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.

# M16C/29 Group

## CAN Demo Set Program

## 1. Summary

This document describes the program for the CAN demonstration set for the M16C/29 group of Renesas microcomputers.
For CAN protocol, please refer to the CAN specification (Robert Bosch GmbH) and ISO11898.

## 2. Introduction

The application examples explained here apply for the use of the following microcomputer under the conditions described below.

- Microcomputer: M16C/29 group

Other M16C family microcomputers that contain the SFR (peripheral function control registers) similar to those of the M16C/29 group can use the program described here. However, since some functions may have been changed in functional additions, etc., be sure to consult the user's manual of your microcomputer for confirmation. Please carefully evaluate the suitability of your microcomputer before using this application note.

The sample program presented here operates under the conditions described below.

- Power supply voltage: 5 V
- Main clock input oscillation frequency: 20 MHz
- M16C/29 starter kit (M3A-0284) (However, the additional devices listed below are required.)
  Additional devices: CAN transceiver (HA13721)
  Capacitors (C12, C13, and C14 in the starter kit circuit diagram)
  Resistor (R8 or termination resistor in the starter kit circuit diagram)

## 3. Outline Specifications of the Sample Program

The sample program presented here is an example program for performing CAN communication.
It uses two M16C/29 starter kits (boards (1) and (2)) to perform CAN transmission/reception.

Board (1):

SW3 and SW4 may be used to change the transmit or receive IDs.
The variable resistor (VR) and SW2 are used to perform CAN transmission.
The LED display pattern is changed by the received data.

Board (2):

CAN transmission is performed by pressing SW2–4.
The first byte of received data is displayed on LEDs.

Figure 1 shows how the boards (1) and (2) should be connected when this program is used. Figure 2 shows a block diagram of the M16C/29.

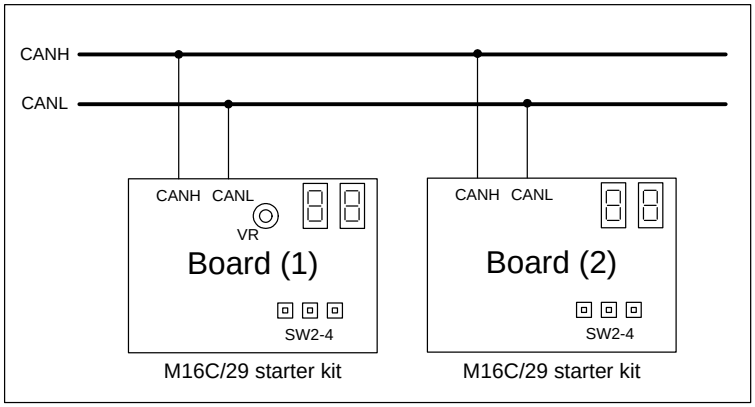


**Figure 1. Connection diagram**

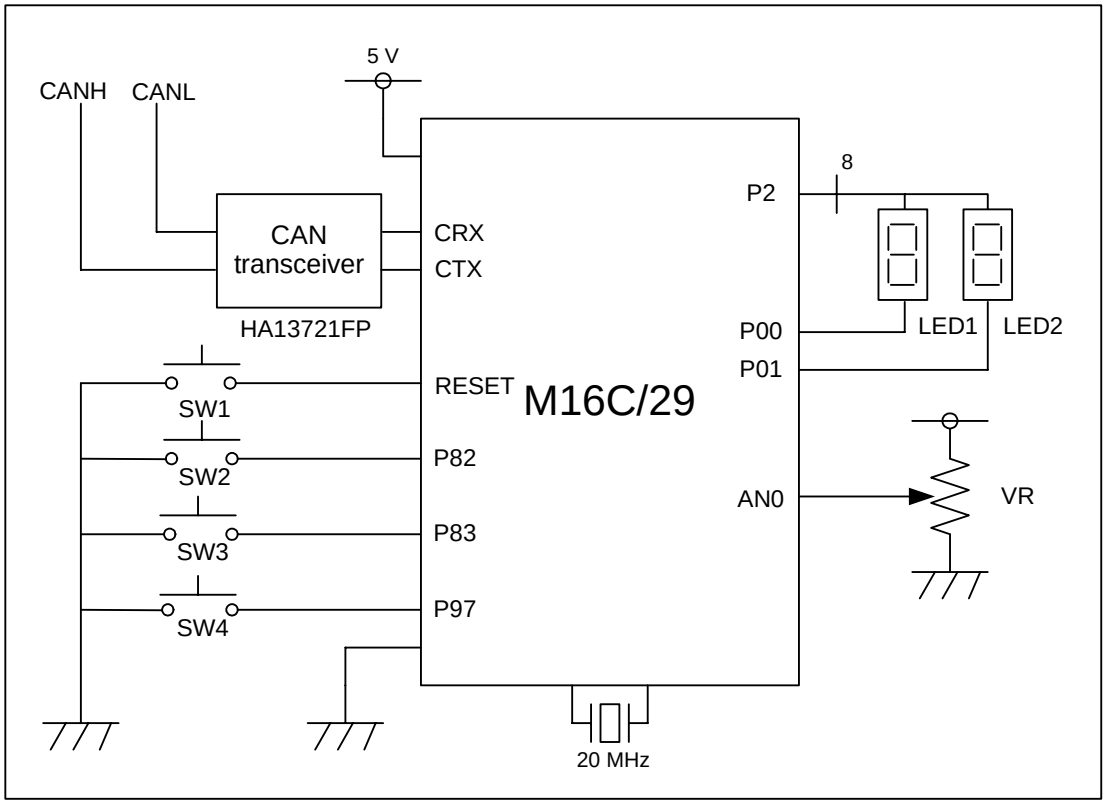


**Figure 2. Block diagram**

Table 1 lists the microcomputer pins used. Table 2 lists the function of each switch (SW).

**Table 1. Microcomputer Pins Used**

| Pin name | Input/output | Function |
|---|---|---|
| P00, P01 | Output | LED output |
| P20 - P27 | Output | LED output |
| P92 ( CRx ) | Input | CANout |
| P93 ( CTx ) | Output | CANin |
| P82 | Input | Switch 2 input |
| P83 | Input | Switch 3 input |
| P84 | Input | Switch 4 input |
| P100 | Input | Variable resistor<br>* Used on only board (1) |

**Table 2. Switch Functions**

| Switch name | SW1 | SW2 | SW3 | SW4 |
|---|---|---|---|---|
| Board (1) | Hardware reset | CAN transmit | Shows receive ID (Transmit ID + 1 while SW4 is held down) | Shows transmit ID (Receive ID + 1 while SW3 is held down) |
| Board (2) | Hardware reset | CAN transmit ( DATA1 ) | CAN transmit ( DATA2 ) | CAN transmit ( DATA3 ) |

# 4. Detailed Specifications of the Sample Program

This sample program permits you to perform CAN transmission by operating the switches or variable resistor on each board. Although CAN communication allows 0 to 8 bytes of data to be transmitted at a time, this program always transmits 1 byte of data. When either board receives data, it changes a 7-segment LED turn-on pattern according to the received data.

## 4.1 Switch Functions

### 4.1.1 Board (1)

SW1: Resets the board.

SW2: Performs CAN transmission.
Three types of data "$01_{16}$," "$02_{16}$," and "$03_{16}$" are used, which are transmitted sequentially each time you press SW2.

SW3: Shows the receive ID.
This switch shows the receive ID while you hold it down.
The receive ID is incremented by 1 when you press SW4 while it is displayed. Any receive ID in the range $00_{16}$ to $FF_{16}$ can be set.

SW4: Shows the transmit ID.
This switch shows the transmit ID while you hold it down.
The transmit ID is incremented by 1 when you press SW3 while it is displayed. Any transmit ID in the range $00_{16}$ to $FF_{16}$ can be set.

### 4.1.2 Board (2)

SW1: Resets the board.

SW2: Performs CAN transmission.
Data "$01_{16}$" is transmitted.

SW3: Performs CAN transmission.
Data "$02_{16}$" is transmitted.

SW4: Performs CAN transmission.
Data "$03_{16}$" is transmitted.

## 4.2 LED Display

### 4.2.1 Board (1)

The LED display pattern is changed by the first byte of received data. Furthermore, pressing SW3 on this board shows the received ID, and pressing SW4 on this board shows the transmit ID, both in hexadecimal. There are three LED display patterns that are changed by received data. These LED display patterns are shown in Table 3 and Figure 3.

**Table 3. LED Display Patterns**

| | Received data | LED1 | LED2 |
|---|---|---|---|
| Display pattern 1 | 01 | Counterclockwise | Clockwise |
| Display pattern 2 | 02 | Clockwise | Counterclockwise |
| Display pattern 3 | 03 | Clockwise using both LEDs | |

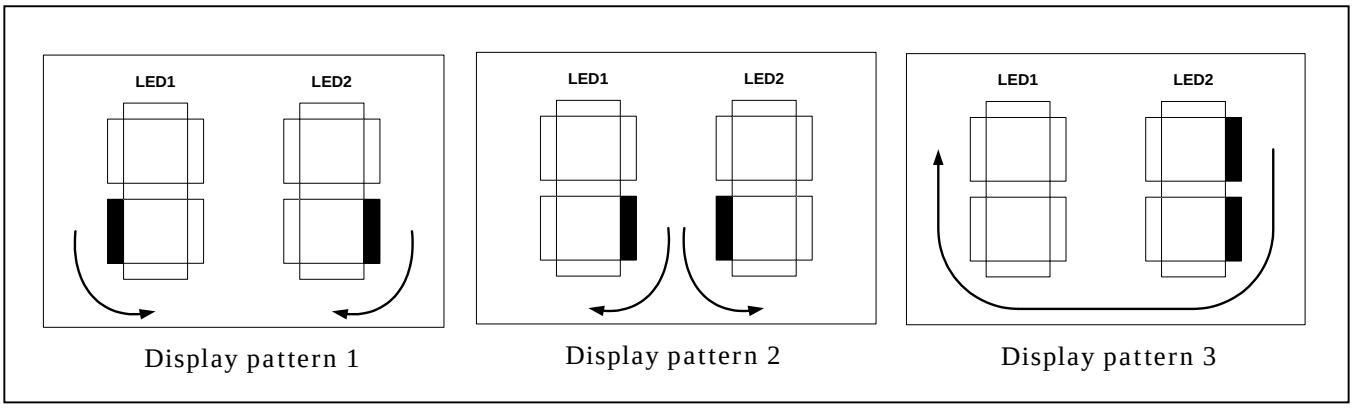


**Figure 3. Display pattern**

### 4.2.2 Board (2)

Shows the received data on LEDs (in hexadecimal).

## 4.3 CAN Settings

CAN-related settings on boards (1) and (2) are the same, except for the transmit ID and receive ID.

- Slot 0: Transmit slot
- Slot 14: Receive slot
- Slot 15: Receive slot
- BasicCAN mode
- CAN0 receive interrupt enable
- CAN0 transmit interrupt disable
- CAN0 error interrupt disable
- DLC (transmit data length): 1 (byte)
- Baud rate: 500 kbps
- Bit timing:
  PTS = 3 Tq's
  PBS1 = 3 Tq's
  PBS2 = 3 Tq's
  SJW = 1 Tq

Figure 4 shows a bit timing diagram for the case where a recessive-to-dominant transition occurs during the SS interval.

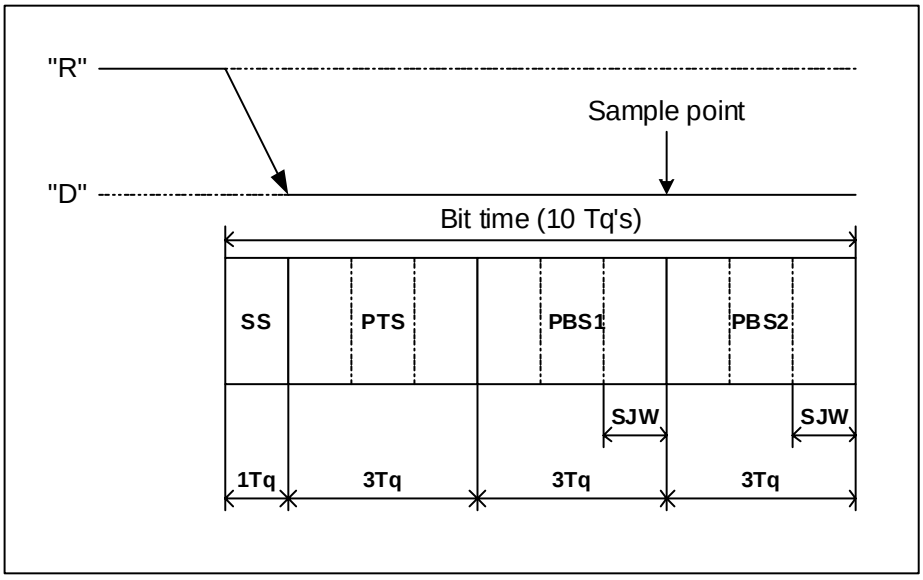


**Figure 4. Bit timing diagram**

Board (1) transmit ID: $00_{16}$ to $FF_{16}$ (initial value $02_{16}$)
Board (1) receive ID: $00_{16}$ to $FF_{16}$ (initial value $01_{16}$)
Board (2) transmit ID: $01_{16}$
Board (2) receive ID: $02_{16}$

## 4.4 A/D Conversion

### 4.4.1 Board (1)

Input from the variable resistor (VR) is A/D converted. If three converted values of input (latched every 2 ms) change by 5 or more on the average, the average value is transmitted by CAN. Although A/D conversion is performed in 10-bit mode, only the 8 high-order bits of converted data are transmitted.

Settings for A/D conversion

- Repeat mode
- 10-bit mode
- A/D conversion frequency: fAD divided by 4 (5 MHz)
- Pins used: P10_0

### 4.4.2 Board (2)

Board (2) does not use A/D conversion.

## 4.5 Error Handling

If an error occurs in either of boards (1) and (2), CAN communication and A/D conversion are stopped, with the error number indicated on LEDs.
LED1 shows the letter “E,” and LED2 shows an error number. Both LEDs flash.
Errors on board (1) and those on board (2) are listed in Tables 4 and 5, respectively.

**Table 4. Errors on Board (1)**

| Error number | Content of error |
|---|---|
| 1 | CAN receive buffer overrun error |
| 2 | Internal error 1 |
| 3 | Internal error 2 |

**Table 5. Errors on Board (2)**

| Error number | Content of error |
|---|---|
| 1 | CAN receive buffer overrun error |
| 2 | Internal error |

# 5. Flowchart

This section shows a flowchart of the sample program.
The functions in the shaded areas of the flowchart shown here are the functions described in the “CAN Application Note” for the M16C/6N, M16C/1N, and M16C/29 groups.

## 5.1 Flowchart of Program (1) (Board (1))

Figure 5 shows a flow of the main functions.

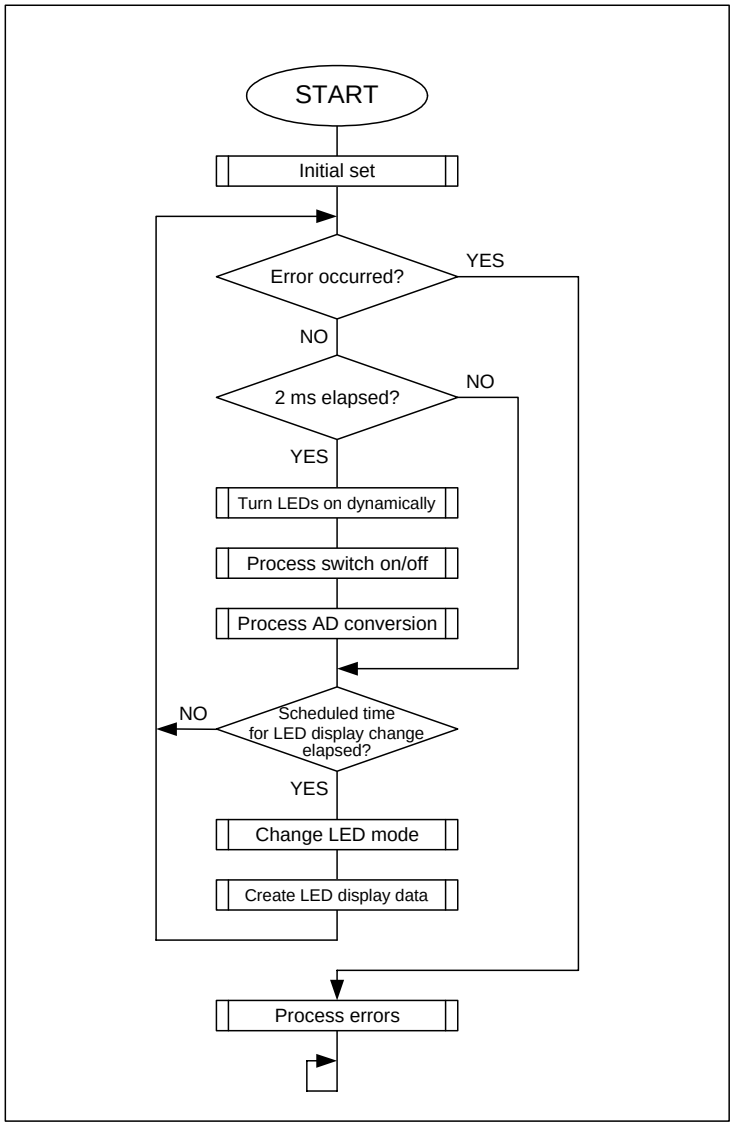


**Figure 5. Main function**

Figure 6 shows the content of initial setting.

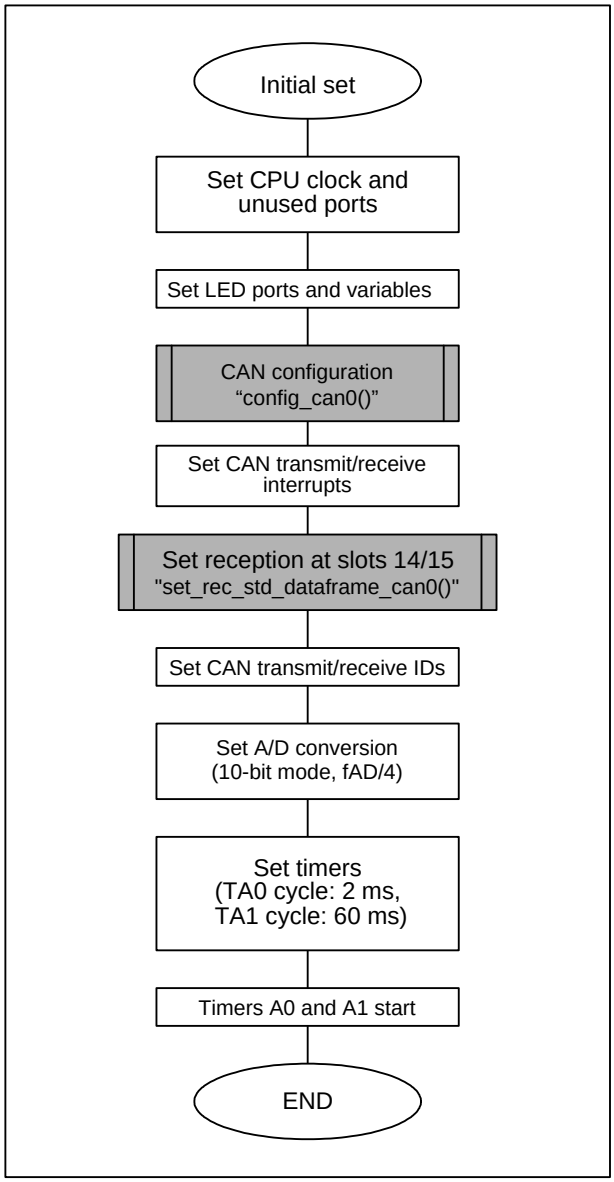


**Figure 6. Initial setting**

Figures 7, 8, and 9 show switch 2–4 depression processing (i.e., processing performed when a switch is depressed).
Furthermore, Figure 10 shows a process in which switch 4 is depressed while switch 3 is low. Figure 11 shows a process in which switch 3 is depressed while switch 4 is low. Figure 12 shows a process in which all switches are in the high state (i.e., no switches are depressed).

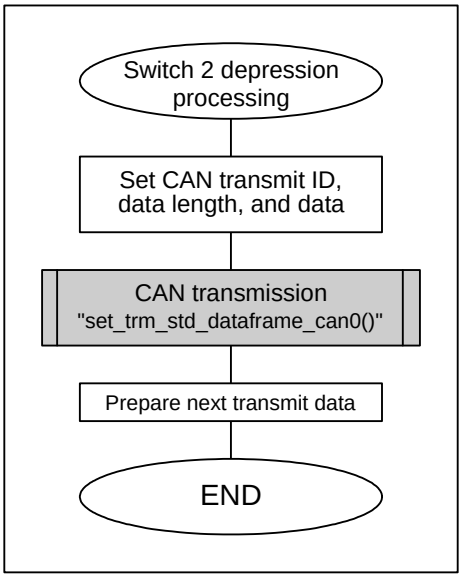


**Figure 7. Switch 2 depression processing**

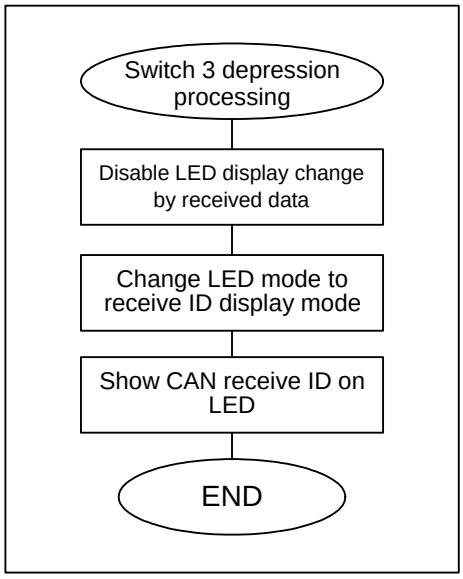


**Figure 8. Switch 3 depression processing**

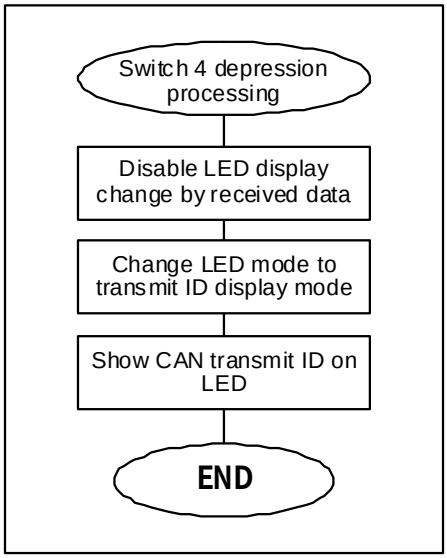


**Figure 9. Switch 4 depression processing**

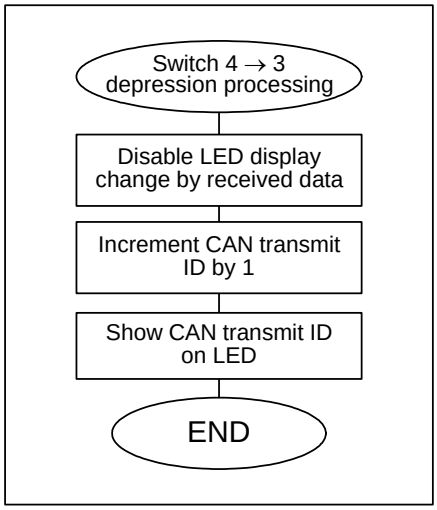


**Figure 10. Switch 4 → 3 depression processing**

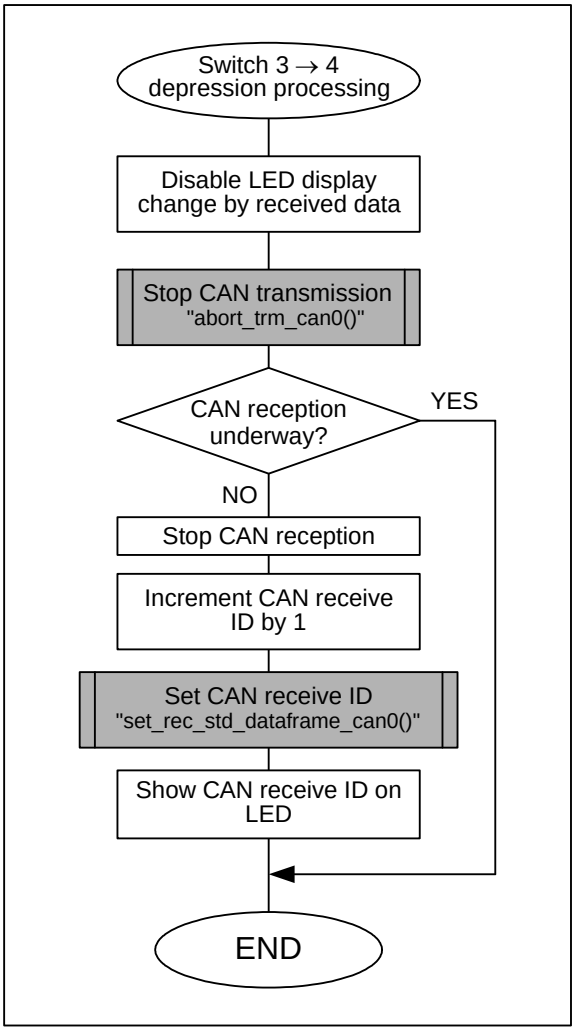


**Figure 11. Switch 3 → 4 depression processing**

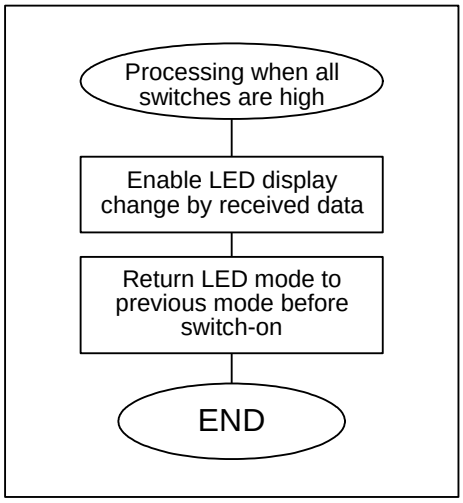


**Figure 12. Processing when all switches are high**

Figure 13 shows A/D conversion processing.

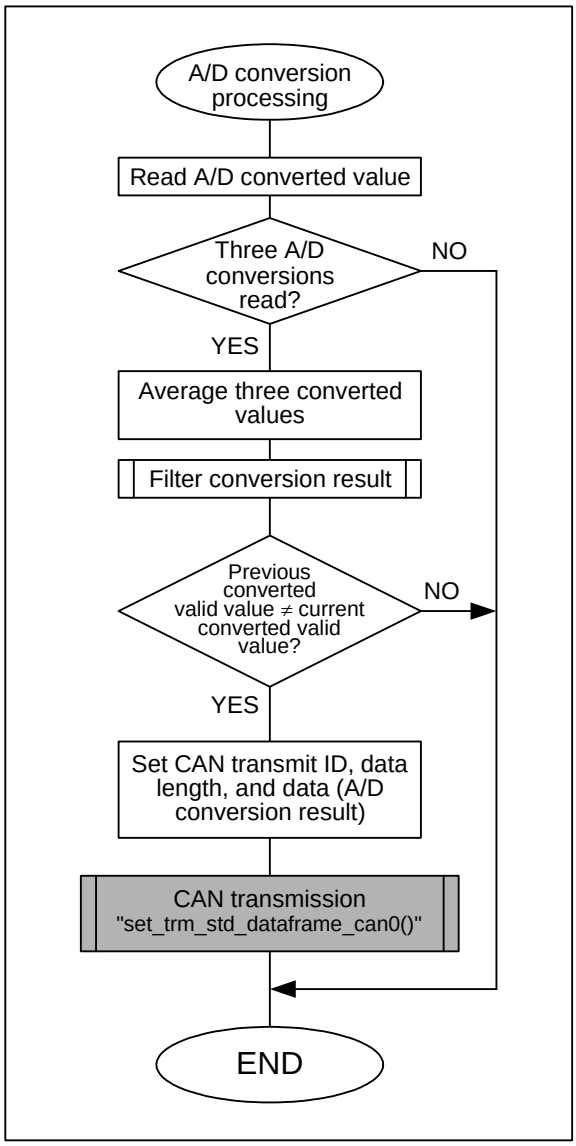


**Figure 13. A/D conversion processing**

Figure 14 shows CAN receive interrupt processing.

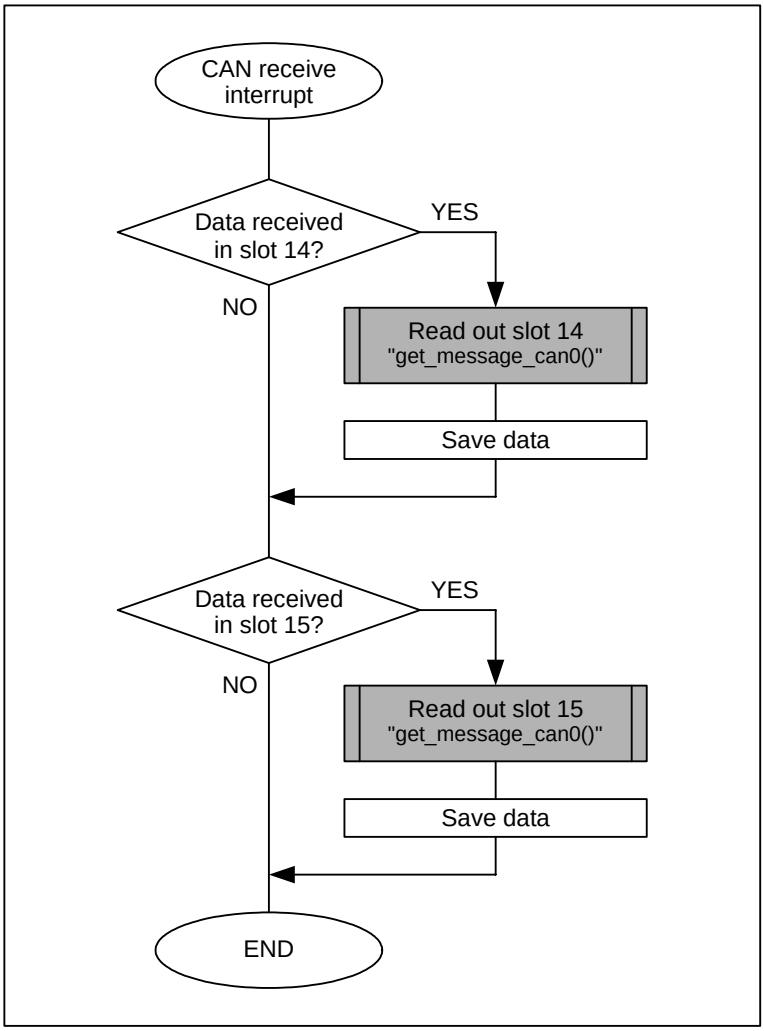


**Figure 14. CAN receive interrupt**

Figure 15 shows error processing.
In error processing, the program shows the letter “E” on LED1 and an error number on LED2. Furthermore, it causes both LEDs to flash every 0.5 seconds.

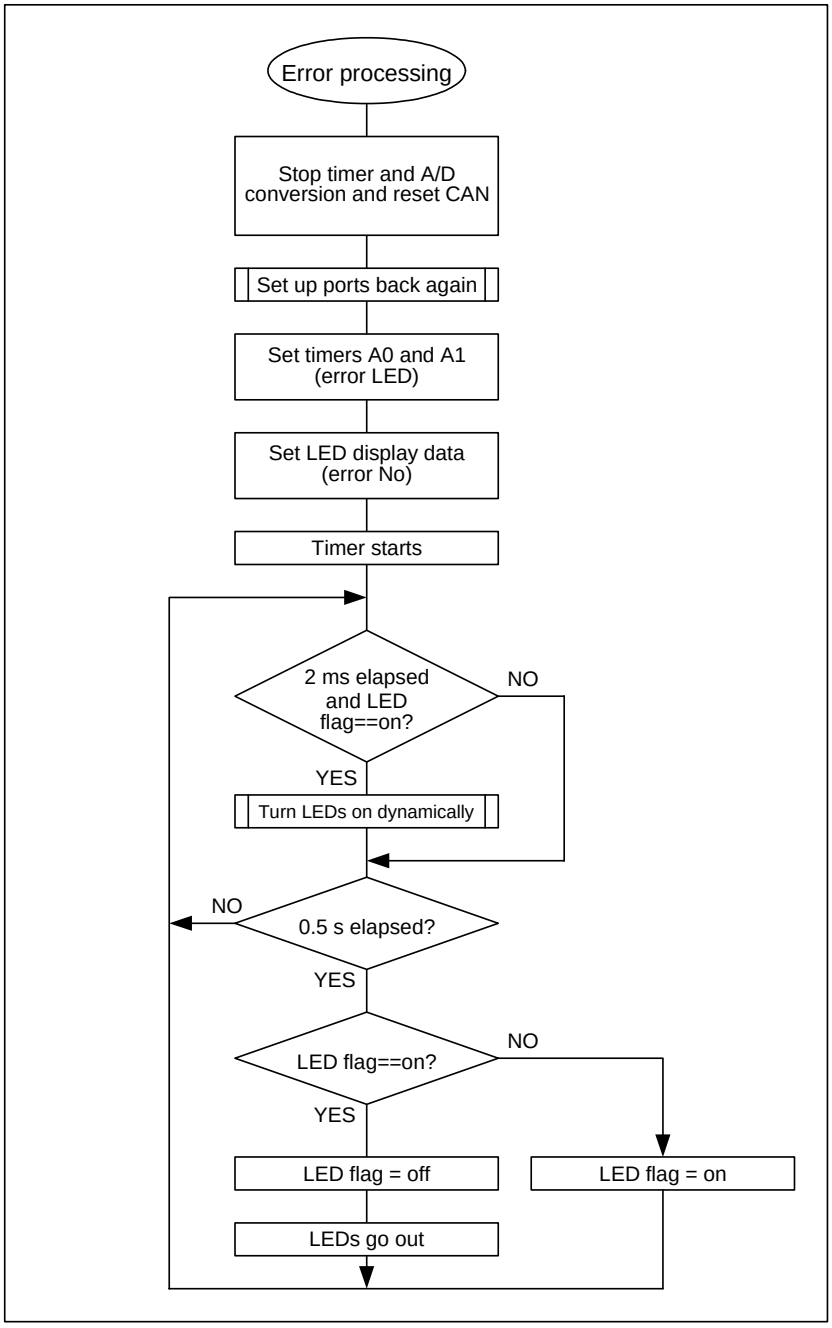


**Figure 15. Error processing**

## 5.2 Program (2) Flowchart (Board (2))

Figure 16 shows processing of the main function.
The “initial setting,” “LED dynamic turn-on,” “switch processing,” and “error processing” all are the same as in board (1) except the differences described below.

- Differences with the program in board (1)
  Initial setting: No A/D conversion related initial settings are involved. Nor is timer A0 used.
  Switch processing: Depression/release processing of each switch are different.
  Error processing: No A/D conversion related error processing is involved.

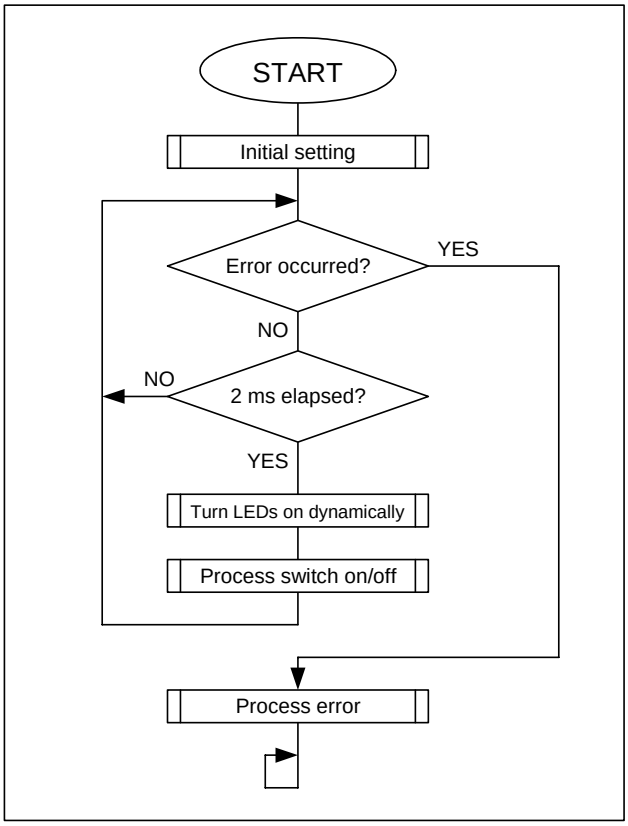


**Figure 16. Main function**

Figure 17 shows switches 2–4 depression processing.
Switches 2–4 on board (2) all perform CAN transmission. CAN transmit processing is the same with respect to the transmit ID and data length, with only data differing on each switch. No switch release processing is involved.

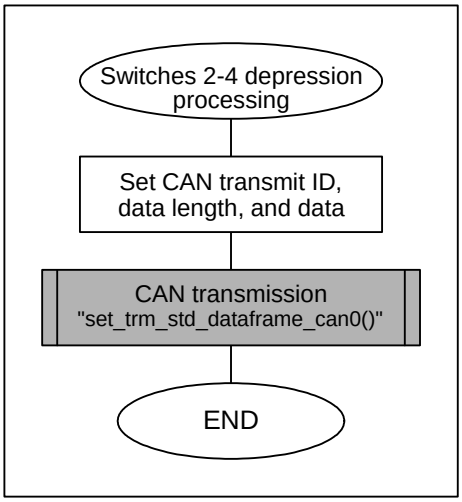


**Figure 17. Switch depression processing**

Figure 18 shows CAN receive interrupt processing.

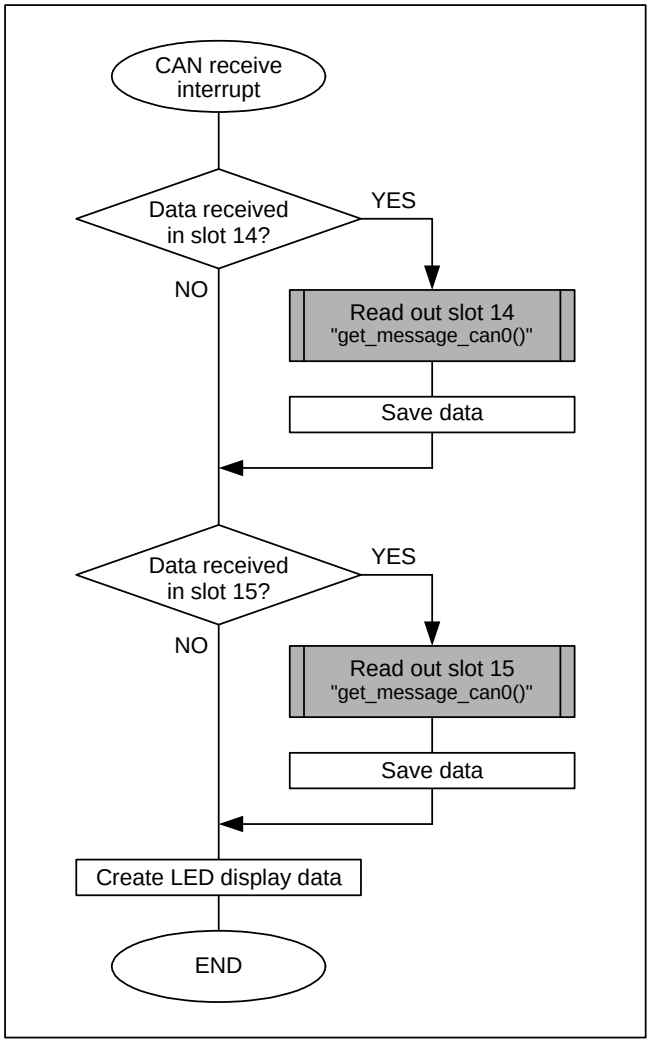


**Figure 18. CAN receive interrupt**

## 6. Sample Programs

Tables 6 and 7 show the file configuration of each sample program in boards (1) and (2), respectively.

**Table 6. File Configuration of the Sample Program in Board (1)**

| File name | Content |
|---|---|
| ncrt0.a30 | NC30 startup program |
| sect30.inc | Section definition |
| main1.c | Main program |
| can1.c | CAN related program |
| ad1.c | A/D conversion related program |
| sw_led1.c | Switch and LED related program |
| int1.c | Interrupt processing program |
| sfr29.h | SFR register header for the M16C/29 |
| define1.h | *define* definition header |
| declare1.h | Function and variable declaration header |

**Table 7. File Configuration of the Sample Program in Board (2)**

| File name | Content |
|---|---|
| ncrt0.a30 | NC30 startup program |
| sect30.inc | Section definition |
| main2.c | Main program |
| can2.c | CAN related program |
| sw_led2.c | Switch and LED related program |
| int2.c | Interrupt processing program |
| sfr29.h | SFR register header for the M16C/29 |
| define2.h | *define* definition header |
| declare2.h | Function and variable declaration header |

## 6.1 Program (1)

```
/*""FILE COMMENT""*****************************************************
*System Name : CAN DEMO PROGRAM1
*File Name    : define1.h
*Version      : 1.00
*Contens      :
*Customer     : RSO
*Model        :
*Order        :
*CPU          :
*Compiler     : NC30 Version 5.30 Release 02
*OS           :
*Programmer   :
*Note         :
************************************************************************
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
************************************************************************
*History          2005.07.1    Ver 1.00
*""FILE COMMENT END""*************************************************/

/* MODE */
#define      MODE_CLWS          1            //mode num
#define      MODE_ANTCLWS       2            //mode num
#define      MODE_HALF_CLWS     3            //mode num
#define      MODE_RCV_ID        4            //mode num
#define      MODE_TRM_ID        5            //mode num

#define      DATA_CLWS          1            //receive data pattern
#define      DATA_ANTCLWS       2            //receive data pattern
#define      DATA_HALF_CLWS     3            //receive data pattern


/* CAN */
#define      MASK_14       0xff            //No mask
#define      MASK_15       0xff            //

#define      TRM_SLOT      0               //CAN0 trm slot

#define      CAN_TRM_ID    1               //CAN0 trm initial ID
#define      CAN_RCV_ID    2               //CAN0 rcv initial ID


/* AD */
#define      AD_COUNT      3               //AD chatter


/* timer */
//TA0
```

```
#define     DATA_CLWS_TM          30 -1    //
#define     DATA_ANTCLWS_TM       30 -1    //
#define     DATA_HALF_CLWS_TM     20 -1    //

//TA1
#define     DATA_TA1      2000000/50 -1    //LED & SW (2ms)


/* SW & LED*/
#define     CHAT_COUNT            2        //
#define     SW_TBL_SIZE           7        //
#define     LED_PT12_TBL_SIZE     6        //
#define     LED_PT34_TBL_SIZE     6        //
#define     LED_PT56_TBL_SIZE     8        //

#define     PT_SW_MASK    0b00011100       //SW MASK

#define     PT_ALL        0b00000000       //SW all
#define     PT_SW34       0b00000100       //SW3 & SW4
#define     PT_SW24       0b00001000       //SW2 & SW4
#define     PT_SW4        0b00001100       //SW4
#define     PT_SW23       0b00010000       //SW2 & SW3
#define     PT_SW3        0b00010100       //SW3
#define     PT_SW2        0b00011000       //SW2
#define     PT_NO_SW      0b00011100       //no SW

/* error */
#define     NO_ERROR              0        //Error number
#define     CAN_RCV_BUFF_ERROR    1        //Error number
#define     TABLE_ERROR           2        //Error number
#define     LED_ERROR             3        //Error number


/* interrupt Lv */
#define TA0_LVL       0                    //
#define TA1_LVL       0                    //LED & SW
#define CAN0_LVL      5                    //CAN0
#define CAN0_ERROR_LVL    0                //CAN0 error


/************************** LED DEFINE *******************************/
#define SEG_A    0x01
#define SEG_B    0x02
#define SEG_C    0x04
#define SEG_D    0x08
#define SEG_E    0x10
#define SEG_F    0x20
#define SEG_G    0x40
#define SEG_H    0x80
#define DISP_0   (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F          ))
#define DISP_1   (~(      SEG_B|SEG_C                            ))
#define DISP_2   (~(SEG_A|SEG_B|      SEG_D|SEG_E|      SEG_G    ))
#define DISP_3   (~(SEG_A|SEG_B|SEG_C|SEG_D|          SEG_G    ))
#define DISP_4   (~(      SEG_B|SEG_C|            SEG_F|SEG_G    ))
```

```
#define DISP_5   (~(SEG_A|       SEG_C|SEG_D|       SEG_F|SEG_G       ))
#define DISP_6   (~(SEG_A|       SEG_C|SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_7   (~(SEG_A|SEG_B|SEG_C|             SEG_F            ))
#define DISP_8   (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_9   (~(SEG_A|SEG_B|SEG_C|SEG_D       |SEG_F|SEG_G       ))
#define DISP_A   (~(SEG_A|SEG_B|SEG_C|       SEG_E|SEG_F|SEG_G       ))
#define DISP_B   (~(            SEG_C|SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_C   (~(SEG_A|             SEG_D|SEG_E|SEG_F            ))
#define DISP_D   (~(      SEG_B|SEG_C|SEG_D|SEG_E|       SEG_G       ))
#define DISP_E   (~(SEG_A|             SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_F   (~(SEG_A|                   SEG_E|SEG_F|SEG_G       ))

#define DISP_CL0     (~(SEG_A                                        ))
#define DISP_CL1     (~(      SEG_B                                  ))
#define DISP_CL2     (~(            SEG_C                            ))
#define DISP_CL3     (~(                  SEG_D                      ))
#define DISP_CL4     (~(                        SEG_E                ))
#define DISP_CL5     (~(                              SEG_F          ))

#define DISP_CL1_0   (~(SEG_A                                        ))
#define DISP_CL1_1   (~(SEG_A|SEG_B                                  ))
#define DISP_CL1_2   (~(      SEG_B|SEG_C                            ))
#define DISP_CL1_3   (~(            SEG_C|SEG_D                      ))
#define DISP_CL1_4   (~(                  SEG_D                      ))
#define DISP_CL1_5   ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL1_6   ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL1_7   ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_0   (~(SEG_A                                        ))
#define DISP_CL2_1   ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_2   ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_3   ((SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G|SEG_H))
#define DISP_CL2_4   (~(                  SEG_D                      ))
#define DISP_CL2_5   (~(                  SEG_D|SEG_E                ))
#define DISP_CL2_6   (~(                        SEG_E|SEG_F          ))
#define DISP_CL2_7   (~(SEG_A                         |SEG_F          ))
/*********************************************************************/

typedef struct{
    unsigned short id;
    unsigned char dlc;
    union {
        unsigned char data[8];
        unsigned long long data_8;
    }data;
}can_std_data_def;
```

```
/*""FILE COMMENT""*****************************************************
*System Name : CAN DEMO PROGRAM1
*File Name    : declare1.h
*Version      : 1.00
*Contens      :
*Customer     : RSO
*Model        :
*Order        :
*CPU          :
*Compiler     : NC30 Version 5.30 Release 02
*OS           :
*Programmer   :
*Note         :
*********************************************************************
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*********************************************************************
*History          2005.07.1    Ver 1.00
*""FILE COMMENT END""*************************************************/

#ifdef DECLARE_MAIN
    #define      READTASK_EXTERN
#else
    #define      READTASK_EXTERN extern
#endif

/*********************************************************************
*                fnc
*********************************************************************/
/* MAIN */
READTASK_EXTERN void main(void);
READTASK_EXTERN void cpu_initial(void);
READTASK_EXTERN void timer_initial(void);
READTASK_EXTERN void sw_led_initial(void);

/* CAN */
READTASK_EXTERN void can_initial(void);
READTASK_EXTERN void set_bitrate_can0(void);
READTASK_EXTERN void config_can0(void);
READTASK_EXTERN void set_rec_std_dataframe_can0(unsigned short, unsigned short);
READTASK_EXTERN void set_mask_can0(void);
READTASK_EXTERN void set_trm_std_dataframe_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void get_message_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void trm_can_ad(void);
READTASK_EXTERN void abort_trm_can0(unsigned short in_slot);

/* SW & LED */
READTASK_EXTERN void led_fnc(void);
READTASK_EXTERN void led_pt_fnc(void);
READTASK_EXTERN void led_mode_change(void);
READTASK_EXTERN void sw_fnc(void);
```

```
READTASK_EXTERN void sw_2(void);
READTASK_EXTERN void sw_3(void);
READTASK_EXTERN void sw_4(void);
READTASK_EXTERN void sw_3_4(void);
READTASK_EXTERN void sw_4_3(void);
READTASK_EXTERN void sw_other(void);
READTASK_EXTERN unsigned char fix_sw_fnc(unsigned char);
READTASK_EXTERN void dsp_return(void);
READTASK_EXTERN unsigned char sw_decode(void);

/* AD */
READTASK_EXTERN void ad_initial(void);
READTASK_EXTERN void ad_fnc(void);
READTASK_EXTERN unsigned char ad_read(void);
READTASK_EXTERN unsigned char ad_check(void);

/* ERROR */
READTASK_EXTERN void error_fnc(unsigned char);
READTASK_EXTERN void error_main_fnc(void);

/************************************************************************
*                  interrupt fnc
************************************************************************/
#pragma INTERRUPT /B CAN0_REC()
READTASK_EXTERN void CAN0_REC(void);

/************************************************************************
*                  variable
************************************************************************/
/* CAN */
READTASK_EXTERN can_std_data_def rec_data[2];       //CAN receive data
READTASK_EXTERN unsigned char F_RCV_DATA;           //Receive data display enable flag
READTASK_EXTERN unsigned char trm_id;               //Transmit ID
READTASK_EXTERN unsigned char rcv_id;               //Receive ID
READTASK_EXTERN unsigned char last_rcv_data;        //Last CAN receive data

/* SW */
READTASK_EXTERN unsigned char trm_count_sw;         //Transmit counter
READTASK_EXTERN unsigned char last_sw_data;         //Last switch data
READTASK_EXTERN unsigned char fix_sw_data;          //Fixed switch data
READTASK_EXTERN unsigned char last_fix_sw_data;     //Last fixed switch data
READTASK_EXTERN unsigned char sw_count;             //Switch counter
READTASK_EXTERN unsigned char F_SW;                 //Switch enable flag

/* LED */
READTASK_EXTERN unsigned char led_count;            //LED counter
READTASK_EXTERN unsigned char led_data[2];          //LED display data
READTASK_EXTERN unsigned char led_mode;             //LED mode
READTASK_EXTERN unsigned char last_led_mode;        //Last LED mode
READTASK_EXTERN unsigned char led_pt_count;         //LED display pattern counter

/* AD */
READTASK_EXTERN unsigned char last_ad_data;         //Last A/D data
READTASK_EXTERN unsigned short ave_ad_data;         //Average of A/D conversion data
```

```
READTASK_EXTERN unsigned char count_ad;                 //A/D counter
READTASK_EXTERN unsigned short ad_data[3];              //A/D data

/* ERROR */
READTASK_EXTERN unsigned char error_num;

/************************************************************************
*                    table
************************************************************************/
/* LED display table */
READTASK_EXTERN const unsigned char LED_TBL[16]
#ifdef DECLARE_MAIN
      = {DISP_0,DISP_1,DISP_2,DISP_3,
         DISP_4,DISP_5,DISP_6,DISP_7,
         DISP_8,DISP_9,DISP_A,DISP_B,
         DISP_C,DISP_D,DISP_E,DISP_F,}
#endif
     ;

/* LED display pattern table (mode clockwise) */
READTASK_EXTERN const unsigned char LED_PT1_TBL[LED_PT12_TBL_SIZE]
#ifdef DECLARE_MAIN
      = {DISP_CL0,DISP_CL1,DISP_CL2,
         DISP_CL3,DISP_CL4,DISP_CL5}
#endif
     ;

/* LED display pattern table (mode clockwise) */
READTASK_EXTERN const unsigned char LED_PT2_TBL[LED_PT12_TBL_SIZE]
#ifdef DECLARE_MAIN
      = {DISP_CL0,DISP_CL5,DISP_CL4,
         DISP_CL3,DISP_CL2,DISP_CL1}
#endif
     ;

/* LED display pattern table (mode anticlockwise) */
READTASK_EXTERN const unsigned char LED_PT3_TBL[LED_PT34_TBL_SIZE]
#ifdef DECLARE_MAIN
      = {DISP_CL0,DISP_CL5,DISP_CL4,
         DISP_CL3,DISP_CL2,DISP_CL1}
#endif
     ;

/* LED display pattern table (mode anticlockwise) */
READTASK_EXTERN const unsigned char LED_PT4_TBL[LED_PT34_TBL_SIZE]
#ifdef DECLARE_MAIN
      = {DISP_CL0,DISP_CL1,DISP_CL2,
         DISP_CL3,DISP_CL4,DISP_CL5}
#endif
     ;
```

```
/* LED display pattern table (mode half clockwise) */
READTASK_EXTERN const unsigned char LED_PT5_TBL[LED_PT56_TBL_SIZE]
#ifdef DECLARE_MAIN
      = {DISP_CL1_0,DISP_CL1_1,DISP_CL1_2,DISP_CL1_3,
         DISP_CL1_4,DISP_CL1_5,DISP_CL1_6,DISP_CL1_7}
#endif
    ;

/* LED display pattern table (mode half clockwise) */
READTASK_EXTERN const unsigned char LED_PT6_TBL[LED_PT56_TBL_SIZE]
#ifdef DECLARE_MAIN
      = {DISP_CL2_0,DISP_CL2_1,DISP_CL2_2,DISP_CL2_3,
         DISP_CL2_4,DISP_CL2_5,DISP_CL2_6,DISP_CL2_7}
#endif
    ;

READTASK_EXTERN const unsigned char trm_data_sw[3]
#ifdef DECLARE_MAIN
      = {DATA_CLWS,DATA_ANTCLWS,DATA_HALF_CLWS}
#endif
    ;
/* Jump table */
READTASK_EXTERN void (* const SW_TBL[SW_TBL_SIZE])(void)
#ifdef DECLARE_MAIN
      = {dsp_return,sw_2,sw_3,sw_4,sw_3_4,sw_4_3,sw_other}
#endif
    ;
```

```
/*""FILE COMMENT""*****************************************************
*System Name : CAN DEMO PROGRAM1
*File Name    : main1.c
*Version      : 1.00
*Contens      :
*Customer     : RSO
*Model        :
*Order        :
*CPU          :
*Compiler     : NC30 Version 5.30 Release 02
*OS           :
*Programmer   :
*Note         :
*             :
************************************************************************
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
************************************************************************
*History          2005.07.1    Ver 1.00
*""FILE COMMENT END""*************************************************/

#define DECLARE_MAIN

#include "sfr29.h"
#include "define1.h"
#include "declare1.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :1.0
  *Description         :
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *--------------------------------------------------------------------
  *Declaration         :void main(void)
  *--------------------------------------------------------------------
  *Function            :main
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :unsigned char error_num          ;error number
  *Ountput             :unsigned char error_num          ;error number
  *--------------------------------------------------------------------
  *Call functions     :cpu_initial()                ;Initialize CPU
  *                    :sw_led_initial()            ;Initialize switch and LED
  *                    :can_initial()               ;Initialize CAN
  *                    :ad_initial()                ;Initialize A/D conberter
  *                    :timer_initial()             ;Initialize timer
```

```
   *                    :led_mode_change()          ;Change LED mode
   *                    :led_pt_fnc()               ;Change LED lightning pattern
   *                    :led_fnc()                  ;LED dynamic lightning
   *                    :sw_fnc()                   ;Read switch port
   *                    :ad_fnc()                   ;A/D convert
   *                    :error_main_fnc()           ;Error function
   *-------------------------------------------------------------------
   *Note                :
   *-------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""***************************************************/
void main(void)
{
    asm("fclr I");

    cpu_initial();
    sw_led_initial();
    can_initial();
    ad_initial();
    timer_initial();

    /* variable initial */
    error_num = NO_ERROR;           //set error flag "No error"

    tabsr = 0x03;                   //LED & SW timer
    asm("fset I");


    while(error_num == NO_ERROR){   //Error check
        if(ir_ta1ic == 1){          //TA1
            ta1ic = TA1_LVL;        //IR clear

            led_fnc();              //LED dynamic lightning
            sw_fnc();               //Switch function
            ad_fnc();               //A/D convert
        }

        if(ir_ta0ic == 1){          //TA0
            ta0ic = TA0_LVL;        //IR clear

            led_mode_change();      //Change LED mode
            led_pt_fnc();           //Change LED lightning pattern
        }
    }



/* ERROR */
    error_main_fnc();               //Error function

    while(1);
}
```

```
 /*""FUNC COMMENT""*****************************************************
  *ID                  :1.1
  *Description         :Initialize CPU
  *---------------------------------------------------------------------
  *Include             :"sfr29.h"
  *---------------------------------------------------------------------
  *Declaration         :void cpu_initial(void)
  *---------------------------------------------------------------------
  *Function            :Initialize main clock,processer mode and ports
  *---------------------------------------------------------------------
  *Arguments           :Nothing
  *---------------------------------------------------------------------
  *Returns             :Nothing
  *---------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :Nothing
  *---------------------------------------------------------------------
  *Call functions      :Nothing
  *---------------------------------------------------------------------
  *Note                :
  *---------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
void cpu_initial(void)
{
    while(pacr != 0x03){
        prcr = 0x04;
        pacr = 0x03;                    //80pin
    }

    prcr = 0x03;

    /* clock,processer mode */
    cm1 = 0x20;
    cm2 = 0;
    pm0 = 0;
    pm1 = 0x08;
    cm0 = 0x08;
    pm2 = 0;

    /* port */
    p0   = 0x00;
    p1   = 0x00;
    p2   = 0x00;
    p3   = 0x00;
#if !_KD_
    p6   = 0x00;
#endif
    p7   = 0x00;
    p8   = 0x00;
    p9   = 0x00;
    p10   = 0x00;
```

```
    pd0 = 0x00;
    pd1 = 0x00;
    pd2 = 0x00;
    pd3 = 0x00;
#if !_KD_
    pd6 = 0x00;
#endif
    pd7 = 0x00;
    pd8 = 0x00;
    prcr = 0x07;
    pd9 = 0x00;
    pd10 = 0x00;

    prcr = 0;

    /* pull up control */
    pur0 = 0x00;
    pur1 = 0x00;
    pur2 = 0x00;

    /* port control */
    pcr = 0x00;
}

 /*""FUNC COMMENT""*****************************************************
  *ID                 :1.2
  *Description        :Initialize timer
  *-------------------------------------------------------------------
  *Include            :"sfr29.h"
  *                   :"define1.h"
  *-------------------------------------------------------------------
  *Declaration        :void timer_initial(void)
  *-------------------------------------------------------------------
  *Function           :Initialize TA0 and TA1
  *-------------------------------------------------------------------
  *Arguments          :Nothing
  *-------------------------------------------------------------------
  *Returns            :Nothing
  *-------------------------------------------------------------------
  *Input              :Nothing
  *Ountput            :Nothing
  *-------------------------------------------------------------------
  *Call functions     :Nothing
  *-------------------------------------------------------------------
  *Note               :
  *-------------------------------------------------------------------
  *History            :
  *                   :
  *""FUNC COMMENT END""************************************************/
void timer_initial(void)
{
    tabsr = 0;                  //all timer stop
    onsf = 0;
    trgsr = 0;
```

```
    ta0mr = 0x01;                   //ivent
    ta0tgl = 1;                     //count sorce:TA1
    ta0tgh = 1;                     //
    ta0 = DATA_CLWS_TM;             //
    ta0ic = TA0_LVL;

    ta1mr = 0x00;                   //timer mode
    ta1 = DATA_TA1;                 //
    ta1ic = TA1_LVL;
}


 /*""FUNC COMMENT""*****************************************************
  *ID                :1.3
  *Description       :Error process
  *-------------------------------------------------------------------
  *Include           :"sfr29.h"
  *-------------------------------------------------------------------
  *Declaration       :void error_fnc(void)
  *-------------------------------------------------------------------
  *Function          :Timer,A/D converter and CAN module are stop.
  *                  :LED off
  *-------------------------------------------------------------------
  *Arguments          :unsigned char in_num              ;Error number
  *-------------------------------------------------------------------
  *Returns           :Nothing
  *-------------------------------------------------------------------
  *Input             :Nothing
  *Ountput            :unsigned char error_num          ;Error number
  *-------------------------------------------------------------------
  *Call functions    :Nothing
  *-------------------------------------------------------------------
  *Note              :
  *-------------------------------------------------------------------
  *History           :
  *                  :
  *""FUNC COMMENT END""************************************************/
void error_fnc(unsigned char in_num)
{
    asm("fclr I");

    tabsr = 0;                      //all timer stop
    adcon0 = 0;                     //AD stop
    reset_c0ctlr = 1;               //CAN reset

    p0 = 0x03;                      //LED OFF

    error_num = in_num;             //Set error number
}

 /*""FUNC COMMENT""*****************************************************
  *ID                :1.4
  *Description       :Error process
```

```
    *-----------------------------------------------------------------
    *Include              :"sfr29.h"
    *-----------------------------------------------------------------
    *Declaration          :void error_main_fnc(void)
    *-----------------------------------------------------------------
    *Function             :Display error number
    *-----------------------------------------------------------------
    *Arguments            :Nothing
    *-----------------------------------------------------------------
    *Returns              :Nothing
    *-----------------------------------------------------------------
    *Input                :unsigned char error_num          ;Error number
    *                     :unsigned char led_count          ;LED counter
    *                     :unsigned char led_data[]         ;LED display data
    *Ountput              :unsigned char led_count          ;LED counter
    *                     :unsigned char led_data[]         ;LED display data
    *-----------------------------------------------------------------
    *Call functions       :sw_led_initial()                 ;Initialize switch and LED
    *                     :led_fnc()                        ;LED dynamic lightning
    *-----------------------------------------------------------------
    *Note                 :
    *-----------------------------------------------------------------
    *History              :
    *                     :
    *""FUNC COMMENT END""***********************************************/
void error_main_fnc(void)
{
    unsigned char in_led_on_off = 0;
    unsigned char i;

    asm("fclr I");

    tabsr = 0;                  //all timer stop
    adcon0 = 0;                 //AD stop
    reset_c0ctlr = 1;           //CAN reset

    sw_led_initial();

    ta0mr = 0x01;               //ivent
    ta0tgl = 1;                 //count sorce:TA1
    ta0tgh = 1;                 //
    ta0 = 250 - 1;              //
    ta0ic = 0x00;

    ta1mr = 0x00;               //timer mode
    ta1 = 2000000/50 -1;        //2ms
    ta1ic = 0x00;

    led_count = 0;
    led_data[1] = LED_TBL[0x0e];        //Set error number
    led_data[0] = LED_TBL[error_num];   //

    tabsr = 0x03;               //TA0,TA1 start
```

```
        while(1){
            if(in_led_on_off == 1){
                if(ir_ta1ic == 1){
                    ta1ic = 0x00;

                    led_fnc();
                }
            }

            if(ir_ta0ic == 1){          //LED flashing
                ta0ic = 0;

                if(in_led_on_off == 0){
                    in_led_on_off = 1;   //LED ON
                }
                else{
                    in_led_on_off = 0;   //LED OFF
                    p0 = 0x03;
                }
            }
        }
    }
```

```
/*""FILE COMMENT""*****************************************************
 *System Name : CAN DEMO PROGRAM1
 *File Name    : can1.c
 *Version      : 1.00
 *Contens      :
 *Customer     : RSO
 *Model        :
 *Order        :
 *CPU          :
 *Compiler     : NC30 Version 5.30 Release 02
 *OS           :
 *Programmer   :
 *Note         :
 *             :
 ************************************************************************
 * Copyright(C)2005, Renesas Technology Corp.
 * Copyright(C)2005, Renesas Solutions Corp.
 * All rights reserved.
 ************************************************************************
 *History          2005.07.1   Ver 1.00
 *""FILE COMMENT END""*************************************************/

#include "sfr29.h"
#include "define1.h"
#include "declare1.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :2.0
  *Description         :Initialize CAN
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *--------------------------------------------------------------------
  *Declaration         :void can_initial(void)
  *--------------------------------------------------------------------
  *Function            :Initialize CAN
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :unsigned char trm_id          ;transmit ID
  *                    :unsigned char rcv_id          ;receive ID
  *--------------------------------------------------------------------
  *Call functions      :config_can0()
  *                    :set_rec_std_dataframe_can0
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
```

```
   *""FUNC COMMENT END""************************************************/
  void can_initial(void)
  {
      config_can0();                    //Configurate can
      c0recic = CAN0_LVL;               //Interrupt lvl
      c01erric = CAN0_ERROR_LVL;        //Interrupt lvl

      set_rec_std_dataframe_can0(14,CAN_RCV_ID);   //Set can slot14
      set_rec_std_dataframe_can0(15,CAN_RCV_ID);   //Set can slot15

      trm_id = CAN_TRM_ID;              //Set transmit ID
      rcv_id = CAN_RCV_ID;              //Set receive ID
  }

   /*""FUNC COMMENT""*****************************************************
    *ID                 :2.1
    *Description        :Set can slot receive
    *--------------------------------------------------------------------
    *Include            :"sfr29.h"
    *                   :"declare1.h"
    *--------------------------------------------------------------------
    *Declaration        :void set_rec_std_dataframe_can0
    *                        (unsigned short in_slot,unsigned short in_sid)
    *--------------------------------------------------------------------
    *Function           :Set can slot receive
    *--------------------------------------------------------------------
    *Arguments          :unsigned short in_slot
    *                   :unsigned short in_sid
    *--------------------------------------------------------------------
    *Returns            :Nothing
    *--------------------------------------------------------------------
    *Input              :Nothing
    *Ountput            :Nothing
    *--------------------------------------------------------------------
    *Call functions     :Nothing
    *--------------------------------------------------------------------
    *Note               :
    *--------------------------------------------------------------------
    *History            :
    *                   :
    *""FUNC COMMENT END""************************************************/
  void set_rec_std_dataframe_can0(
      unsigned short in_slot,
      unsigned short in_sid)
  {
      c0icr |= (0x0001<<in_slot);
      c0idr &= ~(0x0001<<in_slot);

      c0slot[in_slot].ba.sidh = (in_sid>>6) & 0x1f;     //SID10-6
      c0slot[in_slot].ba.sidl = in_sid & 0x3f;          //SID5-0

      while(c0mctl[in_slot].byte != 0x00){
          c0mctl[in_slot].byte = 0x00;
      }
```

```
    c0mctl[in_slot].byte = 0x40;
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :2.0.2
  *Description         :Set can bitrate
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"declare1.h"
  *--------------------------------------------------------------------
  *Declaration         :void set_bitrate_can0(void)
  *--------------------------------------------------------------------
  *Function            :Set can bitrate
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :Nothing
  *--------------------------------------------------------------------
  *Call functions      :Nothing
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
void set_bitrate_can0(void)
{
    prc0 = 1;
    cclkr = 0x01;          // CAN0 clock = f1/2
    prc0 = 0;

    /* 500kbps 10Tq */
    brp_c0conr = 0;        // fcan(10MHz)
    sam_c0conr = 0;        //
    pts_c0conr = 2;        // PTS = 3Tq
    pbs1_c0conr = 2;       // PBS1 = 3 Tq
    pbs2_c0conr = 2;       // PBS2 = 3 Tq
    sjw_c0conr = 0;        // SJW = 1 Tq
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :2.0.1
  *Description         :CAN configuration
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *--------------------------------------------------------------------
  *Declaration         :void config_can0()            ;
```

```
    *-------------------------------------------------------------------
    *Function            :Set can slot receive
    *-------------------------------------------------------------------
    *Arguments           :Nothing
    *-------------------------------------------------------------------
    *Returns             :Nothing
    *-------------------------------------------------------------------
    *Input               :Nothing
    *Ountput             :Nothing
    *-------------------------------------------------------------------
    *Call functions      :set_bitrate_can0()
    *                    :set_mask_can0()
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""***************************************************/
void config_can0(void)
{
    reset_c0ctlr = 1;                   /* configration */
    sleep_c0ctlr = 0;
    while(! state_reset_c0str){}

    porten_c0ctlr = 1;                  /* CAN port enable */
    loopback_c0ctlr = 0;                /* normal */
    msgorder_c0ctlr = 1;                /* byte access */
    basiccan_c0ctlr = 1;                /* 0:normal mode 1:BasicCAN mode */
    buserren_c0ctlr = 0;                /* nus error interrupt disable */
    tsprescale_c0ctlr = 0;

    set_bitrate_can0();                 /* set bit rate */
    set_mask_can0();                    /* mask */

    reset_c0ctlr = 0;
    while(state_reset_c0str){}
}


 /*""FUNC COMMENT""*****************************************************
    *ID                  :2.0.3
    *Description         :Set can mask
    *-------------------------------------------------------------------
    *Include             :"sfr29.h"
    *                    :"define1.h"
    *                    :"declare1.h"
    *-------------------------------------------------------------------
    *Declaration         :void set_mask_can0(void)
    *-------------------------------------------------------------------
    *Function            :Set can mask
    *-------------------------------------------------------------------
    *Arguments           :Nothing
    *-------------------------------------------------------------------
    *Returns             :Nothing
```

```
   *-------------------------------------------------------------
   *Input                :Nothing
   *Ountput              :Nothing
   *-------------------------------------------------------------
   *Call functions     :Nothing
   *-------------------------------------------------------------
   *Note                 :
   *-------------------------------------------------------------
   *History              :
   *                     :
   *""FUNC COMMENT END""***********************************************/
void set_mask_can0(void)
{
    c0lmar.ba.sidh = (MASK_14>>6) & 0x1f;              //SID10-6
    c0lmar.ba.sidl = MASK_14 & 0x3f;                   //SID5-0

    c0lmbr.ba.sidh = (MASK_15>>6) & 0x1f;              //SID10-6
    c0lmbr.ba.sidl = MASK_15 & 0x3f;                   //SID5-0
}

 /*""FUNC COMMENT""*****************************************************
   *ID                   :2.2
   *Description          :Get can data
   *-------------------------------------------------------------
   *Include              :"sfr29.h"
   *                     :"define1.h"
   *                     :"declare1.h"
   *-------------------------------------------------------------
   *Declaration          :void get_message_can0
   *                        (unsigned short in_slot,can_std_data_def *in_rec_data)
   *-------------------------------------------------------------
   *Function             :Get can data
   *-------------------------------------------------------------
   *Arguments             :unsigned short in_slot            ;slot number
   *                     :can_std_data_def *in_rec_data   ;receive data pointer
   *-------------------------------------------------------------
   *Returns              :Nothing
   *-------------------------------------------------------------
   *Input                :Nothing
   *Ountput              :Nothing
   *-------------------------------------------------------------
   *Call functions     :Nothing
   *-------------------------------------------------------------
   *Note                 :
   *-------------------------------------------------------------
   *History              :
   *                     :
   *""FUNC COMMENT END""***********************************************/
void get_message_can0(
    unsigned short in_slot,
    can_std_data_def *in_rec_data)
{
    unsigned char lp_dlc;
```

```
        while(1){
            /* check receive complete */
            while(c0mctl[in_slot].receive.invaldata){}

            c0mctl[in_slot].receive.newdata = 0;

            /* overrun check */
            if(c0mctl[in_slot].receive.msglost){
                c0mctl[in_slot].receive.msglost = 0;
                error_fnc(CAN_RCV_BUFF_ERROR);
                break;
            }

            /* read message */
            in_rec_data->id = ((unsigned short)c0slot[in_slot].ba.sidh<<6)
                                        + c0slot[in_slot].ba.sidl;
            in_rec_data->dlc = c0slot[in_slot].ba.dlc;
            for(lp_dlc=0; lp_dlc<c0slot[in_slot].ba.dlc; ++lp_dlc){
                in_rec_data->data.data[lp_dlc] = c0slot[in_slot].ba.data[lp_dlc];
            }

            /* check new message */
            if(! c0mctl[in_slot].receive.newdata){
                break;
            }
        }
    }


 /*""FUNC COMMENT""*****************************************************
  *ID                 :2.3
  *Description        :Set can slot transmit
  *-------------------------------------------------------------------
  *Include            :"sfr29.h"
  *                   :"define1.h"
  *                   :"declare1.h"
  *-------------------------------------------------------------------
  *Declaration        :void set_trm_std_dataframe_can0
  *                        (unsigned short in_slot,can_std_data_def *in_trm_data)
  *-------------------------------------------------------------------
  *Function           :Set can slot transmit
  *-------------------------------------------------------------------
  *Arguments           :unsigned short in_slot          ;slot number
  *                   :can_std_data_def *in_trm_data   ;transmit data pointer
  *-------------------------------------------------------------------
  *Returns            :Nothing
  *-------------------------------------------------------------------
  *Input              :Nothing
  *Ountput            :Nothing
  *-------------------------------------------------------------------
  *Call functions    :Nothing
  *-------------------------------------------------------------------
  *Note               :
  *-------------------------------------------------------------------
```

```
   *History              :
   *                     :
   *""FUNC COMMENT END""***************************************************/
  void set_trm_std_dataframe_can0(
       unsigned short in_slot,
       can_std_data_def *in_trm_data)
  {
       unsigned char lp_dlc;

       while(c0mctl[in_slot].transmit.trmactive){}

       while(c0mctl[in_slot].byte != 0x00){
            c0mctl[in_slot].byte = 0x00;
       }

    /* transmission */
       c0icr &= ~(0x0001 << in_slot);
       c0idr &= ~(0x0001 << in_slot);

  //------------- set dataframe     ----------------
       c0slot[in_slot].ba.sidh = ((in_trm_data->id)>>6) & 0x1f;
                                                              //SID10-6
       c0slot[in_slot].ba.sidl = (in_trm_data->id) & 0x3f;
                                                              //SID5-0
       c0slot[in_slot].ba.dlc = in_trm_data->dlc;         //DLC

       for(lp_dlc = 0;lp_dlc<(in_trm_data->dlc) && (lp_dlc<8) ; lp_dlc++){
            c0slot[in_slot].ba.data[lp_dlc] = in_trm_data->data.data[lp_dlc];
                                                              //Data
       }
  //-----------------------------------------------
       c0mctl[in_slot].byte = 0x80;
  }


    /*""FUNC COMMENT""*****************************************************
     *ID                   :2.4
     *Description          :Abort can transmit
     *--------------------------------------------------------------------
     *Include              :"sfr29.h"
     *                     :"define1.h"
     *                     :"declare1.h"
     *--------------------------------------------------------------------
     *Declaration          :void abort_trm_can0(unsigned short in_slot)
     *--------------------------------------------------------------------
     *Function             :Abort can transmit
     *--------------------------------------------------------------------
     *Arguments            :unsigned short in_slot            ;slot number
     *--------------------------------------------------------------------
     *Returns              :Nothing
     *--------------------------------------------------------------------
     *Input                :Nothing
     *Ountput              :Nothing
     *--------------------------------------------------------------------
```

```
   *Call functions     :Nothing
   *-------------------------------------------------------------------
   *Note                :
   *-------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""***********************************************/
void abort_trm_can0(unsigned short in_slot)
{
     if(c0mctl[in_slot].transmit.trmreq
          || c0mctl[in_slot].receive.recreq){

          c0mctl[in_slot].byte = 0;
          while(c0mctl[in_slot].transmit.trmactive){}
     }
}


 /*""FUNC COMMENT""*****************************************************
   *ID                  :2.5
   *Description         :Transmit A/D data
   *-------------------------------------------------------------------
   *Include             :"sfr29.h"
   *                    :"define1.h"
   *                    :"declare1.h"
   *-------------------------------------------------------------------
   *Declaration         :void trm_can_ad(void)
   *-------------------------------------------------------------------
   *Function            :Transmit A/D data
   *-------------------------------------------------------------------
   *Arguments           :Nothing
   *-------------------------------------------------------------------
   *Returns             :Nothing
   *-------------------------------------------------------------------
   *Input               :trm_id              ;Transmint ID
   *                    :last_ad_data        ;A/D data
   *Ountput             :Nothing
   *-------------------------------------------------------------------
   *Call functions     :set_trm_std_dataframe_can0()    ;Set can slot transmit
   *-------------------------------------------------------------------
   *Note                :
   *-------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""***********************************************/
void trm_can_ad(void)
{
     can_std_data_def in_send_data;
/* ID */
     in_send_data.id = trm_id;
/* DLC */
     in_send_data.dlc = 1;
/* data */
     in_send_data.data.data[0] = last_ad_data;
```

```
/* transmission */
    set_trm_std_dataframe_can0(TRM_SLOT,&in_send_data);
}
```

```
/*""FILE COMMENT""*****************************************************
 *System Name : CAN DEMO PROGRAM1
 *File Name    : ad1.c
 *Version      : 1.00
 *Contens      :
 *Customer     : RSO
 *Model        :
 *Order        :
 *CPU          :
 *Compiler     : NC30 Version 5.30 Release 02
 *OS           :
 *Programmer   :
 *Note         :
 *             :
 ************************************************************************
 * Copyright(C)2005, Renesas Technology Corp.
 * Copyright(C)2005, Renesas Solutions Corp.
 * All rights reserved.
 ************************************************************************
 *History          2005.07.1   Ver 1.00
 *""FILE COMMENT END""*************************************************/

 #include "sfr29.h"
 #include "define1.h"
 #include "declare1.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :3.0
  *Description         :Initialize A/D converter
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"declare1.h"
  *--------------------------------------------------------------------
  *Declaration         :void ad_initial(void)
  *--------------------------------------------------------------------
  *Function            :Initialize A/D converter
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :Nothing
  *--------------------------------------------------------------------
  *Call functions     :Nothing
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
 void ad_initial(void)
 {
```

```
    unsigned short i;
    /* AD initial */
    adcon0 = 0x08;          //fAD/4 (5MHz)
    adcon1 = 0x28;          //
    adcon2 = 0x01;          //

    for(i = 0; i < 1000/50; i++);          //wait Vref(1us)

    adst = 1;               //start AD

    for(i = 0; i < 1000/50; i++);          //wait first AD conversion
}

 /*""FUNC COMMENT""*****************************************************
  *ID                    :3.1
  *Description           :A/D converter function
  *---------------------------------------------------------------------
  *Include               :"sfr29.h"
  *                      :"declare1.h"
  *---------------------------------------------------------------------
  *Declaration           :void ad_fnc(void)
  *---------------------------------------------------------------------
  *Function              :A/D converter function
  *---------------------------------------------------------------------
  *Arguments             :Nothing
  *---------------------------------------------------------------------
  *Returns               :Nothing
  *---------------------------------------------------------------------
  *Input                 :Nothing
  *Ountput               :Nothing
  *---------------------------------------------------------------------
  *Call functions        :ad_read()              ;Read A/D conversion value
  *                      :ad_check()             ;Check A/D conversion value
  *---------------------------------------------------------------------
  *Note                  :
  *---------------------------------------------------------------------
  *History               :
  *                      :
  *""FUNC COMMENT END""************************************************/
void ad_fnc(void)
{
    if(ad_read() != 0){
        if(ad_check() != 0){
            trm_can_ad();
        }
    }
}

 /*""FUNC COMMENT""*****************************************************
  *ID                    :3.2
  *Description           :Read A/D conversion value
  *---------------------------------------------------------------------
  *Include               :"sfr29.h"
  *                      :"define1.h"
```

```
  *                    :"declare1.h"
  *---------------------------------------------------------------
  *Declaration         :void ad_fnc(void)
  *---------------------------------------------------------------
  *Function            :Read A/D conversion value
  *---------------------------------------------------------------
  *Arguments           :Nothing
  *---------------------------------------------------------------
  *Returns             : Renewal code
  *                    :   0; Not renew "ave_ad_data"
  *                    :   1; Renew "ave_ad_data"
  *---------------------------------------------------------------
  *Input               :unsigned char count_ad          ;A/D counter
  *                    :unsigned char ad_data[]         ;A/D conversion data
  *Ountput             :unsigned char count_ad          ;A/D counter
  *                    :unsigned short ave_ad_data      ;Average of A/D conversion data
  *                    :unsigned char ad_data[]         ;A/D conversion data
  *---------------------------------------------------------------
  *Call functions      :Nothing
  *---------------------------------------------------------------
  *Note                :
  *---------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""***********************************************/
unsigned char ad_read(void)
{
    unsigned long in_ad_sum;
    unsigned short i;

    ad_data[count_ad] = ad0;
    count_ad++;

    if(count_ad >= AD_COUNT){          //Calculate A/D average
        in_ad_sum = 0;
        for(i = 0; i < (sizeof(ad_data) / 2); i++){
            in_ad_sum += ad_data[i];
        }
        ave_ad_data = in_ad_sum / (sizeof(ad_data) / 2);
        count_ad = 0;
        return 1;
    }

    return 0;
}

 /*""FUNC COMMENT""*****************************************************
  *ID                  :3.3
  *Description         :Check A/D conversion value
  *---------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *---------------------------------------------------------------
```

```
    *Declaration         :unsigned char ad_check(void)
    *-------------------------------------------------------------------
    *Function            :Check whether A/D conversion value differ from
    *                     last A/D conversion value
    *-------------------------------------------------------------------
    *Arguments           :Nothing
    *-------------------------------------------------------------------
    *Returns             : Difference code
    *                    :   0; No difference
    *                    :   1; Differing
    *-------------------------------------------------------------------
    *Input               :unsigned char last_ad_data       ;Last A/D countesion data
    *                    :unsigned char ave_ad_data        ;Average of A/D conversion data
    *Ountput             :unsigned char last_ad_data       ;Last A/D countesion data
    *-------------------------------------------------------------------
    *Call functions      :Nothing
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""***********************************************/
unsigned char ad_check(void)
{
    union {
        unsigned short word;
        unsigned char byte;
    }in_ad_data,in_last_ad_data;

    in_last_ad_data.byte = last_ad_data;
    in_last_ad_data.word = in_last_ad_data.word << 2;

    in_ad_data.word = ave_ad_data;

    if(in_last_ad_data.word > in_ad_data.word){
        if((in_last_ad_data.word - in_ad_data.word) > 0x0005){
            in_ad_data.word = (in_ad_data.word + 0x0004) >> 2;
            last_ad_data = in_ad_data.byte;
            return 1;
        }
        else if((in_last_ad_data.word == 0x0004) &&
            (in_ad_data.word <= 0x0001)){
//          in_ad_data.word = in_ad_data.word >> 2;
//          last_ad_data = in_ad_data.byte;
            last_ad_data = 0x00;
            return 1;
        }
    }
    else{
        if((in_ad_data.word - in_last_ad_data.word) > 0x0005){
            in_ad_data.word = in_ad_data.word >> 2;
            last_ad_data = in_ad_data.byte;
            return 1;
        }
```

```
        }

        return 0;
    }
```

```
/*""FILE COMMENT""*****************************************************
 *System Name : CAN DEMO PROGRAM1
 *File Name    : sw_led1.c
 *Version      : 1.00
 *Contens      :
 *Customer     : RSO
 *Model        :
 *Order        :
 *CPU          :
 *Compiler     : NC30 Version 5.30 Release 02
 *OS           :
 *Programmer   :
 *Note         :
 *             :
 ************************************************************************
 * Copyright(C)2005, Renesas Technology Corp.
 * Copyright(C)2005, Renesas Solutions Corp.
 * All rights reserved.
 *
 ************************************************************************
 *History          2005.07.1   Ver 1.00
 *""FILE COMMENT END""*************************************************/

 #include "sfr29.h"
 #include "define1.h"
 #include "declare1.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :4.0
  *Description         :Initialize Switch and LED
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *--------------------------------------------------------------------
  *Declaration         :void sw_led_initial(void)
  *--------------------------------------------------------------------
  *Function            :Initialize Switch and LED
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :unsigned char F_RCV_DATA        ;Receive data display enable flag
  *                    :unsigned char trm_count_sw      ;Transmit switch counter
  *                    :unsigned char led_mode          ;LED mode
  *                    :unsigned char last_led_mode     ;Last LED mode
  *                    :unsigned char led_data[]        ;LED display data
  *                    :unsigned char sw_count          ;Switch counter
  *                    :unsigned char last_sw_data      ;Last switch data
  *--------------------------------------------------------------------
  *Call functions      :Nothing
```

```
   *-------------------------------------------------------------------
   *Note                :
   *-------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""************************************************/
  void sw_led_initial(void)
  {
      unsigned char i;
      /* LED port initial */
      p0   = 0x03;
      p2   = 0xff;
      pd0 = 0x03;
      p0   = 0x03;
      pd2 = 0xff;
      p2   = 0xff;

      /* SW port initial */
      pd8_2 = 0;                          //SW2(INT0)
      pd8_3 = 0;                          //SW3(INT1)
      pd8_4 = 0;                          //SW4(INT2)

      /* variable initial */
      F_RCV_DATA = 1;                       //receive data disp enable
      trm_count_sw = 0;
      led_mode = MODE_CLWS;                  //set LED mode
      last_led_mode = MODE_CLWS;            //
      led_data[0] = LED_PT1_TBL[0];     //set initial LED dsp
      led_data[1] = LED_PT2_TBL[0];     //set initial LED dsp
      sw_count = 0;
      last_sw_data = 0;
  }

   /*""FUNC COMMENT""*****************************************************
   *ID                  :4.1
   *Description         :LED dynamic lightning
   *-------------------------------------------------------------------
   *Include             :"sfr29.h"
   *                    :"define1.h"
   *                    :"declare1.h"
   *-------------------------------------------------------------------
   *Declaration         :void led_fnc(void)
   *-------------------------------------------------------------------
   *Function            :LED dynamic lightning
   *-------------------------------------------------------------------
   *Arguments           :Nothing
   *-------------------------------------------------------------------
   *Returns             :Nothing
   *-------------------------------------------------------------------
   *Input               :unsigned char led_data[]    ;LED display data
   *                    :unsigned char led_count     ;LED counter
   *Ountput             :unsigned char led_count     ;LED counter
   *-------------------------------------------------------------------
   *Call functions      :Nothing
```

```
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""************************************************/
  void led_fnc(void)
  {
  /* LED */
        p0 = 0x03;                          //LED OFF

        led_count++;
        if(led_count > 1){
              led_count = 0;
        }

        p2 = led_data[led_count];          //set led data

        p0 = 0x01 << led_count;             //LED ON
  }

   /*""FUNC COMMENT""*****************************************************
    *ID                  :4.2
    *Description         :Change LED display pattern
    *-------------------------------------------------------------------
    *Include             :"sfr29.h"
    *                    :"define1.h"
    *                    :"declare1.h"
    *-------------------------------------------------------------------
    *Declaration         :void led_pt_fnc(void)
    *-------------------------------------------------------------------
    *Function            :Change LED display pattern
    *-------------------------------------------------------------------
    *Arguments           :Nothing
    *-------------------------------------------------------------------
    *Returns             :Nothing
    *-------------------------------------------------------------------
    *Input               :unsigned char led_mode          ;LED mode
    *                    :unsigned char led_pt_count      ;LED display pattern counter
    *Ountput             :unsigned char led_data[]        ;LED display data
    *                    :unsigned char led_pt_count      ;LED display pattern counter
    *-------------------------------------------------------------------
    *Call functions      :error_fnc()              ;Error function
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""************************************************/
  void led_pt_fnc(void)
  {
        switch (led_mode){
              case MODE_CLWS:
                    led_pt_count++;
```

```
            if(led_pt_count >= sizeof(LED_PT1_TBL)){
                led_pt_count = 0;
            }
            led_data[0] = LED_PT1_TBL[led_pt_count];
            led_data[1] = LED_PT2_TBL[led_pt_count];
            break;

        case MODE_ANTCLWS:
            led_pt_count--;
            if(led_pt_count >= sizeof(LED_PT1_TBL)){
                led_pt_count = sizeof(LED_PT1_TBL)-1;
            }
            led_data[0] = LED_PT1_TBL[led_pt_count];
            led_data[1] = LED_PT2_TBL[led_pt_count];
            break;

        case MODE_HALF_CLWS:
            led_pt_count++;
            if(led_pt_count >= sizeof(LED_PT5_TBL)){
                led_pt_count = 0;
            }
            led_data[0] = LED_PT5_TBL[led_pt_count];
            led_data[1] = LED_PT6_TBL[led_pt_count];
            break;
        case MODE_RCV_ID:
            break;
        case MODE_TRM_ID:
            break;
        default:
            error_fnc(LED_ERROR);
            break;
    }
}

 /*""FUNC COMMENT""*****************************************************
  *ID                 :4.3
  *Description        :Change LED mode
  *-------------------------------------------------------------------
  *Include            :"sfr29.h"
  *                   :"define1.h"
  *                   :"declare1.h"
  *-------------------------------------------------------------------
  *Declaration        :void led_mode_change(void)
  *-------------------------------------------------------------------
  *Function           :Change LED mode
  *-------------------------------------------------------------------
  *Arguments          :Nothing
  *-------------------------------------------------------------------
  *Returns            :Nothing
  *-------------------------------------------------------------------
  *Input              :unsigned char F_RCV_DATA       ;Receive data display enable flag
  *                   :unsigned char led_mode         ;LED mode
  *                   :unsigned char last_rcv_data    ;Last CAN receive data
  *Ountput            :unsigned char led_data[]       ;LED display data
```

```
    *                   :unsigned char led_mode          ;LED mode
    *                   :unsigned char last_led_mode     ;Last LED mode
    *-------------------------------------------------------------------
    *Call functions     :error_fnc()                ;Error function
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""***********************************************/
  void led_mode_change(void)
  {
      unsigned char in_last_rcv_data;

      if(F_RCV_DATA == 0){                      //receive data disable?
          return;
      }

      in_last_rcv_data = last_rcv_data;

      switch (in_last_rcv_data){
          case DATA_CLWS:
              if(led_mode != MODE_CLWS){        //mode change clockwise
                  led_mode = MODE_CLWS;
                  last_led_mode = led_mode;
                  ta0 = DATA_CLWS_TM;
              }
              break;

          case DATA_ANTCLWS:
              if(led_mode != MODE_ANTCLWS){     //mode change anticlockwise
                  led_mode = MODE_ANTCLWS;
                  last_led_mode = led_mode;
                  ta0 = DATA_ANTCLWS_TM;
              }
              break;

          case DATA_HALF_CLWS:
              if(led_mode != MODE_HALF_CLWS){ //mode change half clockwise
                  led_mode = MODE_HALF_CLWS;
                  last_led_mode = led_mode;
                  ta0 = DATA_HALF_CLWS_TM;
              }
              break;

          default:
              break;
      }
  }


   /*""FUNC COMMENT""***************************************************
    *ID                  :4.4
    *Description         :Switch function
```

```
    *-------------------------------------------------------------------
    *Include              :"sfr29.h"
    *                     :"define1.h"
    *                     :"declare1.h"
    *-------------------------------------------------------------------
    *Declaration          :void sw_fnc(void)
    *-------------------------------------------------------------------
    *Function             :Switch function
    *-------------------------------------------------------------------
    *Arguments            :Nothing
    *-------------------------------------------------------------------
    *Returns              :Nothing
    *-------------------------------------------------------------------
    *Input                :Nothing
    *Ountput              :Nothing
    *-------------------------------------------------------------------
    *Call functions       :error_fnc()                ;Error function
    *                     :sw_2()                      ;Switch2 function
    *                     :sw_3()                      ;Switch3 function
    *                     :sw_4()                      ;Switch4 function
    *                     :sw_3_4()                    ;Switch3_4 function
    *                     :sw_4_3()                    ;Switch4_3 function
    *                     :sw_other()                  ;Other switch function
    *                     :fix_sw_fnc()                ;Fix swithc function
    *                     :sw_decode()                 ;Swithc decode function
    *-------------------------------------------------------------------
    *Note                 :
    *-------------------------------------------------------------------
    *History              :
    *                     :
    *""FUNC COMMENT END""***************************************************/
  void sw_fnc(void)
  {
      unsigned char in_data,i;

      in_data = p8 & PT_SW_MASK;
      if(fix_sw_fnc(in_data)){
          i = sw_decode();
          if(i < SW_TBL_SIZE){     //table check
              (*SW_TBL[i])();      //call sw fnc
          }
          else{
              error_fnc(TABLE_ERROR);
          }
      }
  }


   /*""FUNC COMMENT""*****************************************************
    *ID                   :4.4.1
    *Description          :Switch decode function
    *-------------------------------------------------------------------
    *Include              :"define1.h"
    *                     :"declare1.h"
```

```
    *-------------------------------------------------------------------
    *Declaration         :unsigned char sw_decode(void)
    *-------------------------------------------------------------------
    *Function            :Switch decode function
    *-------------------------------------------------------------------
    *Arguments           :Nothing
    *-------------------------------------------------------------------
    *Returns             :unsigned char in_rtn              ;Decode number
    *-------------------------------------------------------------------
    *Input               :unsigned char fix_sw_data         ;Fixed switch data
    *                    :unsigned char last_fix_sw_data ;Last fixed switch data
    *Ountput             :unsigned char last_fix_sw_data ;Last fixed switch data
    *-------------------------------------------------------------------
    *Call functions      :error_fnc()                       ;Error function
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""***********************************************/
unsigned char sw_decode(void)
{
    unsigned char in_rtn = 0;

    switch(fix_sw_data){
        case PT_NO_SW:        //no SW
            in_rtn = 0;
            break;
        case PT_SW2:          //SW2
            if(last_fix_sw_data == PT_NO_SW){
                in_rtn = 1;
            }
            else{
                in_rtn = 6;
            }
            break;
        case PT_SW3:          //SW3
            in_rtn = 2;
            break;
        case PT_SW4:          //SW4
            in_rtn = 3;
            break;
        case PT_SW34:         //SW3 & SW4
            if(last_fix_sw_data == PT_SW3){
                in_rtn = 4; //SW3 --> SW4
            }
            else if(last_fix_sw_data == PT_SW4){
                in_rtn = 5; //SW4 --> SW3
            }
            else{
                in_rtn = 6; //
            }
            break;
        default:
```

```
            in_rtn = 6;
            break;
    }

    last_fix_sw_data = fix_sw_data;
    return in_rtn;
}


 /*""FUNC COMMENT""*****************************************************
  *ID                :4.4.2
  *Description       :Fix switch function
  *-------------------------------------------------------------------
  *Include           :"define1.h"
  *                  :"declare1.h"
  *-------------------------------------------------------------------
  *Declaration       :unsigned char fix_sw_fnc(unsigned char)
  *-------------------------------------------------------------------
  *Function          :Fix switch function
  *-------------------------------------------------------------------
  *Arguments          :unsigned char in_data          ;Switch data
  *-------------------------------------------------------------------
  *Returns           :uneigned char in_rtn           ;Fixed switch data
  *-------------------------------------------------------------------
  *Input             :unsigned char fix_sw_data      ;Fixed switch data
  *                  :unsigned char last_sw_data     ;Last switch data
  *                  :unsigned char sw_count         ;Switch counter
  *Ountput           :unsigned char trm_count_sw     ;Transmit counter
  *-------------------------------------------------------------------
  *Call functions    :error_fnc()                    ;Error function
  *-------------------------------------------------------------------
  *Note              :
  *-------------------------------------------------------------------
  *History           :
  *                  :
  *""FUNC COMMENT END""************************************************/
unsigned char fix_sw_fnc(unsigned char in_data)
{
    unsigned char in_rtn = 0;

    if((fix_sw_data != in_data) && (last_sw_data == in_data)){
        sw_count++;
    }

    if(sw_count > CHAT_COUNT){
        fix_sw_data = in_data;
        sw_count = 0;
        in_rtn = 1;
    }

    last_sw_data = in_data;

    return in_rtn;
}
```

```
 /*""FUNC COMMENT""*****************************************************
  *ID                  :4.4.3
  *Description         :Switch2 function
  *-------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *-------------------------------------------------------------------
  *Declaration         :void sw_2(void)
  *-------------------------------------------------------------------
  *Function            :Switch2 function
  *-------------------------------------------------------------------
  *Arguments           :Nothing
  *-------------------------------------------------------------------
  *Returns             :Nothing
  *-------------------------------------------------------------------
  *Input               :unsigned char trm_id              ;Transmit ID
  *                    :unsigned char trm_data_sw[]       ;Transmit data
  *                    :unsigned char trm_count_sw        ;Transmit counter
  *Ountput             :unsigned char trm_count_sw        ;Transmit counter
  *-------------------------------------------------------------------
  *Call functions      :error_fnc()                     ;Error function
  *-------------------------------------------------------------------
  *Note                :
  *-------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
void sw_2(void)
{
    can_std_data_def in_send_data;

    /* ID */
    in_send_data.id = trm_id;
    /* DLC */
    in_send_data.dlc = 1;
    /* data */
    in_send_data.data.data[0] = trm_data_sw[trm_count_sw];
    /* transmission */
    set_trm_std_dataframe_can0(TRM_SLOT,&in_send_data);       //Transmit CAN

    trm_count_sw++;
    if(trm_count_sw >= sizeof(trm_data_sw)){
        trm_count_sw = 0;
    }
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :4.4.4
  *Description         :Switch3 function
  *-------------------------------------------------------------------
```

```
  *Include              :"define1.h"
  *                     :"declare1.h"
  *-------------------------------------------------------------------
  *Declaration          :void sw_3(void)
  *-------------------------------------------------------------------
  *Function             :Switch3 function
  *-------------------------------------------------------------------
  *Arguments            :Nothing
  *-------------------------------------------------------------------
  *Returns              :Nothing
  *-------------------------------------------------------------------
  *Input                :unsigned char rcv_id              ;Receive ID
  *Ountput              :unsigned char F_RCV_DATA          ;Receive data display enable flag
  *                     :unsigned char led_data[]          ;LED display data
  *                     :unsigned char led_mode            ;LED mode
  *-------------------------------------------------------------------
  *Call functions       :
  *-------------------------------------------------------------------
  *Note                 :
  *-------------------------------------------------------------------
  *History              :
  *                     :
  *""FUNC COMMENT END""************************************************/
void sw_3(void)
{
    F_RCV_DATA = 0;                         //receive data disable

    led_mode = MODE_RCV_ID;                 //Display rceive ID
    led_data[0] = LED_TBL[0x0f & rcv_id];
    led_data[1] = LED_TBL[rcv_id >> 4];
}


 /*""FUNC COMMENT""*****************************************************
  *ID                   :4.4.5
  *Description          :Switch4_3 function
  *-------------------------------------------------------------------
  *Include              :"define1.h"
  *                     :"declare1.h"
  *-------------------------------------------------------------------
  *Declaration          :void sw_4_3(void)
  *-------------------------------------------------------------------
  *Function             :Switch4_3 function
  *-------------------------------------------------------------------
  *Arguments            :Nothing
  *-------------------------------------------------------------------
  *Returns              :Nothing
  *-------------------------------------------------------------------
  *Input                :unsigned char trm_id              ;Transmit ID
  *Ountput              :unsigned char F_RCV_DATA          ;Receive data display enable flag
  *                     :unsigned char trm_id              ;Transmit ID
  *                     :unsigned char led_data[]          ;LED display data
  *                     :unsigned char led_mode            ;LED mode
  *-------------------------------------------------------------------
```

```
   *Call functions     :
   *-------------------------------------------------------------------
   *Note                :
   *-------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""***********************************************/
void sw_4_3(void)
{
      F_RCV_DATA = 0;                          //receive data disable
      trm_id++;                                //Add transmit ID
/*    if(trm_id > 0x03ff){                     //"trm_id" = 8bit
          trm_id = 0;
      }
*/
      led_data[0] = LED_TBL[0x0f & trm_id];    //Display transmit ID
      led_data[1] = LED_TBL[trm_id >> 4];
}


 /*""FUNC COMMENT""*****************************************************
   *ID                  :4.4.6
   *Description         :Switch4 function
   *-------------------------------------------------------------------
   *Include             :"define1.h"
   *                    :"declare1.h"
   *-------------------------------------------------------------------
   *Declaration         :void sw_4(void)
   *-------------------------------------------------------------------
   *Function            :Switch4 function
   *-------------------------------------------------------------------
   *Arguments           :Nothing
   *-------------------------------------------------------------------
   *Returns             :Nothing
   *-------------------------------------------------------------------
   *Input               :unsigned char trm_id          ;Transmit ID
   *Ountput             :unsigned char F_RCV_DATA      ;Receive data display enable flag
   *                    :unsigned char led_data[]      ;LED display data
   *                    :unsigned char led_mode        ;LED mode
   *-------------------------------------------------------------------
   *Call functions     :
   *-------------------------------------------------------------------
   *Note                :
   *-------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""***********************************************/
void sw_4(void)
{
      F_RCV_DATA = 0;                          //receive data disable
      led_mode = MODE_TRM_ID;
      led_data[0] = LED_TBL[0x0f & trm_id];    //Display transmit ID
      led_data[1] = LED_TBL[trm_id >> 4];
}
```

```
 /*""FUNC COMMENT""*****************************************************
  *ID                  :4.4.7
  *Description         :Switch3_4 function
  *-------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *-------------------------------------------------------------------
  *Declaration         :void sw_3_4(void)
  *-------------------------------------------------------------------
  *Function            :Switch3_4 function
  *-------------------------------------------------------------------
  *Arguments           :Nothing
  *-------------------------------------------------------------------
  *Returns             :Nothing
  *-------------------------------------------------------------------
  *Input               :unsigned char rcv_id            ;Receive ID
  *Ountput             :unsigned char F_RCV_DATA        ;Receive data display enable flag
  *                    :unsigned char rcv_id            ;Receive ID
  *                    :unsigned char led_data[]        ;LED display data
  *                    :unsigned char led_mode          ;LED mode
  *-------------------------------------------------------------------
  *Call functions      :abort_trm_can0()                ;Abort CAN transmit
  *-------------------------------------------------------------------
  *Note                :
  *-------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""*************************************************/
void sw_3_4(void)
{
    unsigned short in_id;

    F_RCV_DATA = 0;                                  //receive data disable
        /* When transmitting the message, stop transmitting */
        abort_trm_can0(TRM_SLOT);

        if((c0mctl14.receive.invaldata == 0)
         && (c0mctl15.receive.invaldata == 0)){   //Check receiving

            /* Stop receiveing the message */
            c0mctl14.byte = 0x00;
            c0mctl15.byte = 0x00;

            rcv_id++;                                //Add receive ID
/*          if(rcv_id > 0x03ff){                     //"trm_id" = 8bit
                rcv_id = 0;
            }
*/
            in_id = rcv_id;

            /* Set new ID and start rereceiving */
```

```
                set_rec_std_dataframe_can0(14,in_id);
                set_rec_std_dataframe_can0(15,in_id);

                /* Display receive ID */
                led_data[0] = LED_TBL[0x0f & rcv_id];
                led_data[1] = LED_TBL[rcv_id >> 4];
            }
    }

     /*""FUNC COMMENT""*****************************************************
      *ID                  :4.4.8
      *Description         :Other switch function
      *--------------------------------------------------------------------
      *Include             :"declare1.h"
      *--------------------------------------------------------------------
      *Declaration         :void sw_other(void)
      *--------------------------------------------------------------------
      *Function            :Other switch function
      *--------------------------------------------------------------------
      *Arguments           :Nothing
      *--------------------------------------------------------------------
      *Returns             :Nothing
      *--------------------------------------------------------------------
      *Input               :Nothing
      *Ountput             :Nothing
      *--------------------------------------------------------------------
      *Call functions     :
      *--------------------------------------------------------------------
      *Note                :
      *--------------------------------------------------------------------
      *History             :
      *                    :
      *""FUNC COMMENT END""************************************************/
    void sw_other(void)
    {

    }

     /*""FUNC COMMENT""*****************************************************
      *ID                  :4.4.9
      *Description         :Return display
      *--------------------------------------------------------------------
      *Include             :"define1.h"
      *                    :"declare1.h"
      *--------------------------------------------------------------------
      *Declaration         :void dsp_return(void)
      *--------------------------------------------------------------------
      *Function            :Return display
      *--------------------------------------------------------------------
      *Arguments           :Nothing
      *--------------------------------------------------------------------
      *Returns             :Nothing
      *--------------------------------------------------------------------
      *Input               :unsigned char last_led_mode      ;Last LED mode
```

```
    *Ountput              :unsigned char led_mode          ;LED mode
    *                     :unsigned char F_RCV_DATA        ;Receive data display enable flag
    *-------------------------------------------------------------------
    *Call functions    :
    *-------------------------------------------------------------------
    *Note               :
    *-------------------------------------------------------------------
    *History            :
    *                   :
    *""FUNC COMMENT END""***********************************************/
void dsp_return(void)
{
    F_RCV_DATA = 1;              //receive data enable
    led_mode = last_led_mode;    //Return LED mode
}
```

```
/*""FILE COMMENT""*****************************************************
 *System Name : CAN DEMO PROGRAM1
 *File Name    : int1.c
 *Version      : 1.00
 *Contens      :
 *Customer     : RSO
 *Model        :
 *Order        :
 *CPU          :
 *Compiler     : NC30 Version 5.30 Release 02
 *OS           :
 *Programmer   :
 *Note         :
 ************************************************************************
 * Copyright(C)2005, Renesas Technology Corp.
 * Copyright(C)2005, Renesas Solutions Corp.
 * All rights reserved.
 *
 ************************************************************************
 *History          2005.07.1   Ver 1.00
 *""FILE COMMENT END""************************************************/

 #include "sfr29.h"
 #include "define1.h"
 #include "declare1.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :5.0
  *Description         :CAN0 receive interrupt
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define1.h"
  *                    :"declare1.h"
  *--------------------------------------------------------------------
  *Declaration         :void CAN0_REC(void)
  *--------------------------------------------------------------------
  *Function            :Read CAN0 receive data
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :unsigned char rec_data[]        ;CAN receive data
  *Ountput             :unsigned char last_rcv_data     ;Last CAN receive data
  *--------------------------------------------------------------------
  *Call functions      :get_message_can0()              ;Get CAN data
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
 void CAN0_REC(void)
```

```
{
    /* slot14 */
    if(c0mctl[14].receive.newdata == 1){
        get_message_can0(14, &rec_data[0]);          //Read receive data
        last_rcv_data = rec_data[0].data.data[0];
    }

    /* slot15 */
    if(c0mctl[15].receive.newdata == 1){
        get_message_can0(15, &rec_data[1]);          //Read receive data
        last_rcv_data = rec_data[1].data.data[0];
    }
}
```

## 6.2 Program (2)

```
/*""FILE COMMENT""*****************************************************
*System Name : CAN PROGRAM
*File Name    : define2.h
*Version      : 1.00
*Contens      :
*Customer     : RSO
*Model        :
*Order        :
*CPU          :
*Compiler     : NC30 Version 5.30 Release 02
*OS           :
*Programmer   :
*Note         :
*********************************************************************
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*********************************************************************
*History          2005.07.1    Ver 1.00
*""FILE COMMENT END""*************************************************/

/* CAN */
#define      MASK_14      0xff            //No mask
#define      MASK_15      0xff            //

#define      TRM_SLOT     0               //CAN0 trm slot
#define      TRM_DLC      1               //CAN0 trm dlc

#define      CAN_TRM_ID   2               //CAN0 trm initial ID
#define      CAN_RCV_ID   1               //CAN0 rcv initial ID

#define      DATA_CLWS         1          //Clockwise code
#define      DATA_ANTCLWS      2          //Anticlockwise code
#define      DATA_HALF_CLWS    3          //Half clockwise code

/* SW & LED */
#define      CHAT_COUNT   2               //

#define      SW_TBL_SIZE 4                //
#define      PT_SW_MASK   0b00011100      //SW MASK

#define      PT_ALL       0b00000000      //SW all
#define      PT_SW34      0b00000100      //SW3 & SW4
#define      PT_SW24      0b00001000      //SW2 & SW4
#define      PT_SW4       0b00001100      //SW4
#define      PT_SW23      0b00010000      //SW2 & SW3
#define      PT_SW3       0b00010100      //SW3
#define      PT_SW2       0b00011000      //SW2
#define      PT_NO_SW     0b00011100      //no SW
```

```
/* timer */
#define     DATA_TA1     2000000/50 -1    //LED & SW (2ms)

/* error */
#define     NO_ERROR              0        //Error number
#define     CAN_RCV_BUFF_ERROR    1        //Error number
#define     CAN_TRM_ERROR         2        //Error number
#define     TABLE_ERROR           3        //Error number


/* interrupt Lv */
#define TA1_LVL           0                 //LED & SW
#define CAN0_LVL          5                 //CAN0
#define CAN0_ERROR_LVL    0                 //CAN0 error


/************************** LED DEFINE ********************************/
#define SEG_A    0x01
#define SEG_B    0x02
#define SEG_C    0x04
#define SEG_D    0x08
#define SEG_E    0x10
#define SEG_F    0x20
#define SEG_G    0x40
#define SEG_H    0x80
#define DISP_0   (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F              ))
#define DISP_1   (~(      SEG_B|SEG_C                                ))
#define DISP_2   (~(SEG_A|SEG_B|      SEG_D|SEG_E|      SEG_G       ))
#define DISP_3   (~(SEG_A|SEG_B|SEG_C|SEG_D|            SEG_G       ))
#define DISP_4   (~(      SEG_B|SEG_C|            SEG_F|SEG_G       ))
#define DISP_5   (~(SEG_A|      SEG_C|SEG_D|      SEG_F|SEG_G       ))
#define DISP_6   (~(SEG_A|      SEG_C|SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_7   (~(SEG_A|SEG_B|SEG_C|            SEG_F             ))
#define DISP_8   (~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_9   (~(SEG_A|SEG_B|SEG_C|SEG_D      |SEG_F|SEG_G       ))
#define DISP_A   (~(SEG_A|SEG_B|SEG_C|      SEG_E|SEG_F|SEG_G       ))
#define DISP_B   (~(            SEG_C|SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_C   (~(SEG_A|            SEG_D|SEG_E|SEG_F             ))
#define DISP_D   (~(      SEG_B|SEG_C|SEG_D|SEG_E|      SEG_G       ))
#define DISP_E   (~(SEG_A|            SEG_D|SEG_E|SEG_F|SEG_G       ))
#define DISP_F   (~(SEG_A|                  SEG_E|SEG_F|SEG_G       ))
/**********************************************************************/

typedef struct{
    unsigned short id;
    unsigned char dlc;
    union {
        unsigned char data[8];
        unsigned long long data_8;
    }data;
}can_std_data_def;
```

```
/*""FILE COMMENT""*****************************************************
*System Name : CAN PROGRAM
*File Name    : declare2.h
*Version      : 1.00
*Contens      :
*Customer     : RSO
*Model        :
*Order        :
*CPU          :
*Compiler     : NC30 Version 5.30 Release 02
*OS           :
*Programmer   :
*Note         :
*********************************************************************
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
*********************************************************************
*History          2005.07.1    Ver 1.00
*""FILE COMMENT END""*************************************************/

#ifdef DECLARE_MAIN
    #define      READTASK_EXTERN
#else
    #define      READTASK_EXTERN extern
#endif

/*********************************************************************
*                fnc
*********************************************************************/
READTASK_EXTERN void main(void);
READTASK_EXTERN void cpu_initial(void);
READTASK_EXTERN void timer_initial(void);
READTASK_EXTERN void sw_led_initial(void);

/* CAN */
READTASK_EXTERN void can_initial(void);
READTASK_EXTERN void set_bitrate_can0(void);
READTASK_EXTERN void config_can0(void);
READTASK_EXTERN void set_rec_std_dataframe_can0(unsigned short, unsigned short);
READTASK_EXTERN void set_mask_can0(void);
READTASK_EXTERN void get_message_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void set_trm_std_dataframe_can0(unsigned short, can_std_data_def *);
READTASK_EXTERN void abort_trm_can0(unsigned short in_slot);

/* SW & LED */
READTASK_EXTERN void led_fnc(void);
READTASK_EXTERN void sw_fnc(void);
READTASK_EXTERN void sw_down(unsigned char);
READTASK_EXTERN unsigned char fix_sw_fnc(unsigned char);
READTASK_EXTERN unsigned char sw_decode(void);
```

```
/* ERROR */
READTASK_EXTERN void error_fnc(unsigned char);
READTASK_EXTERN void error_main_fnc(void);

/************************************************************************
*                  interrupt fnc
************************************************************************/
#pragma INTERRUPT /B CAN0_REC()
READTASK_EXTERN void CAN0_REC(void);

/************************************************************************
*                  variable
************************************************************************/

/* CAN */
READTASK_EXTERN can_std_data_def rec_data[2];          //CAN receive data
READTASK_EXTERN unsigned char trm_id;                   //Transmit ID
READTASK_EXTERN unsigned char rcv_id;                  //Receive ID
READTASK_EXTERN unsigned char trm_led_data[3];         //Transmit data

/* LED */
READTASK_EXTERN unsigned char led_data[2];             //LED display data
READTASK_EXTERN unsigned char led_count;                //LED counter

/* SW */
READTASK_EXTERN unsigned char sw_count;                 //Switch counter
READTASK_EXTERN unsigned char last_sw_data;            //Last switch data
READTASK_EXTERN unsigned char fix_sw_data;             //Fixed switch data
READTASK_EXTERN unsigned char last_fix_sw_data;       //Last fixed switch data

/* ERROR */
READTASK_EXTERN unsigned char error_num;               //Error number


/************************************************************************
*                  table
************************************************************************/
/* LED display table */
READTASK_EXTERN const unsigned char LED_TBL[16]
#ifdef DECLARE_MAIN
      = {DISP_0,DISP_1,DISP_2,DISP_3,
         DISP_4,DISP_5,DISP_6,DISP_7,
         DISP_8,DISP_9,DISP_A,DISP_B,
         DISP_C,DISP_D,DISP_E,DISP_F}
#endif
     ;
```

```
/*""FILE COMMENT""*****************************************************
*System Name : CAN DEMO PROGRAM
*File Name    : main2.c
*Version      : 1.00
*Contens      :
*Customer     : RSO
*Model        :
*Order        :
*CPU          :
*Compiler     : NC30 Version 5.30 Release 02
*OS           :
*Programmer   :
*Note         :
*             :
************************************************************************
* Copyright(C)2005, Renesas Technology Corp.
* Copyright(C)2005, Renesas Solutions Corp.
* All rights reserved.
*
************************************************************************
*History          2005.07.1    Ver 1.00
*""FILE COMMENT END""*************************************************/

#define DECLARE_MAIN

#include "sfr29.h"
#include "define2.h"
#include "declare2.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :1.0
  *Description         :
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *--------------------------------------------------------------------
  *Declaration         :void main(void)
  *--------------------------------------------------------------------
  *Function            :main
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :unsigned char error_num          ;error number
  *Ountput             :unsigned char error_num          ;error number
  *--------------------------------------------------------------------
  *Call functions     :cpu_initial()                ;Initialize CPU
  *                    :sw_led_initial()            ;Initialize switch and LED
  *                    :can_initial()               ;Initialize CAN
  *                    :timer_initial()             ;Initialize timer
  *                    :led_fnc()                    ;LED dynamic lightning
```

```
   *                    :sw_fnc()                    ;Read switch port
   *-------------------------------------------------------------------
   *Note                 :
   *-------------------------------------------------------------------
   *History              :
   *                    :
   *""FUNC COMMENT END""************************************************/
void main(void)
{
      unsigned int i;

      asm("fclr I");

      cpu_initial();
      sw_led_initial();
      can_initial();
      timer_initial();

/* variable initial */
      error_num = NO_ERROR;              //set error flag "No error"


      ta1s = 1;                          //LED & SW timer
      asm("fset I");


      while(error_num == NO_ERROR){
            if(ir_ta1ic == 1){
                  ta1ic = TA1_LVL;          //IR clear

                  led_fnc();                //LED dynamic lightning
                  sw_fnc();                 //Switch function
            }
      }


/* ERROR */
      error_main_fnc();

      while(1);
}

 /*""FUNC COMMENT""*****************************************************
   *ID                   :1.1
   *Description          :Initialize CPU
   *-------------------------------------------------------------------
   *Include              :"sfr29.h"
   *-------------------------------------------------------------------
   *Declaration          :void cpu_initial(void)
   *-------------------------------------------------------------------
   *Function             :Initialize main clock,processer mode and ports
   *-------------------------------------------------------------------
   *Arguments            :Nothing
   *-------------------------------------------------------------------
```

```
    *Returns              :Nothing
    *-------------------------------------------------------------------
    *Input                :Nothing
    *Ountput              :Nothing
    *-------------------------------------------------------------------
    *Call functions     :Nothing
    *-------------------------------------------------------------------
    *Note                 :
    *-------------------------------------------------------------------
    *History              :
    *                     :
    *""FUNC COMMENT END""***************************************************/
  void cpu_initial(void)
  {
        while(pacr != 0x03){
              prcr = 0x04;
              pacr = 0x03;                    //80pin
        }

        prcr = 0x03;

        /* clock */
        cm1 = 0x20;
        cm2 = 0;
        pm0 = 0;
        pm1 = 0x08;
        cm0 = 0x08;
        pm2 = 0;

        /* port */
        p0    = 0x00;
        p1    = 0x00;
        p2    = 0x00;
        p3    = 0x00;
  #if !_KD_
        p6    = 0x00;
  #endif
        p7    = 0x00;
        p8    = 0x00;
        p9    = 0x00;
        p10    = 0x00;

        pd0 = 0x00;
        pd1 = 0x00;
        pd2 = 0x00;
        pd3 = 0x00;
  #if !_KD_
        pd6 = 0x00;
  #endif
        pd7 = 0x00;
        pd8 = 0x00;
        prcr = 0x07;
        pd9 = 0x00;
        pd10 = 0x00;
```

```
    prcr = 0;

    /* pull up control */
    pur0 = 0x00;
    pur1 = 0x00;
    pur2 = 0x00;

    /* port control */
    pcr = 0x00;
}


 /*""FUNC COMMENT""*****************************************************
  *ID                 :1.2
  *Description        :Initialize timer
  *-------------------------------------------------------------------
  *Include            :"sfr29.h"
  *                   :"define2.h"
  *-------------------------------------------------------------------
  *Declaration        :void timer_initial(void)
  *-------------------------------------------------------------------
  *Function           :Initialize TA1
  *-------------------------------------------------------------------
  *Arguments          :Nothing
  *-------------------------------------------------------------------
  *Returns            :Nothing
  *-------------------------------------------------------------------
  *Input              :Nothing
  *Ountput            :Nothing
  *-------------------------------------------------------------------
  *Call functions     :Nothing
  *-------------------------------------------------------------------
  *Note               :
  *-------------------------------------------------------------------
  *History            :
  *                   :
  *""FUNC COMMENT END""************************************************/
void timer_initial(void)
{
    tabsr = 0;                  //all timer stop
    onsf = 0;
    trgsr = 0;

    ta1mr = 0x00;                //timer mode
    ta1 = DATA_TA1;              //
    ta1ic = TA1_LVL;
}


 /*""FUNC COMMENT""*****************************************************
  *ID                 :1.3
  *Description        :Error process
  *-------------------------------------------------------------------
```

```
    *Include              :"sfr29.h"
    *-------------------------------------------------------------------
    *Declaration          :void error_fnc(void)
    *-------------------------------------------------------------------
    *Function             :Timer,A/D converter and CAN module are stop.
    *                     :LED off
    *-------------------------------------------------------------------
    *Arguments             :unsigned char in_num                ;Error number
    *-------------------------------------------------------------------
    *Returns              :Nothing
    *-------------------------------------------------------------------
    *Input                :Nothing
    *Ountput              :unsigned char error_num             ;Error number
    *-------------------------------------------------------------------
    *Call functions     :Nothing
    *-------------------------------------------------------------------
    *Note                 :
    *-------------------------------------------------------------------
    *History              :
    *                     :
    *""FUNC COMMENT END""***********************************************/
  void error_fnc(unsigned char in_num)
  {
      asm("fclr I");

      tabsr = 0;                    //all timer stop
      adcon0 = 0;                    //AD stop
      reset_c0ctlr = 1;             //CAN reset

      p0 = 0x03;                     //LED OFF

      error_num = in_num;
  }

   /*""FUNC COMMENT""*****************************************************
    *ID                   :1.4
    *Description          :Error process
    *-------------------------------------------------------------------
    *Include              :"sfr29.h"
    *-------------------------------------------------------------------
    *Declaration          :void error_main_fnc(void)
    *-------------------------------------------------------------------
    *Function             :Display error number
    *-------------------------------------------------------------------
    *Arguments             :Nothing
    *-------------------------------------------------------------------
    *Returns              :Nothing
    *-------------------------------------------------------------------
    *Input                :unsigned char error_num          ;Error number
    *                     :unsigned char led_count          ;LED counter
    *                     :unsigned char led_data[]         ;LED display data
    *Ountput               :unsigned char led_count          ;LED counter
    *                     :unsigned char led_data[]         ;LED display data
    *-------------------------------------------------------------------
```

```
    *Call functions     :sw_led_initial()                   ;Initialize switch and LED
    *                   :led_fnc()                            ;LED dynamic lightning
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""***********************************************/
void error_main_fnc(void)
{
    unsigned char in_led_on_off = 0;
    unsigned char i;

    asm("fclr I");

    tabsr = 0;                      //all timer stop
    reset_c0ctlr = 1;               //CAN reset

    sw_led_initial();

    ta0mr = 0x01;                     //ivent
    ta0tgl = 1;                     //count sorce:TA1
    ta0tgh = 1;                     //
    ta0 = 250 - 1;                  //
    ta0ic = 0x00;

    ta1mr = 0x00;                     //timer mode
    ta1 = 2000000/50 -1;            //2ms
    ta1ic = 0x00;

    led_count = 0;
    led_data[1] = LED_TBL[0x0e];            //Set error number
    led_data[0] = LED_TBL[error_num];     //

    tabsr = 0x03;                   //TA0,TA1 start

    while(1){
        if(in_led_on_off == 1){
            if(ir_ta1ic == 1){
                ta1ic = 0x00;

                led_fnc();
            }
        }

        if(ir_ta0ic == 1){          //LED flashing
            ta0ic = 0;

            if(in_led_on_off == 0){
                in_led_on_off = 1;   //LED ON
            }
            else{
                in_led_on_off = 0;   //LED OFF
                p0 = 0x03;
```

```
                }
            }
        }
    }
```

```
/*""FILE COMMENT""*****************************************************
 *System Name : CAN DEMO PROGRAM
 *File Name    : can2.c
 *Version      : 1.00
 *Contens      :
 *Customer     : RSO
 *Model        :
 *Order        :
 *CPU          :
 *Compiler     : NC30 Version 5.30 Release 02
 *OS           :
 *Programmer   :
 *Note         :
 *             :
 ************************************************************************
 * Copyright(C)2005, Renesas Technology Corp.
 * Copyright(C)2005, Renesas Solutions Corp.
 * All rights reserved.
 ************************************************************************
 *History          2005.07.1   Ver 1.00
 *""FILE COMMENT END""************************************************/

#include "sfr29.h"
#include "define2.h"
#include "declare2.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :2.0
  *Description         :Initialize CAN
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *--------------------------------------------------------------------
  *Declaration         :void can_initial(void)
  *--------------------------------------------------------------------
  *Function            :Initialize CAN
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :unsigned char trm_id          ;transmit ID
  *                    :unsigned char rcv_id          ;receive ID
  *--------------------------------------------------------------------
  *Call functions      :config_can0()
  *                    :set_rec_std_dataframe_can0
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
```

```
   *""FUNC COMMENT END""************************************************/
void can_initial(void)
{
      config_can0();                                  //Configurate can
      c0recic = CAN0_LVL;                               //Interrupt lvl
      c01erric = CAN0_ERROR_LVL;                          //Interrupt lvl

      set_rec_std_dataframe_can0(14,CAN_RCV_ID);   //Set can slot14
      set_rec_std_dataframe_can0(15,CAN_RCV_ID);   //Set can slot15

      trm_id = CAN_TRM_ID;                             //Set transmit ID
      rcv_id = CAN_RCV_ID;                            //Set receive ID
}


 /*""FUNC COMMENT""*****************************************************
   *ID                  :2.1
   *Description         :Set can slot receive
   *--------------------------------------------------------------------
   *Include             :"sfr29.h"
   *                    :"declare2.h"
   *--------------------------------------------------------------------
   *Declaration         :void set_rec_std_dataframe_can0
   *                          (unsigned short in_slot,unsigned short in_sid)
   *--------------------------------------------------------------------
   *Function            :Set can slot receive
   *--------------------------------------------------------------------
   *Arguments             :unsigned short in_slot
   *                    :unsigned short in_sid
   *--------------------------------------------------------------------
   *Returns             :Nothing
   *--------------------------------------------------------------------
   *Input               :Nothing
   *Ountput             :Nothing
   *--------------------------------------------------------------------
   *Call functions     :Nothing
   *--------------------------------------------------------------------
   *Note                :
   *--------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""************************************************/
void set_rec_std_dataframe_can0(
      unsigned short in_slot,
      unsigned short in_sid)
{
      c0icr |= (0x0001<<in_slot);
      c0idr &= ~(0x0001<<in_slot);

      c0slot[in_slot].ba.sidh = (in_sid>>6) & 0x1f;     //SID10-6
      c0slot[in_slot].ba.sidl = in_sid & 0x3f;          //SID5-0

      while(c0mctl[in_slot].byte != 0x00){
            c0mctl[in_slot].byte = 0x00;
```

```
    }
    c0mctl[in_slot].byte = 0x40;
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :2.0.2
  *Description         :Set can bitrate
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"declare2.h"
  *--------------------------------------------------------------------
  *Declaration         :void set_bitrate_can0(void)
  *--------------------------------------------------------------------
  *Function            :Set can bitrate
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :Nothing
  *--------------------------------------------------------------------
  *Call functions      :Nothing
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
void set_bitrate_can0(void)
{
    prc0 = 1;
    cclkr = 0x01;          // CAN0 clock = f1/2
    prc0 = 0;

    /* 500kbps 10Tq */
    brp_c0conr = 0;        // fcan(10MHz)
    sam_c0conr = 0;        //
    pts_c0conr = 2;        // PTS = 3Tq
    pbs1_c0conr = 2;       // PBS1 = 3 Tq
    pbs2_c0conr = 2;       // PBS2 = 3 Tq
    sjw_c0conr = 0;        // SJW = 1 Tq
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :2.0.1
  *Description         :CAN configuration
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *--------------------------------------------------------------------
```

```
    *Declaration         :void set_rec_std_dataframe_can0
    *                         (unsigned short in_slot,unsigned short in_sid)
    *-------------------------------------------------------------------
    *Function            :Set can slot receive
    *-------------------------------------------------------------------
    *Arguments            :Nothing
    *-------------------------------------------------------------------
    *Returns             :Nothing
    *-------------------------------------------------------------------
    *Input               :Nothing
    *Ountput             :Nothing
    *-------------------------------------------------------------------
    *Call functions      :set_bitrate_can0()
    *                    :set_mask_can0()
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""***********************************************/
  void config_can0(void)
  {
      reset_c0ctlr = 1;                 /* configration */
      sleep_c0ctlr = 0;
      while(! state_reset_c0str){}

      porten_c0ctlr = 1;                /* CAN port enable */
      loopback_c0ctlr = 0;              /* normal */
      msgorder_c0ctlr = 1;               /* byte access */
      basiccan_c0ctlr = 1;              /* 0:normal mode 1:BasicCAN mode */
      buserren_c0ctlr = 0;              /* nus error interrupt disable */
      tsprescale_c0ctlr = 0;

      set_bitrate_can0();               /* set bit rate */
      set_mask_can0();                    /* mask */

      reset_c0ctlr = 0;
      while(state_reset_c0str){}
  }


   /*""FUNC COMMENT""*****************************************************
    *ID                  :2.0.3
    *Description         :Set can mask
    *-------------------------------------------------------------------
    *Include             :"sfr29.h"
    *                    :"define2.h"
    *                    :"declare2.h"
    *-------------------------------------------------------------------
    *Declaration         :void set_mask_can0(void)
    *-------------------------------------------------------------------
    *Function            :Set can mask
    *-------------------------------------------------------------------
    *Arguments            :Nothing
```

```
  *-------------------------------------------------------------------
  *Returns              :Nothing
  *-------------------------------------------------------------------
  *Input                :Nothing
  *Ountput              :Nothing
  *-------------------------------------------------------------------
  *Call functions     :Nothing
  *-------------------------------------------------------------------
  *Note                 :
  *-------------------------------------------------------------------
  *History              :
  *                     :
  *""FUNC COMMENT END""***************************************************/
void set_mask_can0(void)
{
    c0lmar.ba.sidh = (MASK_14>>6) & 0x1f;              //SID10-6
    c0lmar.ba.sidl = MASK_14 & 0x3f;                   //SID5-0

    c0lmbr.ba.sidh = (MASK_15>>6) & 0x1f;              //SID10-6
    c0lmbr.ba.sidl = MASK_15 & 0x3f;                   //SID5-0
}


 /*""FUNC COMMENT""*******************************************************
  *ID                   :2.2
  *Description          :Get can data
  *-------------------------------------------------------------------
  *Include              :"sfr29.h"
  *                     :"define2.h"
  *                     :"declare2.h"
  *-------------------------------------------------------------------
  *Declaration          :void get_message_can0
  *                       (unsigned short in_slot,can_std_data_def *in_rec_data)
  *-------------------------------------------------------------------
  *Function             :Get can data
  *-------------------------------------------------------------------
  *Arguments             :unsigned short in_slot            ;slot number
  *                     :can_std_data_def *in_rec_data   ;receive data pointer
  *-------------------------------------------------------------------
  *Returns              :Nothing
  *-------------------------------------------------------------------
  *Input                :Nothing
  *Ountput              :Nothing
  *-------------------------------------------------------------------
  *Call functions     :Nothing
  *-------------------------------------------------------------------
  *Note                 :
  *-------------------------------------------------------------------
  *History              :
  *                     :
  *""FUNC COMMENT END""***************************************************/
void get_message_can0(
    unsigned short in_slot,
    can_std_data_def *in_rec_data)
```

```
{
    unsigned char lp_dlc;

    while(1){
        /* check receive complete */
        while(c0mctl[in_slot].receive.invaldata){}

        c0mctl[in_slot].receive.newdata = 0;

        /* overrun check */
        if(c0mctl[in_slot].receive.msglost){
            c0mctl[in_slot].receive.msglost = 0;
            error_fnc(CAN_RCV_BUFF_ERROR);
            break;
        }

        /* read message */
        in_rec_data->id = ((unsigned short)c0slot[in_slot].ba.sidh<<6)
                                    + c0slot[in_slot].ba.sidl;
        in_rec_data->dlc = c0slot[in_slot].ba.dlc;
        for(lp_dlc=0; lp_dlc<c0slot[in_slot].ba.dlc; ++lp_dlc){
            in_rec_data->data.data[lp_dlc] = c0slot[in_slot].ba.data[lp_dlc];
        }

        /* check new message */
        if(! c0mctl[in_slot].receive.newdata){
            break;
        }
    }
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :2.3
  *Description         :Set can slot transmit
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *--------------------------------------------------------------------
  *Declaration         :void set_trm_std_dataframe_can0
  *                         (unsigned short in_slot,can_std_data_def *in_trm_data)
  *--------------------------------------------------------------------
  *Function            :Set can slot transmit
  *--------------------------------------------------------------------
  *Arguments           :unsigned short in_slot           ;slot number
  *                    :can_std_data_def *in_trm_data    ;transmit data pointer
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :Nothing
  *--------------------------------------------------------------------
  *Call functions      :Nothing
```

```
    *-------------------------------------------------------------------
    *Note                :
    *-------------------------------------------------------------------
    *History             :
    *                    :
    *""FUNC COMMENT END""************************************************/
  void set_trm_std_dataframe_can0(
        unsigned short in_slot,
        can_std_data_def *in_trm_data)
  {
        unsigned char lp_dlc;

        while(c0mctl[in_slot].transmit.trmactive){}

        while(c0mctl[in_slot].byte != 0x00){
              c0mctl[in_slot].byte = 0x00;
        }

    /* transmission */
        c0icr &= ~(0x0001 << in_slot);
        c0idr &= ~(0x0001 << in_slot);

  //------------- set dataframe     ----------------
        c0slot[in_slot].ba.sidh = ((in_trm_data->id)>>6) & 0x1f;
                                                          //SID10-6
        c0slot[in_slot].ba.sidl = (in_trm_data->id) & 0x3f;
                                                          //SID5-0
        c0slot[in_slot].ba.dlc = in_trm_data->dlc;        //DLC

        for(lp_dlc = 0;lp_dlc<(in_trm_data->dlc) && (lp_dlc<8) ; lp_dlc++){
              c0slot[in_slot].ba.data[lp_dlc] = in_trm_data->data.data[lp_dlc];
                                                          //Data
        }
  //-----------------------------------------------
        c0mctl[in_slot].byte = 0x80;
  }


    /*""FUNC COMMENT""*****************************************************
    *ID                  :2.4
    *Description         :Abort can transmit
    *-------------------------------------------------------------------
    *Include             :"sfr29.h"
    *                    :"define2.h"
    *                    :"declare2.h"
    *-------------------------------------------------------------------
    *Declaration         :void abort_trm_can0(unsigned short in_slot)
    *-------------------------------------------------------------------
    *Function            :Abort can transmit
    *-------------------------------------------------------------------
    *Arguments           :unsigned short in_slot           ;slot number
    *-------------------------------------------------------------------
    *Returns             :Nothing
    *-------------------------------------------------------------------
```

```
    *Input                :Nothing
    *Ountput              :Nothing
    *---------------------------------------------------------------
    *Call functions     :Nothing
    *---------------------------------------------------------------
    *Note                 :
    *---------------------------------------------------------------
    *History              :
    *                     :
    *""FUNC COMMENT END""***********************************************/
void abort_trm_can0(unsigned short in_slot)
{
      if(c0mctl[in_slot].transmit.trmreq
            || c0mctl[in_slot].receive.recreq){

            c0mctl[in_slot].byte = 0;
            while(c0mctl[in_slot].transmit.trmactive){}
      }
}
```

```
/*""FILE COMMENT""*****************************************************
 *System Name : CAN DEMO PROGRAM2
 *File Name    : sw_led2.c
 *Version      : 1.00
 *Contens      :
 *Customer     : RSO
 *Model        :
 *Order        :
 *CPU          :
 *Compiler     : NC30 Version 5.30 Release 02
 *OS           :
 *Programmer   :
 *Note         :
 *             :
 ***********************************************************************
 * Copyright(C)2005, Renesas Technology Corp.
 * Copyright(C)2005, Renesas Solutions Corp.
 * All rights reserved.
 *
 ***********************************************************************
 *History          2005.07.1   Ver 1.00
 *""FILE COMMENT END""*************************************************/

 #include "sfr29.h"
 #include "define2.h"
 #include "declare2.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :3.0
  *Description         :Initialize Switch and LED
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *--------------------------------------------------------------------
  *Declaration         :void sw_led_initial(void)
  *--------------------------------------------------------------------
  *Function            :Initialize Switch and LED
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :unsigned char led_data[]        ;LED display data
  *                    :unsigned char trm_led_data[]    ;Transmit data
  *--------------------------------------------------------------------
  *Call functions      :Nothing
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
```

```
   *""FUNC COMMENT END""************************************************/
void sw_led_initial(void)
{
      /* LED initial */
      p0    = 0x03;
      p2    = 0xff;
      pd0 = 0x03;
      pd2 = 0xff;

      led_data[0] = LED_TBL[0x00];      //initial LED dsp
      led_data[1] = LED_TBL[0x00];      //

      /* SW initial */
      pd8_2 = 0;                        //SW2(INT0)
      pd8_3 = 0;                        //SW3(INT1)
      pd8_4 = 0;                        //SW4(INT2)

      /* variable initial */
      trm_led_data[0] = DATA_CLWS;           //set transmit data
      trm_led_data[1] = DATA_ANTCLWS;        //
      trm_led_data[2] = DATA_HALF_CLWS;      //
}

 /*""FUNC COMMENT""*****************************************************
   *ID                  :3.1
   *Description         :LED dynamic lightning
   *--------------------------------------------------------------------
   *Include             :"sfr29.h"
   *                    :"define2.h"
   *                    :"declare2.h"
   *--------------------------------------------------------------------
   *Declaration         :void led_fnc(void)
   *--------------------------------------------------------------------
   *Function            :LED dynamic lightning
   *--------------------------------------------------------------------
   *Arguments           :Nothing
   *--------------------------------------------------------------------
   *Returns             :Nothing
   *--------------------------------------------------------------------
   *Input               :unsigned char led_data[]    ;LED display data
   *                    :unsigned char led_count     ;LED counter
   *Ountput             :unsigned char led_count     ;LED counter
   *--------------------------------------------------------------------
   *Call functions      :Nothing
   *--------------------------------------------------------------------
   *Note                :
   *--------------------------------------------------------------------
   *History             :
   *                    :
   *""FUNC COMMENT END""************************************************/
void led_fnc(void)
{
      p0 = 0x03;                        //LED OFF
```

```
    led_count++;
    if(led_count > 1){
        led_count = 0;
    }
    //set led data
    p2 = led_data[led_count];

    p0 = 0x01 << led_count;      //LED ON
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :3.2
  *Description         :Switch function
  *--------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *--------------------------------------------------------------------
  *Declaration         :void sw_fnc(void)
  *--------------------------------------------------------------------
  *Function            :Switch function
  *--------------------------------------------------------------------
  *Arguments           :Nothing
  *--------------------------------------------------------------------
  *Returns             :Nothing
  *--------------------------------------------------------------------
  *Input               :Nothing
  *Ountput             :Nothing
  *--------------------------------------------------------------------
  *Call functions     :sw_down()                ;Switch down function
  *                    :fix_sw_fnc()             ;Fix switch function
  *                    :sw_decode()              ;Switch decode function
  *--------------------------------------------------------------------
  *Note                :
  *--------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
void sw_fnc(void)
{
    unsigned char in_data,i;

    in_data = p8 & PT_SW_MASK;
    if(fix_sw_fnc(in_data)){
        i = sw_decode();
        if(i){
            sw_down(i-1);
        }
    }
}


 /*""FUNC COMMENT""*****************************************************
```

```
    *ID                 :3.2.1
    *Description        :Switch decode function
    *-------------------------------------------------------------------
    *Include            :"define2.h"
    *                   :"declare2.h"
    *-------------------------------------------------------------------
    *Declaration        :unsigned char sw_decode(void)
    *-------------------------------------------------------------------
    *Function           :Switch decode function
    *-------------------------------------------------------------------
    *Arguments          :Nothing
    *-------------------------------------------------------------------
    *Returns            :unsigned char in_rtn              ;Decode number
    *-------------------------------------------------------------------
    *Input              :unsigned char fix_sw_data         ;Fixed switch data
    *                   :unsigned char last_fix_sw_data ;Last fixed switch data
    *Ountput            :unsigned char last_fix_sw_data ;Last fixed switch data
    *-------------------------------------------------------------------
    *Call functions     :error_fnc()                       ;Error function
    *-------------------------------------------------------------------
    *Note               :
    *-------------------------------------------------------------------
    *History            :
    *                   :
    *""FUNC COMMENT END""***********************************************/
unsigned char sw_decode(void)
{
    unsigned char in_rtn = 0;

    if(last_fix_sw_data != PT_NO_SW){
        in_rtn = 0;
    }
    else{
        switch(fix_sw_data){
            case PT_NO_SW:      //no SW
                in_rtn = 0;
                break;
            case PT_SW2:        //SW2
                in_rtn = 1;
                break;
            case PT_SW3:        //SW3
                in_rtn = 2;
                break;
            case PT_SW4:        //SW4
                in_rtn = 3;
                break;
            default:
                in_rtn = 0;
                break;
        }
    }

    last_fix_sw_data = fix_sw_data;
    return in_rtn;
```

```
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :3.2.2
  *Description         :Fix switch function
  *-------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *-------------------------------------------------------------------
  *Declaration         :unsigned char fix_sw_fnc(unsigned char)
  *-------------------------------------------------------------------
  *Function            :Fix switch function
  *-------------------------------------------------------------------
  *Arguments           :unsigned char in_data          ;Switch data
  *-------------------------------------------------------------------
  *Returns             :uneigned char in_rtn           ;Fixed switch data
  *-------------------------------------------------------------------
  *Input               :unsigned char fix_sw_data      ;Fixed switch data
  *                    :unsigned char last_sw_data     ;Last switch data
  *                    :unsigned char sw_count         ;Switch counter
  *Ountput             :unsigned char trm_count_sw     ;Transmit counter
  *-------------------------------------------------------------------
  *Call functions      :error_fnc()                    ;Error function
  *-------------------------------------------------------------------
  *Note                :
  *-------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""************************************************/
unsigned char fix_sw_fnc(unsigned char in_data)
{
    unsigned char in_rtn = 0;

    if((fix_sw_data != in_data) && (last_sw_data == in_data)){
        sw_count++;
    }

    if(sw_count > CHAT_COUNT){
        fix_sw_data = in_data;
        sw_count = 0;
        in_rtn = 1;
    }

    last_sw_data = in_data;

    return in_rtn;
}


 /*""FUNC COMMENT""*****************************************************
  *ID                  :3.2.3
  *Description         :Switch down function
```

```
   *-----------------------------------------------------------------------
   *Include              :"sfr29.h"
   *                     :"define2.h"
   *                     :"declare2.h"
   *-----------------------------------------------------------------------
   *Declaration          :void sw_down(unsigned char in_trm_num)
   *-----------------------------------------------------------------------
   *Function             :Switch2 down function
   *-----------------------------------------------------------------------
   *Arguments            :Nothing
   *-----------------------------------------------------------------------
   *Returns              :Nothing
   *-----------------------------------------------------------------------
   *Input                :unsigned char trm_id              ;Transmit ID
   *                     :unsigned char trm_led_data[]     ;Transmit data
   *Ountput              :Nothing
   *-----------------------------------------------------------------------
   *Call functions       :error_fnc()                      ;Error function
   *-----------------------------------------------------------------------
   *Note                 :
   *-----------------------------------------------------------------------
   *History              :
   *                     :
   *""FUNC COMMENT END""***************************************************/
void sw_down(unsigned char in_trm_num)
{
    can_std_data_def in_trm_data;
/* ID */
    in_trm_data.id = trm_id;
/* DLC */
    in_trm_data.dlc = TRM_DLC;
/* data */
    if(in_trm_num >= sizeof(trm_led_data)){
        error_fnc(CAN_TRM_ERROR);
    }

    in_trm_data.data.data[0] = trm_led_data[in_trm_num];

/* transmission */
    set_trm_std_dataframe_can0(TRM_SLOT,&in_trm_data);
}
```

```
/*""FILE COMMENT""*****************************************************
 *System Name : CAN DEMO PROGRAM
 *File Name    : int2.c
 *Version      : 1.00
 *Contens      :
 *Customer     : RSO
 *Model        :
 *Order        :
 *CPU          :
 *Compiler     : NC30 Version 5.30 Release 02
 *OS           :
 *Programmer   :
 *Note         :
 ********************************************************************
 * Copyright(C)2005, Renesas Technology Corp.
 * Copyright(C)2005, Renesas Solutions Corp.
 * All rights reserved.
 *
 ********************************************************************
 *History          2005.07.1   Ver 1.00
 *""FILE COMMENT END""*************************************************/

#include "sfr29.h"
#include "define2.h"
#include "declare2.h"

 /*""FUNC COMMENT""*****************************************************
  *ID                  :4.0
  *Description         :CAN0 receive interrupt
  *-------------------------------------------------------------------
  *Include             :"sfr29.h"
  *                    :"define2.h"
  *                    :"declare2.h"
  *-------------------------------------------------------------------
  *Declaration         :void CAN0_REC(void)
  *-------------------------------------------------------------------
  *Function            :Read CAN0 receive data
  *-------------------------------------------------------------------
  *Arguments           :Nothing
  *-------------------------------------------------------------------
  *Returns             :Nothing
  *-------------------------------------------------------------------
  *Input               :unsigned char rec_data[]        ;CAN receive data
  *Ountput             :unsigned char led_data[]        ;LED display data
  *-------------------------------------------------------------------
  *Call functions     :get_message_can0()              ;Get CAN data
  *-------------------------------------------------------------------
  *Note                :
  *-------------------------------------------------------------------
  *History             :
  *                    :
  *""FUNC COMMENT END""*************************************************/
void CAN0_REC(void)
```

```
{
    unsigned char in_rcv_data = 0;

    /* slot14 */
    if(c0mctl[14].receive.newdata == 1){
        get_message_can0(14, &rec_data[0]);
        in_rcv_data = rec_data[0].data.data[0];
    }

    /* slot15 */
    if(c0mctl[15].receive.newdata == 1){
        get_message_can0(15, &rec_data[1]);
        in_rcv_data = rec_data[1].data.data[0];
    }

    led_data[0] = LED_TBL[0x0f & in_rcv_data];
    led_data[1] = LED_TBL[in_rcv_data >> 4];
}
```

# 7. Reference

## Renesas Technology Corporation Home Page

http://www.renesas.com/en/m16c

## E-mail Support

E-mail: csc@renesas.com

## Hardware Manual

M16C/29 Group Hardware Manual Rev.1.0
(Use the latest version on the home page: http://www.renesas.com)

## TECHNICAL UPDATE/TECHNICAL NEWS

(Use the latest information on the home page: http://www.renesas.com)

REVISION HISTORY

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | 2005.01.25 | - | First edition issued |
| | | | |

## 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.