R8C/Tiny Series
Sample Program: 3-Minute Timer

1. Abstract

The 3-minute timer uses the following functions:

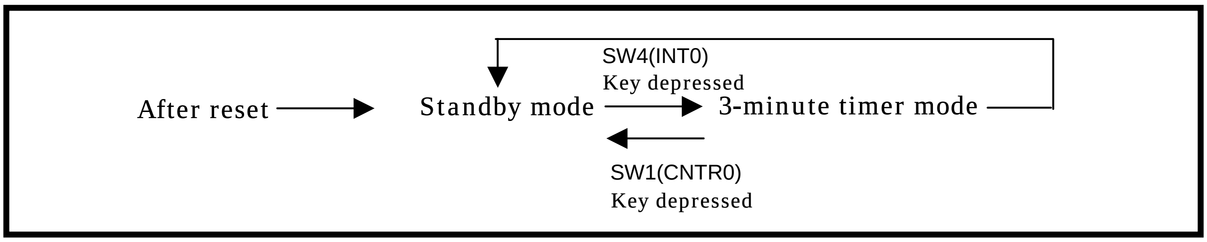
- Timer X (timer mode)
- Timer Z (programmable waveform generation mode)

2. Introduction

The example described in this application note is intended for use in the R8C/17 group of microcomputers.
This program can also be used when operating other microcomputers within the R8C/Tiny, provided they have the same SFR (Special Function Registers) as the R8C/17 microcomputers. However, since it is possible that some functions of the R8C/Tiny series will have been altered for functional enhancements, etc., please be sure to obtain the SFR header file from the Renesas Web site shown below.
http://www.renesas.com/en/r8ctiny

Operation (1) The 3-minute timer has two separate modes:

- Standby mode
- 3-minute timer mode



(2) After reset, the microcomputer goes to standby mode.
(3) If SW4 (INT0) is pressed during standby mode, the microcomputer goes to 3-minute timer mode.
(4) In 3-minute timer mode, the microcomputer executes a 3-minute count operation.
(5) When the microcomputer has finished counting 3 minutes, it sounds a buzzer two times and goes to standby mode. (Buzzer operation will be detailed later.)
(6) If SW1 (CNTR0) is pressed during 3-minute timer mode, the microcomputer goes to standby mode.
(7) Key input is sampled at 10 ms intervals, and is confirmed to have been entered when sampled low three times consecutively. (Chattering elimination)
(8) The key inputs SW4 (INT0) and SW1 (CNTR0) are active-low. (When low, the key is pressed; when high, not pressed.)
(9) During 3-minute timer mode, the status is indicated by LEDs according to the passage of time, as shown below.

LED indications for the passage of time:

	LED1 (red)	LED2 (green)	LED3 (green)	LED4 (green)
2 minutes or more remaining time	Off	Off	On	On
1 minute or more remaining time	Off	Off	Flashes every 500 ms	Flashes every 500 ms
10 seconds or more remaining time	Off	Off	Flashes every 500 ms	Off
10 seconds or less remaining time	Off	Off	Flashes every 250 ms	Off
0 second remaining time (buzzer sounded)	On	Off	Off	Off

(10) During standby mode, the status is indicated by LEDs, as shown below.

LED indications during standby mode:

	LED1 (red)	LED2 (green)	LED3 (green)	LED4 (green)
During standby mode	On	Off	Off	Off

(11) The LED1, LED2, LED3 and LED4 are active-low. (When low, the LED lights; when high, the LED goes out.)
(12) The buzzer on/off signal is output from P13 (TZOUT) using timer-Z programmable waveform mode.
(13) The buzzer is output with 1 kHz (50% duty cycle), repeatedly turned on and off twice at 200 ms intervals, and at the end of the sequence, turned off again for another 200 ms.

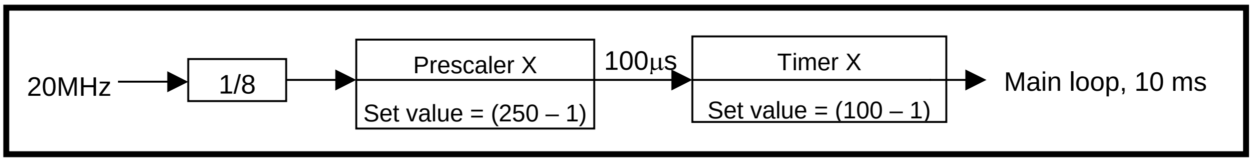
Buzzer output timing sequence:

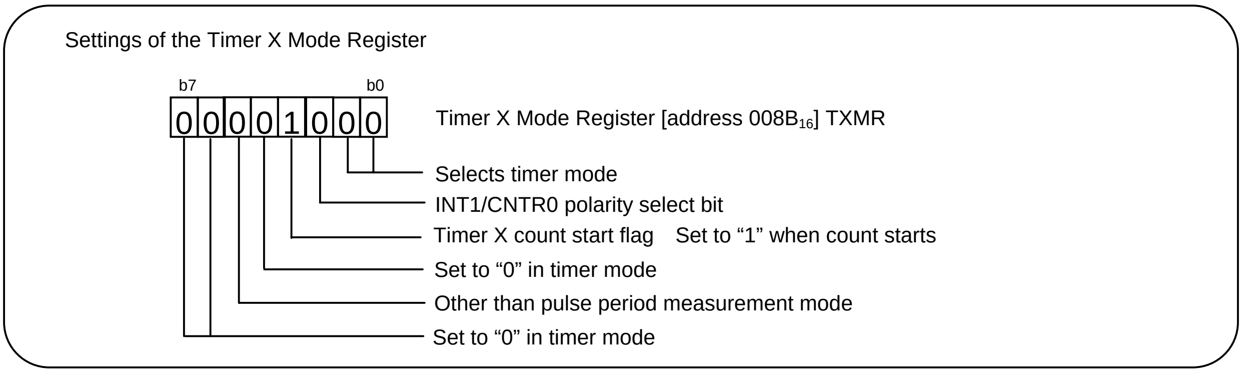
Sequence No.	1	2	3	4
Buzzer status	ON (output)	OFF (stop)	ON (output)	OFF (stop)
Output and stop time	200 ms	200 ms	200 ms	400 ms

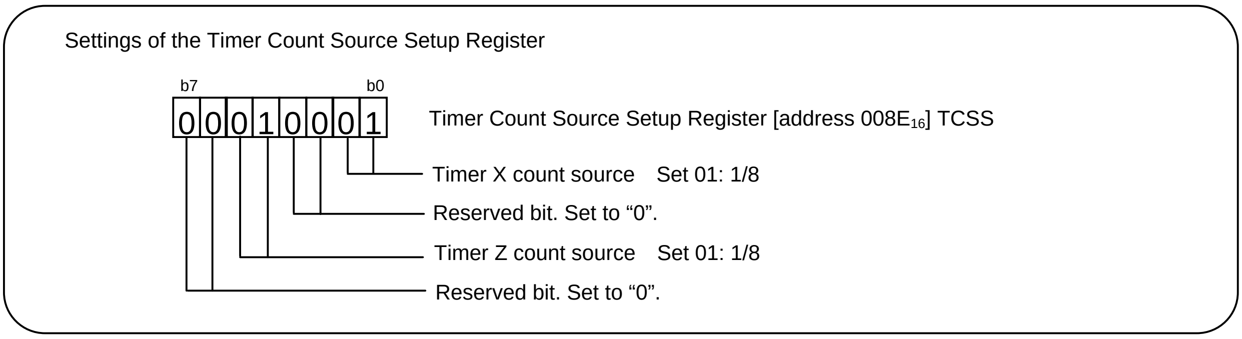
3. About the Program

3.1 Timer X (Timer Mode)

The timer X (timer mode) is used to maintain a constant main loop period.
The set value of the timer X is shown below.



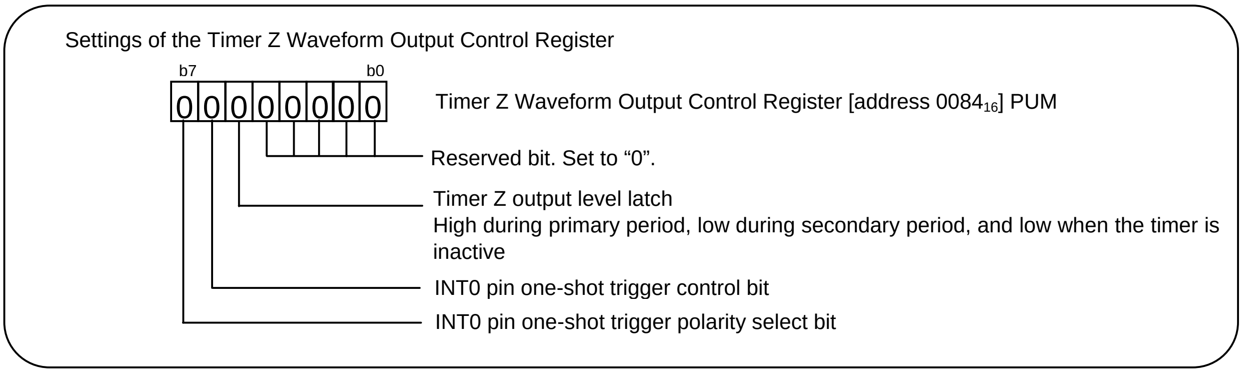

Settings of the Timer X Mode Register
b7
b0
0 0 0 0 1 0 0 0
Timer X Mode Register [address 008B16] TXMR
Selects timer mode
INT1/CNTR0 polarity select bit
Timer X count start flag Set to “1” when count starts
Set to “0” in timer mode
Other than pulse period measurement mode
Set to “0” in timer mode


Settings of the Timer Count Source Setup Register
b7
b0
0 0 0 1 0 0 0 1
Timer Count Source Setup Register [address 008E16] TCSS
Timer X count source Set 01: 1/8
Reserved bit. Set to “0”.
Timer Z count source Set 01: 1/8
Reserved bit. Set to “0”.

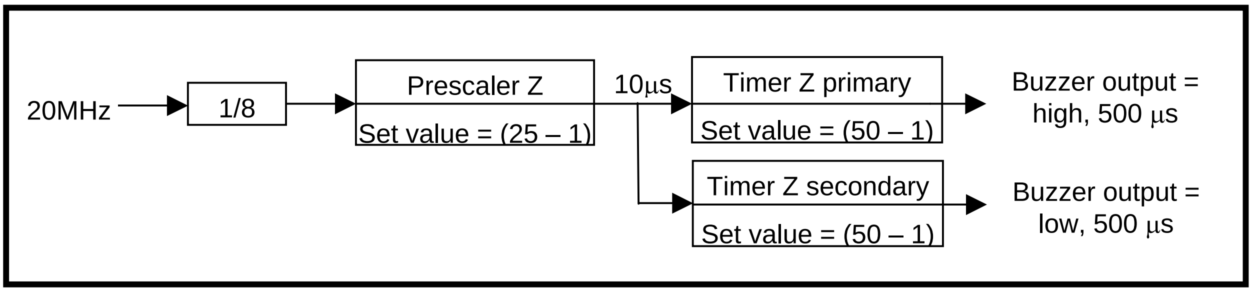
When the respective registers are set up as shown above, the timer X interrupt request bit is cyclically set at 10 ms intervals.
Before the main processing, the program checks the timer X interrupt request bit to see if 10 ms has elapsed. If 10 ms has elapsed, the program clears the timer X interrupt request bit and executes the main processing. If 10 ms has not elapsed yet, the program waits until the timer X interrupt request bit is set.

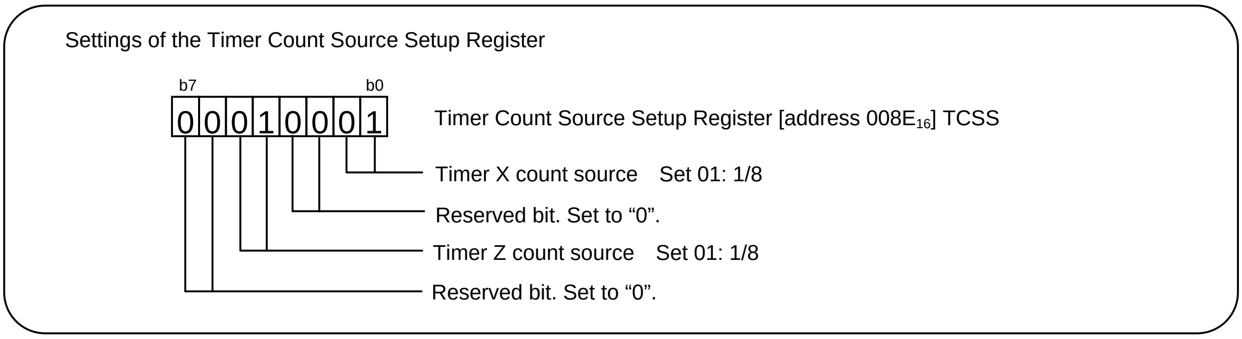
3.2 Timer Z (Programmable Waveform Generation Mode)

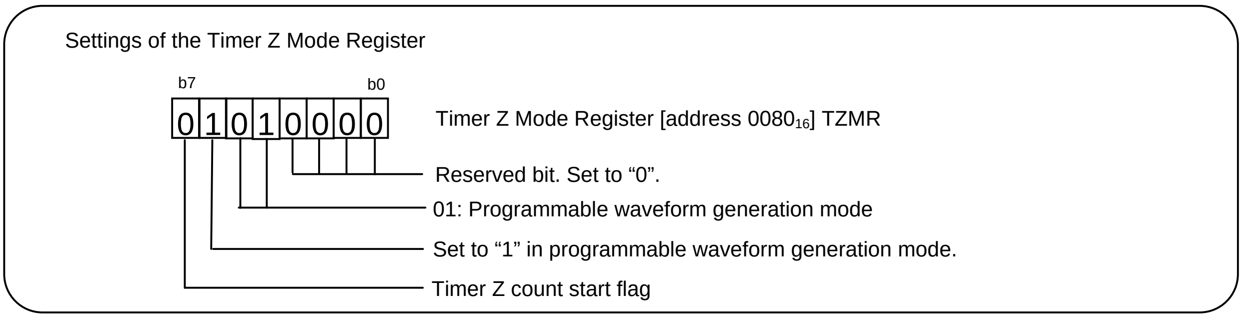
The timer Z (programmable waveform generation mode) is used to sound a buzzer.
Waveform generation with 1 kHz (50% duty cycle) is accomplished by setting up the registers as shown below. The output level is set by using the timer Z output level latch (TZOPL) in the Timer Z Waveform Output Control Register (PUM). Setting the timer Z output level latch (TZOPL) to “0” causes the device to output a high during the primary period, a low during the secondary period, and a low when the timer is inactive.


Settings of the Timer Z Waveform Output Control Register
b7
b0
0 0 0 0 0 0 0 0
Timer Z Waveform Output Control Register [address 0084₁₆] PUM
Reserved bit. Set to “0”.
Timer Z output level latch
High during primary period, low during secondary period, and low when the timer is inactive
INT0 pin one-shot trigger control bit
INT0 pin one-shot trigger polarity select bit

When the timer Z starts counting, the device outputs a high during the primary period and a low during the secondary period from P13 (TZOUT), and when the timer Z stops counting, the device outputs a low. The timer Z set values are shown below.




Settings of the Timer Count Source Setup Register
b7
b0
0 0 0 1 0 0 0 1
Timer Count Source Setup Register [address 008E₁₆] TCSS
Timer X count source Set 01: 1/8
Reserved bit. Set to “0”.
Timer Z count source Set 01: 1/8
Reserved bit. Set to “0”.

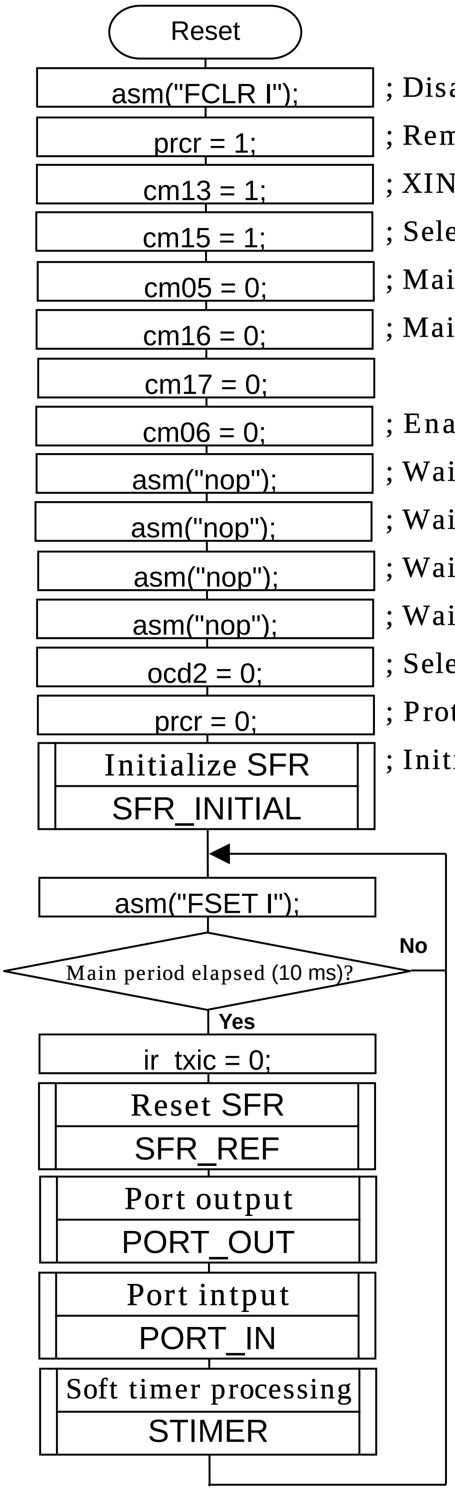

Settings of the Timer Z Mode Register
b7
b0
0 1 0 1 0 0 0 0
Timer Z Mode Register [address 0080₁₆] TZMR
Reserved bit. Set to “0”.
01: Programmable waveform generation mode
Set to “1” in programmable waveform generation mode.
Timer Z count start flag

3.3 Chattering Elimination

To protect the key input against chattering noise, SW1 and SW4 are read multiple times to confirm that the detected input levels are the same. For the 3-minute timer, key input is sampled at 10 ms intervals, and is confirmed to have been entered when sampled low three times consecutively.

4. Flowchart

4.1 Initial Settings and Main Loop



; Disables interrupts
; Removes protection of the system control registers
; XIN-XOUT pin
; Selects XIN-XOUT drive capability; set to HIGH
; Main clock oscillation
; Main clock not divided
; Enables main clock division CM16, CM17
; Wait until stabilizes
; Wait until stabilizes
; Wait until stabilizes
; Wait until stabilizes
; Selects the main clock
; Protects the system control registers
; Initializes the SFR (port initialization and timer setup)

; Enables interrupts
; Wait for timer X request

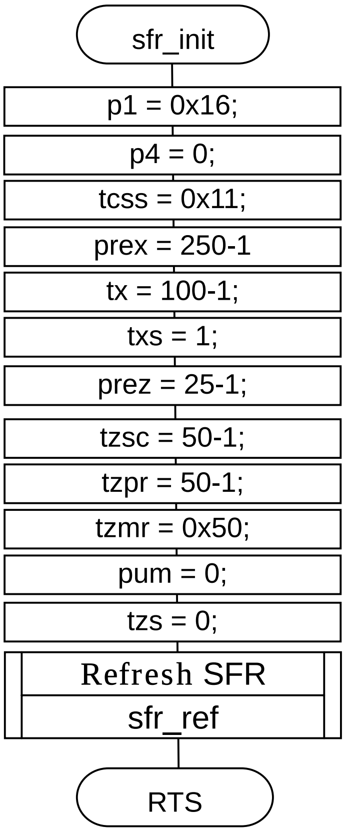
; Clears the timer X request flag
; Refreshes the port direction register

; Sets the port for output (LED and buzzer)

; Sets the port for input (key input processing)

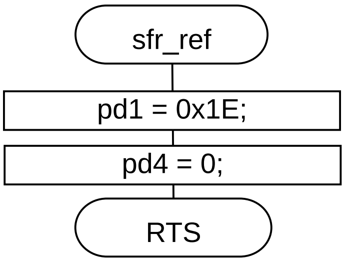
; Soft timer (10 ms based timer value subtraction)

4.2 SFR Initialization



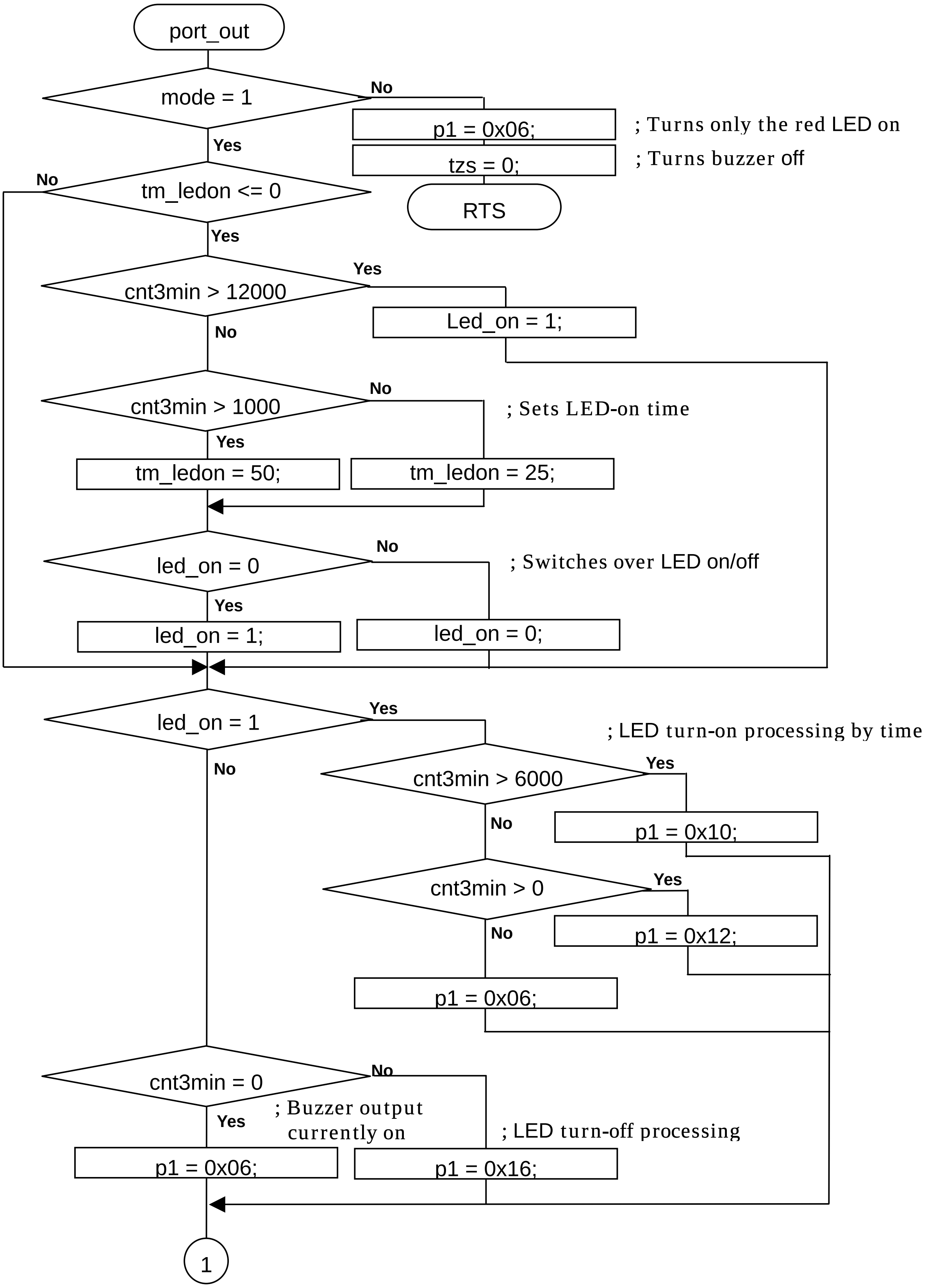
; Initializes port P1
; Initializes port P4
; Timer X = divided-by-8, timer Z = divided-by-8
; Prescaler X = 250 –1
; Timer X = 100 – 1
; Timer X starts counting
; Prescaler Z = 25 – 1
; Timer Z secondary register = 50 – 1
; Timer Z primary register = 50 – 1
; Timer Z mode = programmable waveform generation mode
; Timer Z waveform output control register = 0
; Timer Z stops counting
; Refreshes the SFR

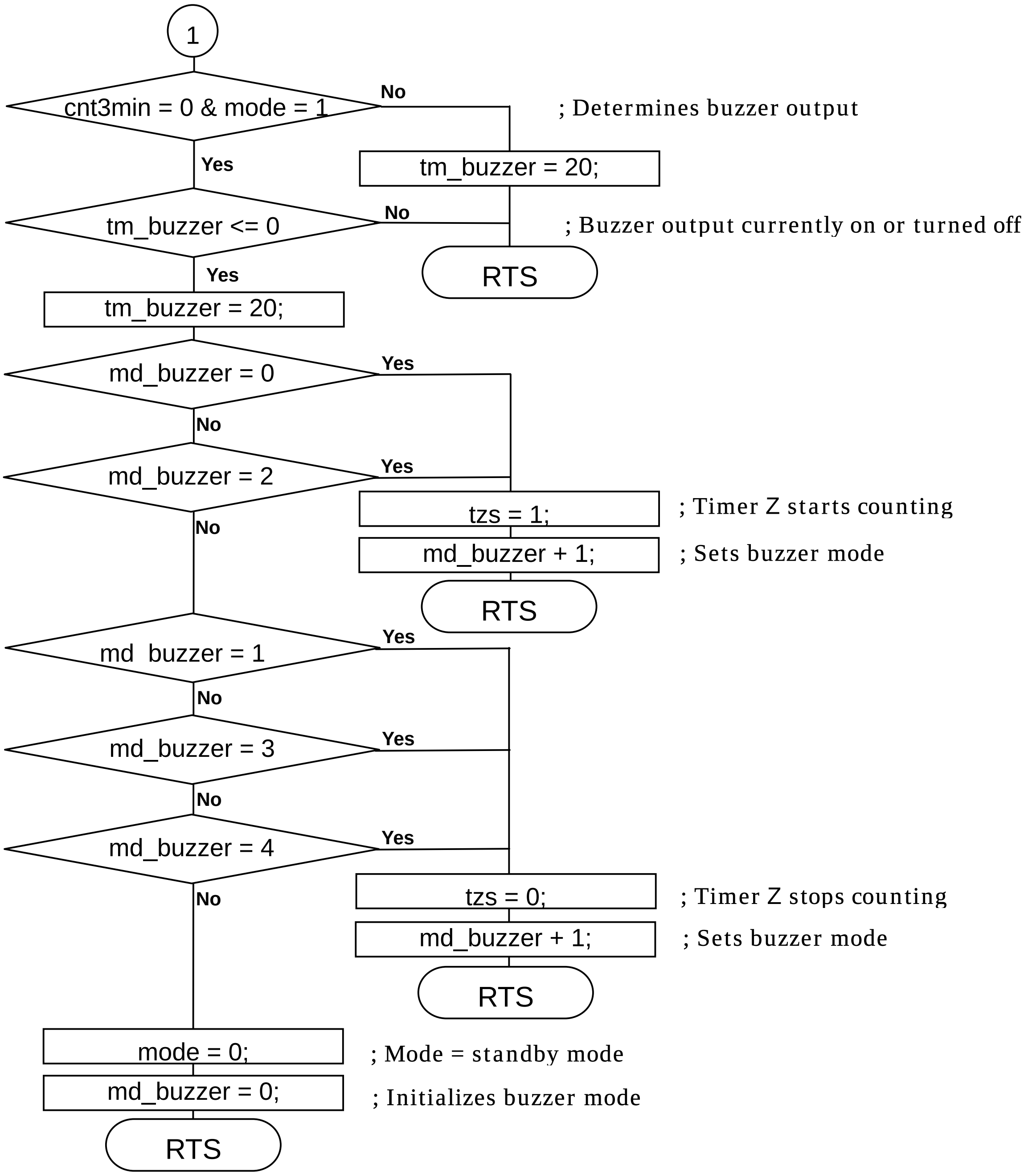
4.3 SFR Refresh



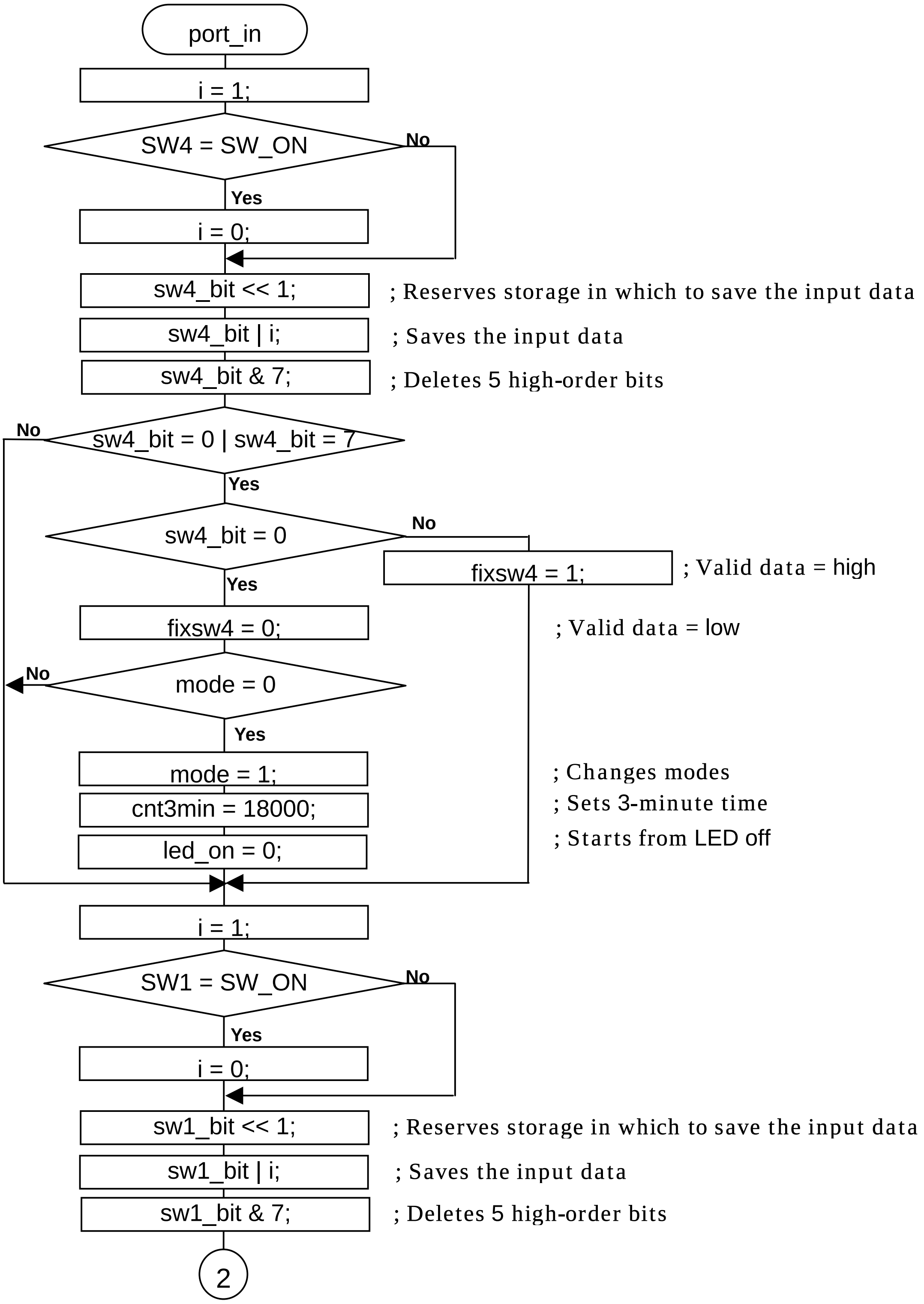
; Initializes/refreshes the port P1 direction register
; Initializes/refreshes the port P4 direction register

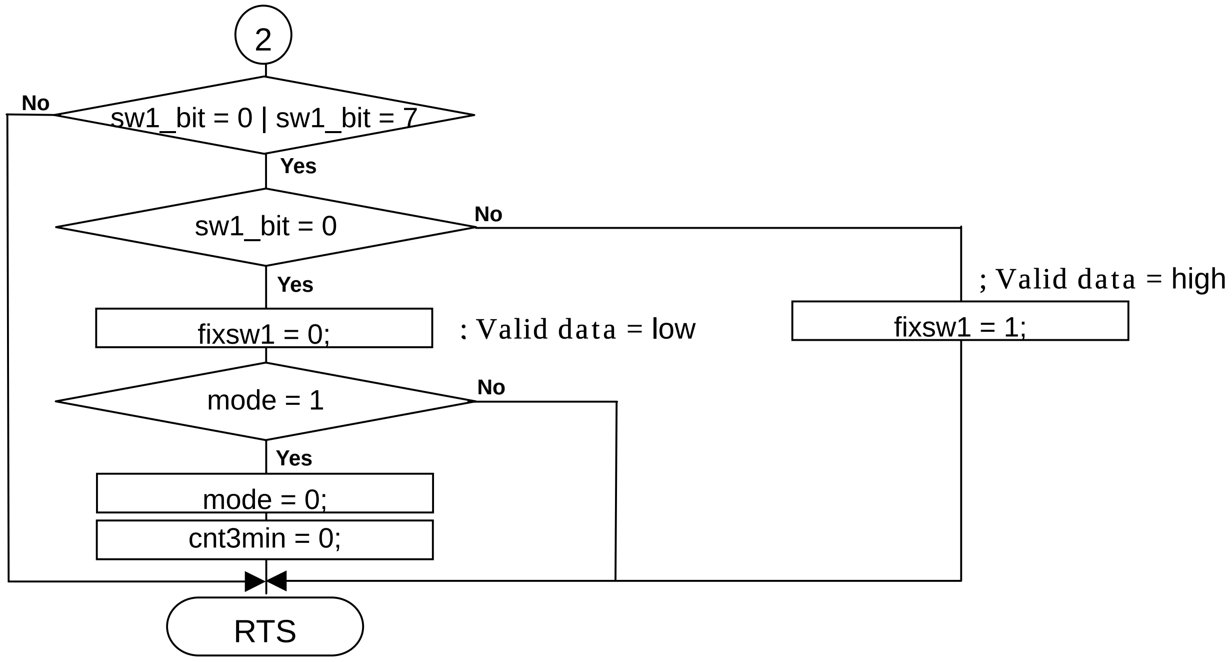
4.4 Port Output



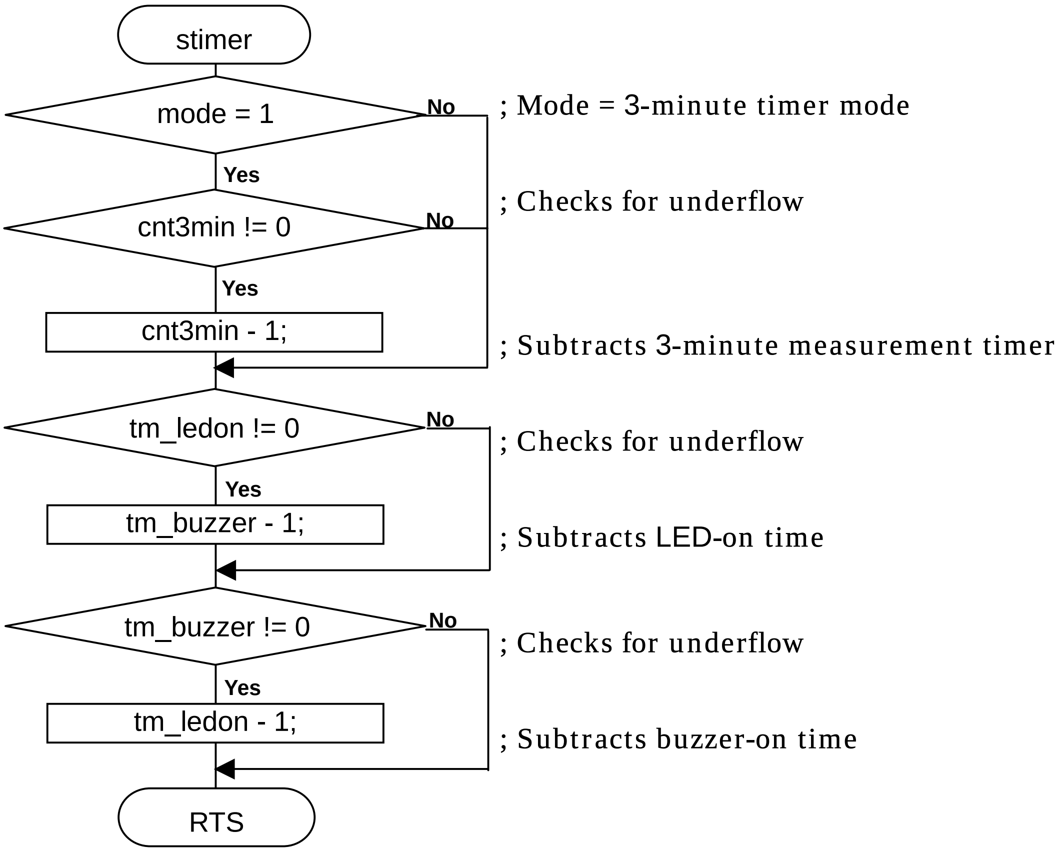

1
cnt3min = 0 & mode = 1
No
; Determines buzzer output
tm_buzzer = 20;
Yes
tm_buzzer <= 0
No
; Buzzer output currently on or turned off
RTS
Yes
tm_buzzer = 20;
md_buzzer = 0
Yes
No
md_buzzer = 2
Yes
tzs = 1;
; Timer Z starts counting
No
md_buzzer + 1;
; Sets buzzer mode
RTS
md buzzer = 1
Yes
No
md_buzzer = 3
Yes
No
md_buzzer = 4
Yes
No
tzs = 0;
; Timer Z stops counting
md_buzzer + 1;
; Sets buzzer mode
RTS
mode = 0;
; Mode = standby mode
md_buzzer = 0;
; Initializes buzzer mode
RTS

4.5 Port Input




2
sw1_bit = 0 | sw1_bit = 7
No
Yes
sw1_bit = 0
No
Yes
fixsw1 = 0;
: Valid data = low
; Valid data = high
fixsw1 = 1;
mode = 1
No
Yes
mode = 0;
cnt3min = 0;
RTS

4.6 Soft Timer Processing



5. Sample Program

```
/************************************************************************************
*                                                                                  *
*  File Name   : 3min_timer.h                                                      *
*  Contents    : definition of R8C/17 Group SFR                                    *
*  Copyright, 2004 RENESAS TECHNOLOGY CORPORATION                                  *
*                AND RENESAS SOLUTIONS CORPORATION                                 *
*  Version     : 1.00                                                              *
*  note       :                                                                    *
*                                                                                  *
************************************************************************************/

/* Definition of RAM area */

char mode           = 0;                    /* Mode number */
char sw1_bit        = 0;                    /* Input SW1 data */
char fixsw1         = 1;                    /* Input SW1 Settlement data */
char sw4_bit        = 0;                    /* Input SW4 data */
char fixsw4         = 1;                    /* Input SW4 Settlement data */

char md_buzzer;                             /* Buzzer control mode */
char led_on;                                /* Display on-off */
unsigned int tm_ledon;                      /* LED control timer */
unsigned int tm_buzzer;                     /* Buzzer control timer */
unsigned long int cnt3min;                  /* Counting area in the 3 minute timer */

/* Declaration of function prototype */

extern void sfr_init(void);                 /* Initial setting of SFR registers */
extern void sfr_ref(void);                  /* Refresh of SFR registers */
extern void port_out(void);                 /* Port output */
extern void port_in(void);                  /* Port input */
extern void stimer(void);                   /* Software timer */

/* Definition of base section */

#define SW_ON       0
#define SW_OFF      1
#define LOW         0
#define HIGH        1

/* Definition of port */

#define GREEN_LED2  p1_1
#define GREEN_LED1  p1_2
#define GREEN_LED0  p1_3
#define RED_LEDp1_4
#define SW1         p1_7
#define SW4         p4_5
#define BUZZER      p1_3
```

```
/**************************************************************************************
*                                                                                    *
*   File Name   : main.c                                                             *
*   Contents    : definition of R8C/17 Group SFR                                     *
*   Copyright, 2004 RENESAS TECHNOLOGY CORPORATION                                   *
*                 AND RENESAS SOLUTIONS CORPORATION                                  *
*   Version     : 1.00                                                               *
*   note        :                                                                    *
*                                                                                    *
**************************************************************************************/
#include "sfr_r817.h"                   /* Definition of the R8C/17 SFR */
#include "3min_timer.h"                 /* Definition of processing for 3 minutes timer */

main(){
     asm("FCLR I");                     /* Interrupt disable */
     prcr = 1;                          /* Protect off */
     cm13 = 1;                          /* X-in X-out */
     cm15 = 1;                          /* XCIN-XCOUT drive capacity select bit : HIGH */
     cm05 = 0;                          /* X-in on */
     cm16 = 0;                          /* Main clock = No division mode */
     cm17 = 0;
     cm06 = 0;                          /* CM16 and CM17 enable */
     asm("nop");
     asm("nop");
     asm("nop");
     asm("nop");
     ocd2 = 0;                          /* Main clock change */
     prcr = 0;                          /* Protect on */

     sfr_init();                        /* Initial setting of SFR registers */

     while(1){                          /* Main processing */
          asm("FSET I");                /* Interrupt enable */
          while(ir_txic == 0){}         /* Main cycle 10ms */
          ir_txic = 0;
          sfr_ref();                    /* Refreshment of SFR registers */
          port_out();                   /* Port output */
          port_in();                    /* Port input */
          stimer();                     /* Software timer */
     }
}
```

```
/************************************************************************************
Name:       sfr_init
Parameters:  None
Returns:    None
Description: Initial setting of SFR registers
************************************************************************************/
void sfr_init(void){
     /* Setting port registers */
     p1 = 0x16;                         /* p14,12,11 = H(Led 4.-.2.1) P13 = L(Buzzer) */
     p4 = 0;
     tcss = 0x11;                       /* division = X:1/8,Z:1/8 */
     /* Setting main cycle timer        */
     /* 20MHz* 1/8 * 250 * 100 =10ms    */
     prex = 250 - 1;                    /* Setting Prescaler X register */
     tx = 100 - 1;                      /* Setting timer X register */
     txs = 1;                           /* Timer X count start flag = start */
     /* Setting Buzzer output timer */
     /* the period of Timer Z primary = 20MHz* 1/8 * 25 * 50 = 0.5ms */
     /* the period of Timer Z secondary = 20MHz* 1/8 * 25 * 50 = 0.5ms */
     prez = 25 - 1;                     /* Setting Prescaler Z register */
     tzsc = 50 - 1;                     /* Setting timer Z secondary register */
     tzpr = 50 - 1;                     /* Setting timer Z Primary register */
     tzmr = 0x50;                       /* Timer Z mode = Programmable waveform generation mode */
     pum = 0;                           /* Timer Z waveform output control register */
     tzs = 0;                           /* Timer Z count start flag = stop */

     sfr_ref();                         /* Refreshment of SFR registers */
}
/************************************************************************************
Name:       sfr_ref
Parameters:  None
Returns:    None
Description: Refresh of SFR registers
************************************************************************************/
void sfr_ref(void){
     /* Setting port direction registers */
     pd1 = 0x1E;
     pd4 = 0x0;
}
```

```
/************************************************************************************
Name:       port_out
Parameters:  None
Returns:    None
Description: Port output
************************************************************************************/
void port_out(void){
     if (mode == 1){
          /* LED output processing */
          if (tm_ledon <= 0){
               if (cnt3min > 12000){         /* The remainder time is more than 2 minutes */
                    led_on = 1;
               }else{
                    if (cnt3min > 1000){     /* Change of LED control timer */
                         tm_ledon = 50;      /* Display during 500ms */
                    }else{
                         tm_ledon = 25;      /* Display during 250ms */
                    }
                    if (led_on == 0){        /* Change of display on-off */
                         led_on = 1;         /* Display start */
                    }else{
                         led_on = 0;         /* Display stoppage */
                    }
               }
          }
          if (led_on == 1){
               if (cnt3min > 6000){          /* The remainder time is more than 1 minutes */
                     p1 = 0x10;
               }else if (cnt3min > 0){       /* The remainder time is less than 1 minutes */
                    p1 = 0x12;
               }else{                        /* The remainder time is nothing */
                    p1 = 0x06;
               }
          }else{
               if (cnt3min == 0){            /* Counting end */
                    p1 = 0x06;
               }else{
                    p1 = 0x16;               /* LED turn-off */
               }
          }
          /* Buzzer output processing */
          if (cnt3min == 0 && mode == 1){
               if (tm_buzzer <= 0){
                    tm_buzzer = 20;          /* Buzzer control timer = 200ms */
                    switch (md_buzzer){
                    case 0:
                    case 2:
                         tzs = 1;            /* Counting start of timer Z */
                         md_buzzer = md_buzzer + 1;
                         break;
                    case 1:
                    case 3:
                    case 4:
                         tzs = 0;                 /* Counting stoppage of timer Z */
                         md_buzzer = md_buzzer + 1;
                         break;
                    default:
                         mode = 0;                /* Move to the standby mode */
                         md_buzzer = 0;
                    }
               }
          }else{
               tm_buzzer = 20;                    /* Buzzer output of 200ms */
          }
     }else{
          p1 = 0x06;
          tzs = 0;                                /* Timer Z stops Counting */
     }

}
```

```
/************************************************************************************
Name:       port_in
Parameters:  None
Returns:    None
Description: Port input
************************************************************************************/
void port_in(void){

 unsigned char i;

     /* Determination of input level SW4 */
     i = 1;
     if (SW4 == SW_ON)i = 0;                 /* Now determination SW4 */

     sw4_bit = sw4_bit << 1;                 /* Check pulses matching a trigger input level 3 times */
     sw4_bit = sw4_bit | i;
     sw4_bit = sw4_bit & 7;

     if (sw4_bit == 0 || sw4_bit == 7){      /* Determinate input SW4 */
          if (sw4_bit == 0){
               fixsw4 = 0;                   /* Input on */
               /* Counting start of the 3 minutes timer */
               if (mode == 0){
                    mode = 1;                /* Setting the 3 minute timer mode */
                    cnt3min = 18000;         /* Initialization of the 3 minutes timer */
                    led_on = 0;              /* Display start */
               }
          }else{
               fixsw4 = 1;                   /* Input off */
          }
     }

     /* Determination of input level SW1 */
     i = 1;
     if (SW1 == SW_ON)i = 0;                 /* Now determination SW1 */

     sw1_bit = sw1_bit << 1;                 /* Check pulses matching a trigger input level 3 times */
     sw1_bit = sw1_bit | i;
     sw1_bit = sw1_bit & 7;

     if (sw1_bit == 0 || sw1_bit == 7){      /* Determinate input SW1 */
          if (sw1_bit == 0){
               fixsw1 = 0;                   /* Input on */
               if (mode == 1){
                    mode = 0;                /* Setting the standby mode */
                    cnt3min = 0;
               }
          }else{
               fixsw1 = 1;                   /* Input off */
          }
     }
}
```

```
/************************************************************************************
Name:       stimer
Parameters:  None
Returns:     None
Description: Software timer
************************************************************************************/
void stimer(void){
     /* Countdown of 3 minutes timer */
     if (mode == 1){
          if (cnt3min != 0){
               cnt3min = cnt3min - 1;
          }
     }
     /* Countdown of LED control timer */
     if (tm_ledon != 0){
          tm_ledon = tm_ledon - 1;
     }
     /* Countdown of buzzer control timer */
     if (tm_buzzer != 0){
          tm_buzzer = tm_buzzer - 1;
     }
}
```

6. Reference Documents

Datasheet
R8C/Tiny Series Datasheet
(For the most current version, please visit Renesas Technology Home Page)

Hardware Manual
R8C/Tiny Series Hardware Manual
(For the most current version, please visit Renesas Technology Home Page)

7. Home Page and Support Information Window

Renesas Technology Home Page
http://www.renesas.com/

M16C Family MCU Technical Support Information Window
support_apl@renesas.com

REVISION HISTORY

Rev.	Date	Description	
		Page	Summary
1.00	Sep 09, 2003	-	First edition issued
1.10	Nov 10, 2003	ALL	For R8C/11 Group
2.00	Jul 1, 2004	ALL	For R8C/Tiny Series

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.