'FFB6	H'06 (Initial setting)
	TOLH	Toggle Output Level H Sets TMOFH pin output level. When TOLH = 0, it is set to low.	Bit 7	0
	CKSH2	Clock Select H	Bit 6	0
	CKSH1	With CKSH2 = 0, CKSH1 = 0 and CKSH0 = 0, 16-bit mode is selected and counting is of the TCFL overflow signal.	Bit 5	0
	CKSH0		Bit 4	0
	TOLL	Toggle Output Level L Sets TMOFL pin output level. When TOLL = 0, it is set to low.	Bit 3	0
	CKSL2	Clock Select L	Bit 2	1
	CKSL1	With CKSL2 = 1, CKSL1 = 1 and CKSL0 = 0, counting is of the internal clock $\phi/4$.	Bit 1	1
	CKSL0		Bit 0	0

Register		Function	Address	Setting
TCSRF		Timer Control/Status Register F Selects counter clearing method, sets overflow flag, sets compare-match status flag, and controls enable/disable of interrupt requests output in response to an overflow.	H'FFB7	H'10
	OVFH	Timer Overflow Flag H Status flag to indicate that TCFH has overflowed (H'FF → H'00).	Bit 7	0
	CMFH	Compare Match Flag H Status flag to indicate that the values of TCFH and OCRFH match.	Bit 6	0
	OVIEH	Timer Overflow Interrupt Enable H When OVIEH = 0, interrupt requests in response to a TCFH overflow are disabled.	Bit 5	0
	CCLRH	Counter Clear H When CCLRH = 1, TCF clearing on a compare match is enabled.	Bit 4	1
	OVFL	Timer Overflow Flag L Status flag to indicate that TCFL has overflowed (H'FF → H'00).	Bit 3	0
	CMFL	Compare Match Flag L Status flag to indicate that the values of TCFL and OCRFL match.	Bit 2	0
	OVIEL	Timer Overflow Interrupt Enable L When OVIEL = 0, interrupt requests in response to a TCFL overflow are disabled.	Bit 1	0
	CCLRL	Counter Clear L When CCLRL = 0, TCFL clearing on a compare match is enabled.	Bit 0	0
TCF		16-bit Timer Counter TCF is a 16-bit readable/writable up-counter.	H'FFB8	H'00
OCRF		16-bit Output Compare Register OCRF is compared with TCF and the output on the TMOFH pin is toggled upon a match.	H'FFBA	H'00 (initial setting)
CKSTPR1		Clock Stop Register 1 Controls module standby mode. When TFCKSTP = 0 (CKSTPR1 = H'FB), the timer F is in module standby mode. When TFCKSTP = 1 (CKSTPR1 = H'FF), the timer F module standby mode is canceled.	H'FFFA	H'FB (initial setting)

Register		Function	Address	Setting
PMR3		Port Mode Register 3 Sets functions for the pins of port 3.	H'FFCA	H'04
	AEVL	P37/AEVL Pin Function Switching When AEVL = 0, this pin functions as P37 I/O pin.	Bit 7	0
	AEVH	P36/AEVH Pin Function Switching When AEVH = 0, this pin functions as P36 I/O pin.	Bit 6	0
	TMOFH	P32/TMOFH Pin Function Switching When TMOFH = 1, this pin functions as TMOFH output pin.	Bit 2	1
	TMOFL	P31/TMOFL Pin Function Switching When TMOFL = 0, this pin functions as P31 I/O pin.	Bit 1	0
	UD	P30/UD Pin Function Switching When UD = 0, this pin functions as P30 I/O pin.	Bit 0	0
PDR7		Port data register 7 General-purpose I/O port data register for port 7.	H'FFDA	H'00 (initial setting)
PCR7		Port control register 7 Provides pin-by-pin control of input/output selection for the pins of port 7 that have been set as general purpose I/O pins. When PCR7 = H'0F, the pins P77 to P74 function as general-purpose input pins and P73 to P70 as general-purpose output pins.	H'FFEA	H'0F

4.4 Description of RAM

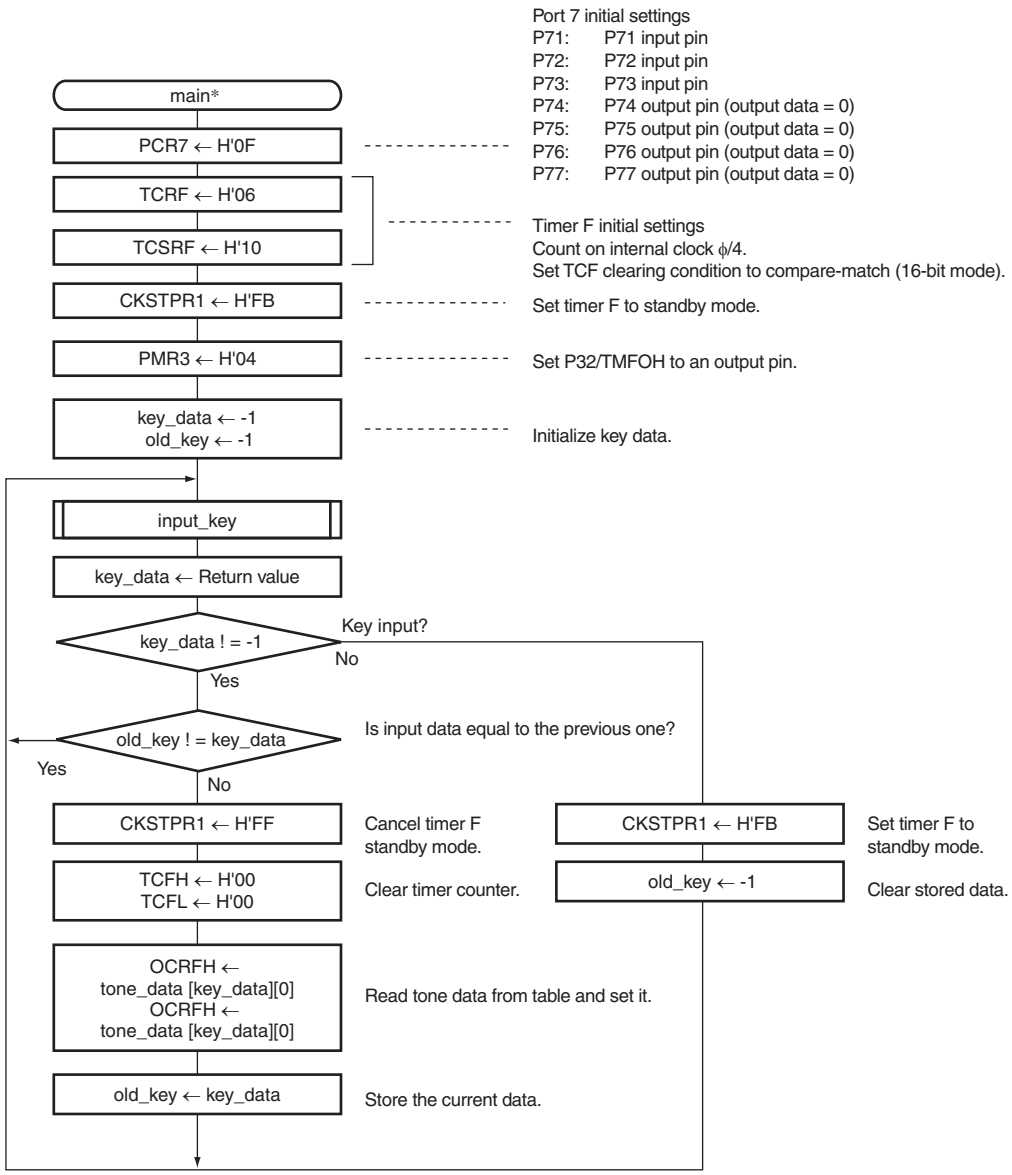
Table 4.3 describes the RAM used in this sample task.

Table 4.3 Description of RAM

Label	Function	Address	Used in
key_data	Stores input key data. (2 bytes)	H'FB80	main
old_key	Stores previous key data. (2 bytes)	H'FB82	main
i	Stores a loop counter for key scans. (2 bytes)	H'FB84	input_key
j	Stores a loop counter for key scans. (2 bytes)	H'FB86	input_key
key	Key data for key scans (2 bytes)	H'FB88	input_key

5. Flowchart

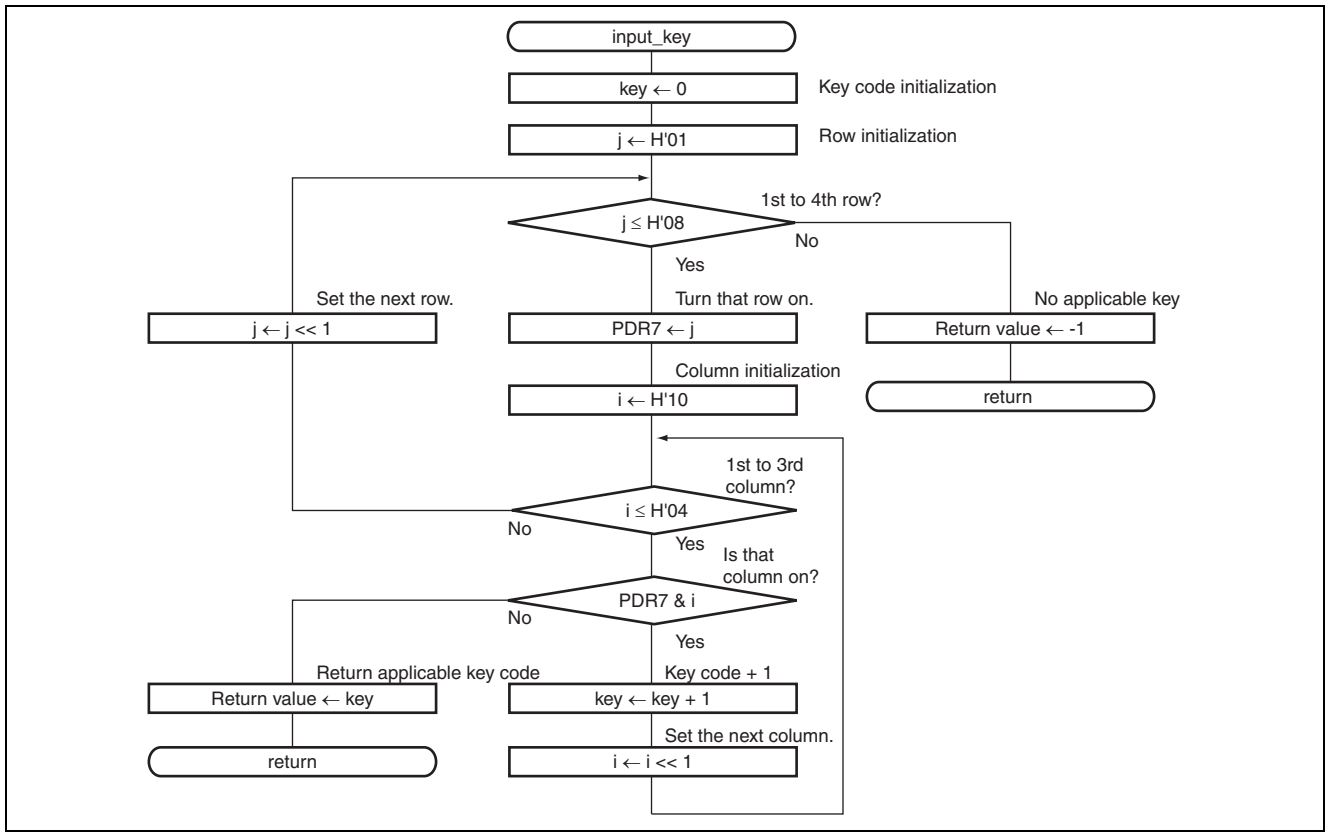
1. Main Routine (main)



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

Reading/writing to TCF and writing to OCRF in 16-bit mode should be performed in 16-bit units (execute two byte-size MOV instructions continuously), in which the upper byte should be accessed first and then the lower byte.

2. Keyboard Input Processing Routine (input_key)



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'                                                                                               */
/*  ' Outputting musical notes '                                                                                        */
/*                                                                                                                      */
/*  Function                                                                                                            */
/*  : Outputting musical notes                                                                                          */
/*                                                                                                                      */
/*  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    PMR3         *(volatile unsigned char *)0xFFCA        /* Port mode register 3                    */

#define    PDR7         *(volatile unsigned char *)0xFFDA        /* Port data register 7                    */
#define    PCR7         *(volatile unsigned char *)0xFFEA        /* Port control register 7                 */

#define    TCRF         *(volatile unsigned char *)0xFFB6        /* timer control register F                */
#define    TCSRF        *(volatile unsigned char *)0xFFB7        /* timer control status register F         */
#define    TCFH         *(volatile unsigned char *)0xFFB8        /* 8 bit timer counter F (HIGH)            */
#define    TCFL         *(volatile unsigned char *)0xFFB9        /* 8 bit timer counter F (LOW)             */
#define    OCRFH        *(volatile unsigned char *)0xFFBA        /* 8 bit output compare register F (HIGH)  */
#define    OCRFL        *(volatile unsigned char *)0xFFBB        /* 8 bit output compare register F (LOW)   */
#define    CKSTPR1      *(volatile unsigned char *)0xFFFA        /* clock stop register 1                   */
```

```
/*****************************************************************************************************************/
/*  Function Definition                                                                                          */
/*****************************************************************************************************************/
extern void INIT(void);                                 /* Stack pointer set                    */
void main(void);                                        /* main routine                         */
int input_key(void);                                    /* Input key                            */

const unsigned char tone_data[12][2] =
{
    0x09, 0x55,                                         /* C (261.63Hz)                         */
    0x08, 0xCF,                                         /* C#(277.18Hz)                         */
    0x08, 0x50,                                         /* D (293.66Hz)                         */
    0x07, 0xD9,                                         /* D#(311.13Hz)                         */
    0x07, 0x68,                                         /* E (329.63Hz)                         */
    0x06, 0xFE,                                         /* F (349.23Hz)                         */
    0x06, 0x99,                                         /* F#(369.99Hz)                         */
    0x06, 0x3A,                                         /* G (392.00Hz)                         */
    0x05, 0xE1,                                         /* G#(415.30Hz)                         */
    0x05, 0x8C,                                         /* A (440.00Hz)                         */
    0x05, 0x3D,                                         /* A#(466.16Hz)                         */
    0x04, 0xF1,                                         /* B (493.88Hz)                         */
};
/*****************************************************************************************************************/
/*  Data Table                                                                                                   */
/*****************************************************************************************************************/
int key_data, old_key;                                  /* Input key code New & Old             */
int i,j;                                                /* Loop counter                         */
int key;                                                /* Input key code                       */

/*****************************************************************************************************************/
/*  Vector Address                                                                                               */
/*****************************************************************************************************************/
#pragma section V1                                      /* Vector section set                   */
void (*const VEC_TBL1[])(void) = {
    INIT                                                /* 0x0000 Reset vector                  */
};
#pragma section                                         /* P                                    */
```

```
/*****************************************************************************************************************/
/*  Main Program                                                                                                 */
/*****************************************************************************************************************/
void main(void)
{
    PCR7 = 0x0F;                                        /* Initialize for input key                */

    TCRF = 0x06;                                        /* Set internal clock: phi/4               */
    TCSRF = 0x10;                                       /* Enable TCF clear (16 bit mode)          */
    CKSTPR1 = 0xfb;                                     /* clock F STAND-BY-MODE ON                */
    PMR3  = 0x04;                                       /* Set TMOFH output mode                   */

    key_data = old_key = -1;                            /* Initialize key data                     */
    while(1){
        key_data = input_key();                         /* Input key                               */
        if ( key_data != -1) {
            if ( old_key != key_data ) {
            CKSTPR1 = 0xff;                             /* Clock F STAND-BY-MODE OFF               */
            TCFH = 0x00;                                /* Clear Timer Counter A to 0 (HIGH)       */
            TCFL = 0x00;                                /* Clear Timer Counter A to 0 (LOW)        */
            OCRFH = tone_data[key_data][0];             /* Set tone data (HIGH)                    */
            OCRFL = tone_data[key_data][1];             /* Set tone data (LOW)                     */
            old_key = key_data;                         /* Save current key data                   */
            }
        }
        else {
        CKSTPR1 = 0xfb;                                 /* clock F STAND-BY-MODE ON                */
        old_key = -1;                                   /* Clear old input data                    */
        }
    }
}

/*****************************************************************************************************************/
/*  Input Key Data                                                                                               */
/*****************************************************************************************************************/
int input_key(void)
{
    key = 0;                                            /* Set start key code                      */
    for ( j = 0x01; j < = 0x08; j << = 1 ) {
        PDR7 = j;                                       /* Output level to line                    */
        for ( I = 0x10; I < = 0x40; I << = 1 ) {
            if ( PDR7 & i ) {                           /* Read level from line                    */
            return key;
            }
            key++;
        }
    }
    return -1;
}
```

Link address specifications

Section Name	Address
CV1	H'0000
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.