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.

SH7262/SH7264 Group

Controller Area Network, Configuration to Transmit Remote Frames

Summary

This application note describes the configuration example of the SH7264 microcomputers (MCUs) to transmit remote frames using the Controller Area Network.

Target Device

SH7264 MCU (In this document, SH7262/SH7264 are described as "SH7264".)

Contents

1. Introduction

1.1 Specifications

- Uses the Controller Area Network channel 0
- Transmission speed: 1 Mbps
- Receive mailbox: Mailbox 1
- Transmit mailbox: Mailbox 0
- Transmits a remote frame with following specifications
  Identifier: 0; standard remote frame; DLC: 2
- Receives a data frame with following specifications
  Identifier: 0; standard data frame; DLC: 2; Data: H'C1C2

1.2 Modules Used

- Controller Area Network (CAN) module

1.3 Applicable Conditions

MCU	SH7262/SH7264
Operating Frequencies	Internal clock: 144 MHz Bus clock: 72 MHz Peripheral clock: 36 MHz
Integrated Development Environment	Renesas Technology Corp. High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Technology SuperH RISC engine Family C/C++ Compiler Package Ver.9.03 Release 00
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu -fpu=single -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 –nologo)

1.4 Related Application Notes

For more information, refer to the following application notes:

- SH7262/SH7264 Group Controller Area Network, Configuration to Receive Data Frames
- SH7262/SH7264 Group Controller Area Network, Configuration to Transmit Data Frames
- SH7262/SH7264 Group Controller Area Network, Configuration to Receive Remote Frames

2. Applications

This application note uses the CAN module to transmit a standard remote frame with identifier 0, DLC: 2. Then, it receives a standard data frame with identifier 0.

2.1 CAN Overview

The SH7264 includes two channels of a CAN module which is compliant with the CAN protocol, version 2.0B active, and ISO 11898.

The CAN module has 31 programmable mailboxes for transmission/reception, one mailbox for reception, and a programmable receive filtering mask to provide flexible communication procedure. Figure 1 shows the CAN block diagram. For more details, refer to the Controller Area Network chapter in the SH7262 Group, SH7264 Group Hardware Manual.

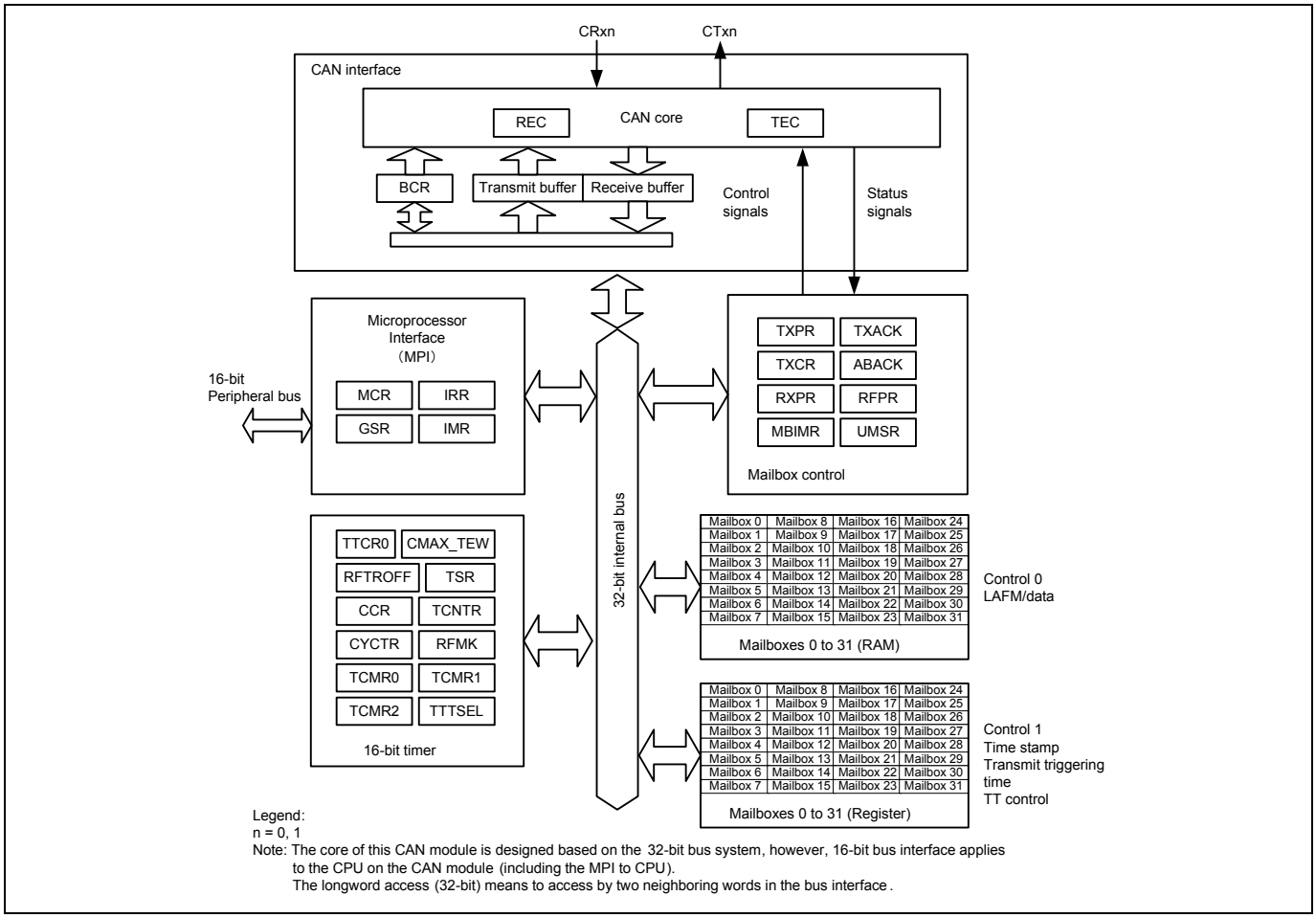


Figure 1 CAN Block Diagram (For One Channel)

2.2 Configuration Procedure

This section describes how to configure the SH7264 MCU to receive remote frames using the CAN module.

Configure the CAN module in reset mode (configuration mode). After configuration is complete, clear the reset mode to join the CAN bus activity. The sample program configures mailbox 1 for both transmitting data frames and receiving remote frames. Figure 2 and Figure 3 show the flow charts for configuring the CAN module. For details on register settings, refer to the Controller Area Network chapter in the SH7262 Group, SH7264 Group Hardware Manual.

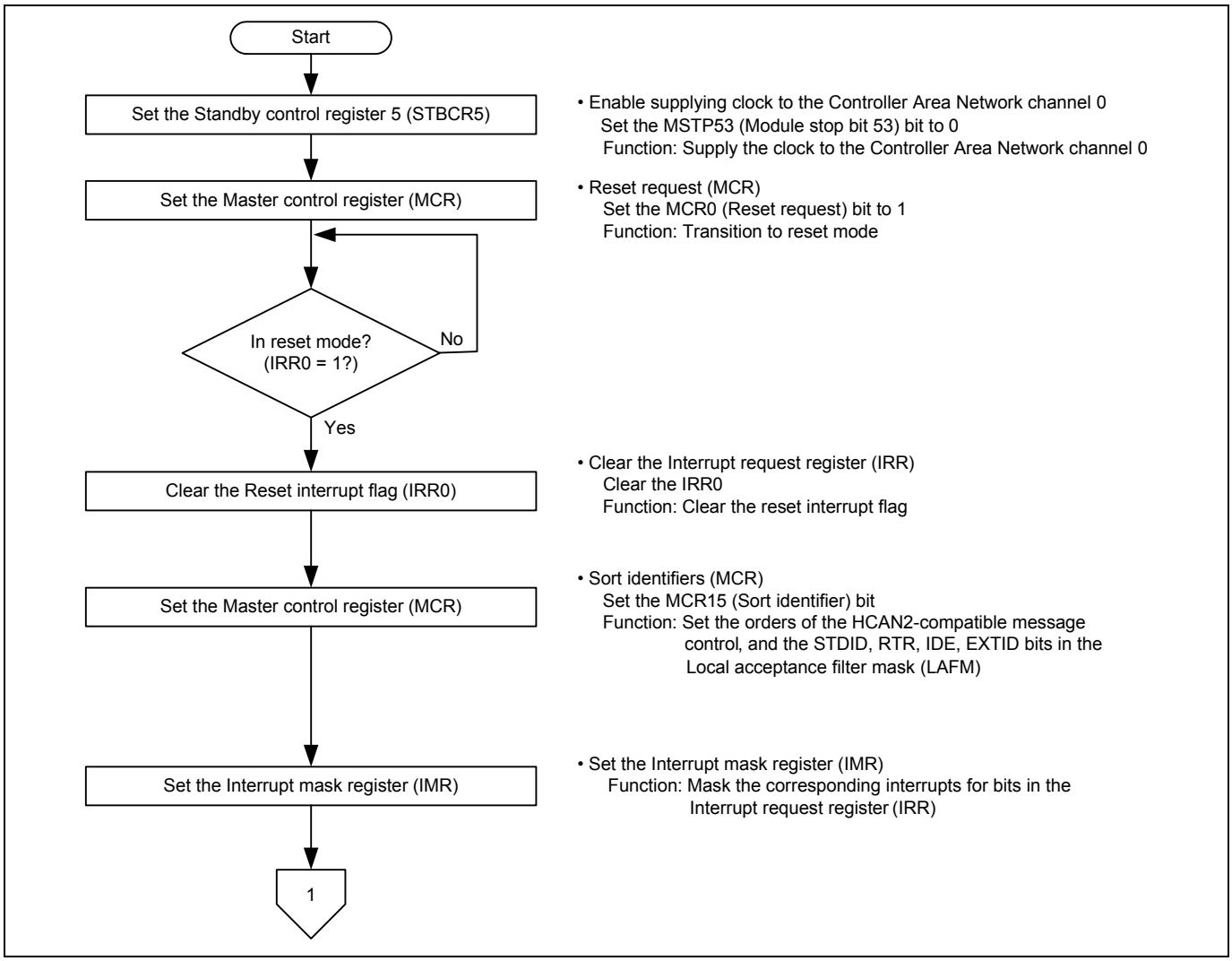
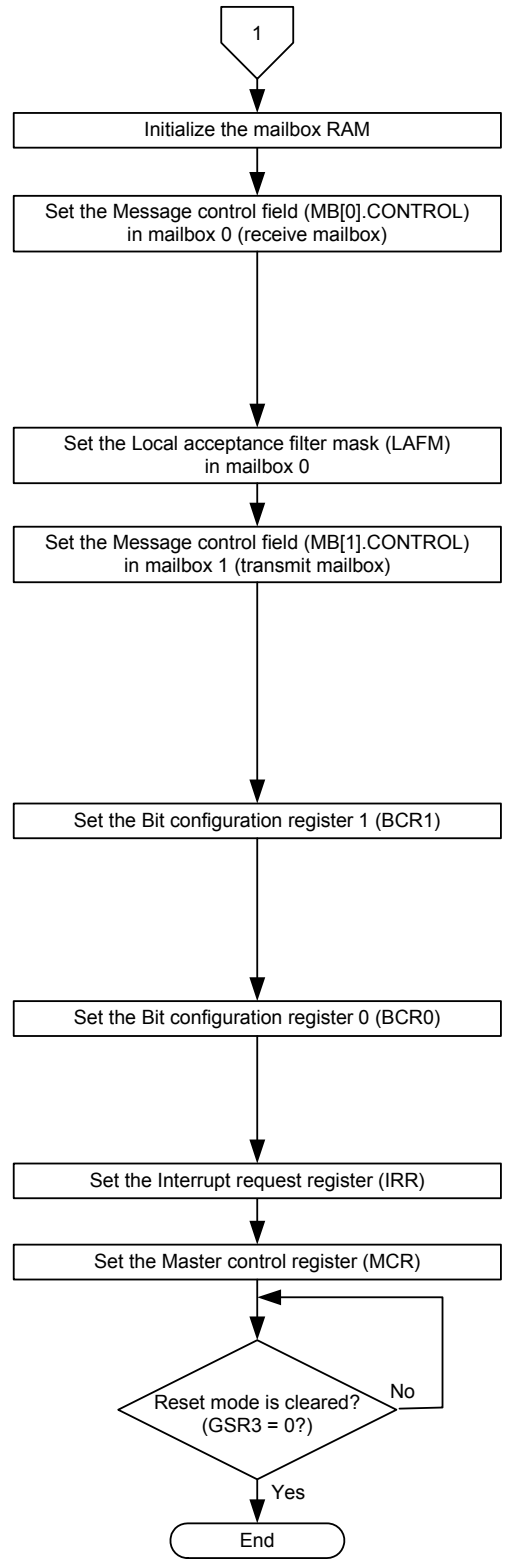


Figure 2 Flow Chart for Configuring the CAN Module (1/2)



• Clear the RAM area in mailbox

• Set the Message control register (MB[0].CONTROL)
Specify "B'010" in MBC [2:0] bits (Mailbox configuration)
Function: Receiving a remote frame or a data frame is enabled
Specify "0" in the IDE bit (Identifier extension)
Function: Specify the standard format
Specify "0" in the RTR bit (Remote transmission request)
Function: Specify a data frame
Specify "0" in STDID [10:0] bits (Standard Identifier)
Function: Specify 0 as the ID

• Set the Local acceptance filter mask (LAFM)
Specify "H'0000 0000" in the LAFM
Function: Mask is not specified

• Set the Message control register (MB[1].CONTROL)
Specify "B'000" in MBC [2:0] bits (Mailbox configuration)
Function: Transmitting a remote frame or a data frame is enabled
Specify "B'010" in the DLC bit (Data length code)
Function: Specify the data length as 2 bytes
Specify "0" in the IDE bit (Identifier extension)
Function: Specify the standard format
Specify "0" in the RTR bit (Remote transmission request)
Function: Specify a data frame
Specify "0" in the STDID [10:0] bits
Function: Specify 0 as the ID

• Set the Bit configuration register 1 (BCR1)
Specify "B'1001" (10 time quanta) in TSG1 [3:0] bits (Time segment 1)
Specify "B'110" (7 time quanta) in TSG2 [2:0] bits (Time segment 2)
Specify "B'00" (1 time quantum) in SJW [1:0] bits (Resynchronization jump width)
Function: Avoid moving the sample point substantially
Set the BSP (Bit sample point)
Function: Specify to sample at one point or to sample at three points and decide by majority of them

• Set the Bit configuration register 0 (BCR0)
Specify "B'0000 0000" in BRP [7:0] bits (Baud rate prescaler)
Function: Specify 1 time quantum as 2 × peripheral bus clock
Specify the transfer speed as 1 Mbps by TSG1, TSG2 and the baud rate prescaler
(Refer to 2.3 Bit Confuguration and Transmission Speed)

• Clear the Interrupt request register (IRR)
Clear IRR0 to IRR15 bits
Function: Clear interrupt flags

• Set the Master control register (MCR)
Clear the MCR0 bit (Reset request)
Function: Clear the reset mode transition request
Clear the MCR1 bit (Halt request)
Function: Clear the halt mode transition request

Figure 3 Flow Chart for Configuring the CAN Module (2/2)

2.3 Bit Configuration and Transmission Speed

One bit in the CAN module consists of the following four segments:

1. Synchronization segment (SS)
2. Propagation time segment (PRSEG)
3. Phase buffer segment 1 (PHSEG1)
4. Phase buffer segment 2 (PHSEG2)

Each segment is composed of the reference time Tq (time quanta). Figure 4 shows the bit configuration example when SS = 1 Tq, PRSEG = 8 Tq, PHSEG1 = 8 Tq, and PHSEG2 = 8 Tq.

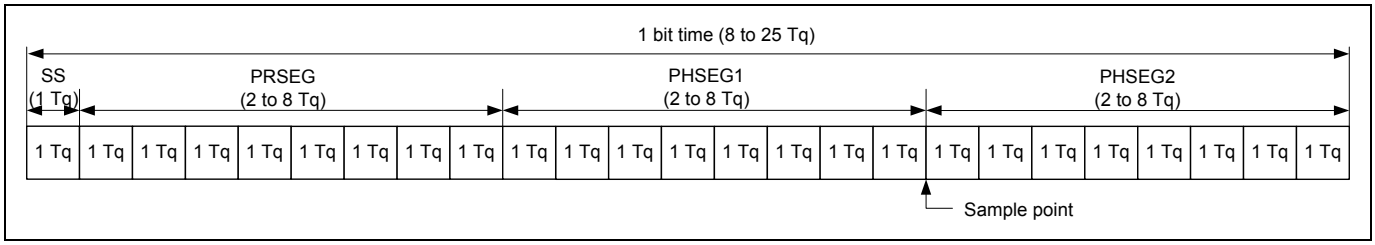


Figure 4 Bit Configuration

The CAN module sets the number of Tqs of PRSEG + PHSEG1 to bits TSG1 [3:0] in the BCR1, and the number of Tqs of PSEG2 to bits TSG2 [2:0] in this register (Value + 1 is the number of Tqs). Also, the number of peripheral bus clocks for 1 Tq is set to bits BRP [7:0] in the BCR0.

In the following description, bits BRP [7:0], TSEG1 [3:0], and TSEG2 [2:0] are register values, and bits BRP, TSEG1, TSEG2, and SJW are the corresponding values for the register values. For the corresponding values for register values, refer to the Controller Area Network chapter in the SH7262 Group, SH7264 Group Hardware Manual.

The CAN module defines $1\,\mathrm{Tq} = \frac{2\times(\mathrm{BRP}[7:0]+1)}{\text{Peripheral bus clock}}$ By this formula, the transmission speed is calculated as follows:

$$\text{Transmission speed} = \frac{\text{Peripheral bus clock}}{2\times(\mathrm{BRP}[7:0]+1)\times\text{the number of Tqs/bit}}$$

$$= \frac{\text{Peripheral bus clock}}{2\times(\mathrm{BRP}[7:0]+1)\times\{(\mathrm{TSEG}[3:0]+1)+(\mathrm{TSEG2}[2:0]+1)+1\}}$$

Following is the restriction on setting the bit configuration register.

TSEG1 (Min.) > TSEG2 ≥ SJW (Max.) (SJW = 1 to 4)

SJW is the resynchronization jump width. It is a segment that lengthens phase buffer segment 1 or shortens phase buffer segment 2 to correct the phase difference.

8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta
TSEG2 ≥ 2

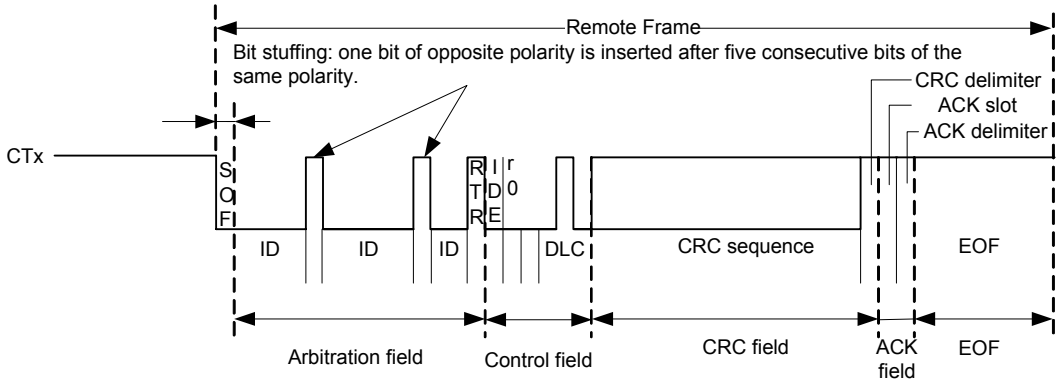
As this sample program specifies the peripheral bus clock as 36 MHz, BRP [7:0] = 0, TSEG1 [3:0] = 9, and TSEG2 [2:0] = 6, the transmission speed is calculated as follows:

$$\text{Transmission speed} = \frac{36\mathrm{M}}{2\times(0+1)\times\{(9+1)+(6+1)+1\}} = 1\mathrm{M}\ldots\ldots 1\ \text{Mbps}$$

2.4 Sample Program Operation

This sample program transmits a standard remote frame with identifier 0, DLC: 2 from mailbox 1 at 1 Mbps. Then it receives a standard data frame with identifier 0 in mailbox 0. Figure 5 shows the transmission waveform of a remote frame.

Note: The sample program transmits and receives data frames, however, this application note describes only the transmission.



Legend:
SOF (Start of frame): Indicates the beginning of the data frame and remote frame
ID (Identifier): Indicates the data priority
RTR (Remote transmission request): Indicates differences between the data frame and remote frame
IDE (Identifier extension): Indicates that it is the standard format or extended format
r0: This is a reserved bit.
DLC (Data length code) Indicates the number of bytes in the data field (0 to 8 bytes can be specified)
CRC sequence: This is the value to check the frame.
CRC delimiter: Delimits the CRC field and ACK field
ACK slot: When the transmitted message includes no error, this is overwritten by a dominant.
ACK delimiter: Delimits the ACK field and EOF
EOF (End of frame): Indicates the end of the frame

Figure 5 CAN Transmission Waveform

2.5 Sample Program Procedure

The following table lists setting example of the CAN. Figure 6 shows the configuration flow chart of this sample program.

Note: The sample program transmits and receives remote frames, however, this application note describes only the transmission.

Table 1 Controller Area Network Settings

Register Name	Address	Setting	Description
Standby control register (STBCR5)	H'FFFE 0410	H'F7	MSTP53 = "0": Controller Area Network channel 0 is operating
Master control register (MCR)	H'FFFF 5000	H'0001	MCR0 = "1": Reset mode transition request
		H'8001	MCR15 = "1": The order of the RCAN message and of the HCAN2 message are different
		H'8000	MCR0 = "0": Reset mode is cleared
Interrupt mask register(IMR)	H'FFFF 500A	H'FFFF	All interrupts in the Controller Area Network are disabled
Bit configuration register 1 (BCR1)	H'FFFF 5004	H'9600	TSEG1 [3:0] = "B'1001": PRSEG + PHSEG1 = 10 Tq TSEG2 [2:0] = "B'110": PHSEG2 = 7 Tq SJW="0": SJW = 1 Tq BSP = "0": Bit sampling at one point
Bit configuration register 0 (BCR0)	H'FFFF D006	H'0000	BRP [7:0] = "0": 1 Tq = 2 × Pφ
Message control field in mailbox 0 (MB[0].CONTROL1)	H'FFFF 5110	H'0200	MBC [2:0] = “B’010”: Receiving the data frame or remote frame is enabled
Message control field in mailbox 0 (MB[0].CONTROL0)	H'FFFE 5100	H'0000 0000	IDE = "0": Standard format RTR = "0": Data frame STDID [10:0] = "0": Standard identifier is 0
Message control field in mailbox 1 (MB[1].CONTROL1)	H'FFFF 5130	H'0002	MBC [2:0] = "B'000": Transmitting the data frame or remote frame is enabled DLC [3:0] = "B'0010": Data length is 2 bytes
Message control field in mailbox 1 (MB[1].CONTROL0)	H'FFFF 5120	H'4000 0000	IDE = "0": Standard format RTR = "1": Remote frame STDID [10:0] = "0": Standard identifier is 0
Local acceptance filter mask in mailbox 0 (MB[0].LAFM)	H'FFFF 5104	H'0000 0000	Clear: Mask is not specified
Transmit pending register 1 (TXPR1)	H'FFFF 5020	H'0000 0002	TXPR [31:0] = H'0000 0002: A transmission request occurred in mailbox 1
Transmit acknowledge register 0 (TXACK0)	H'FFFF 5032	H'0002	Clear the transmit acknowledge flag
Data frame receive pending register (RxPR0)	H'FFFF 5042	H'0001	Clear the data frame receive pending flag

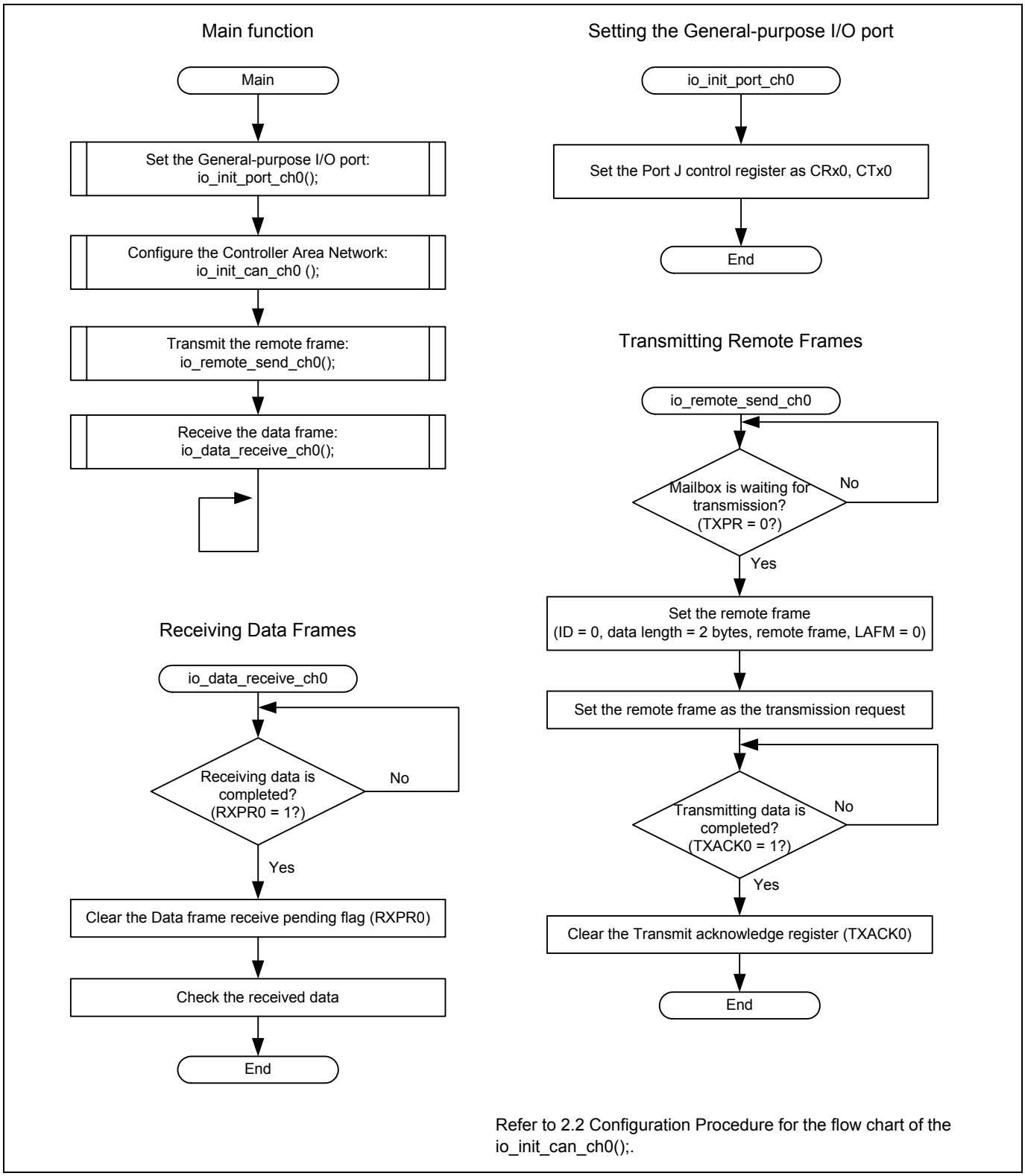


Figure 6 Sample Program Flow Chart

3. Sample Program Listing

3.1 Supplement to the Sample Program

As the capacity of the SH7264 large-capacity internal RAM varies as 1 MB or 640 KB, depending on the MCU type, the section alignment and register setting must be partly altered. To support both MCU types, this application note provides two types of sample programs (workspaces) for 1-MB RAM and 640-KB RAM.

As the MCU with 640-KB RAM must be write-enabled before writing data in the data-retention RAM, the System control register 5 (SYSCR5) is set to write-enable the RAM in the sample program for 640-KB RAM.

Review your product and use the appropriate workspace.

3.2 Sample Program Listing "main.c" (1/3)

```
1       /***********************************************************************
2       *   DISCLAIMER
3       *
4       *   This software is supplied by Renesas Technology Corp. and is only
5       *   intended for use with Renesas products. No other uses are authorized.
6       *
7       *   This software is owned by Renesas Technology Corp. and is protected under
8       *   all applicable laws, including copyright laws.
9       *
10      *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11      *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12      *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13      *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14      *   DISCLAIMED.
15      *
16      *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17      *   TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18      *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19      *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20      *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21      *
22      *   Renesas reserves the right, without notice, to make changes to this
23      *   software and to discontinue the availability of this software.
24      *   By using this software, you agree to the additional terms and
25      *   conditions found by accessing the following link:
26      *   http://www.renesas.com/disclaimer
27      ************************************************************************
28      *   Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.
29      *""FILE COMMENT""*********** Technical reference data *********************
30      *   System Name : SH7264 Sample Program
31      *   File Name   : main.c
32      *   Abstract    : CAN Module Application (Remote Frame Transmit and Receive)
33      *   Version     : 1.00.00
34      *   Device      : SH7262/SH7264
35      *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36      *               : C/C++ compiler package for the SuperH RISC engine family
37      *               :                             (Ver.9.03 Release00).
38      *   OS          : None
39      *   H/W Platform: M3A-HS64G50 (CPU board) + M3A-HS64G02 (IO board)
40      *   Description :
41      ************************************************************************
42      *   History     : Nov.20,2009 ver.1.00.00
43      *""FILE COMMENT END""*****************************************************/
44      #include "iodefine.h"       /* SH7264 iodefine  */
45
```

3.3 Sample Program Listing "main.c" (2/3)

```
46    /* ---- prototype declaration ---- */
47    void main(void);
48    extern void io_init_port_ch0(void);
49    extern void io_init_port_ch1(void);
50    extern void io_init_can_ch0(void);
51    extern void io_init_can_ch1(void);
52    extern void io_remote_send_ch0(void);
53    extern void io_remote_receive_ch1(void);
54    extern void io_data_receive_ch0(void);
55
56    /*""FUNC COMMENT""*****************************************************************
57     * ID            :
58     * Outline       : Sample program main
59     *-----------------------------------------------------------------------------
60     * Include       : "iodefine.h"
61     *-----------------------------------------------------------------------------
62     * Declaration   : void main(void);
63     *-----------------------------------------------------------------------------
64     * Description   : After configuring the Controller Area Network (RCAN), channel 0
65     *               : transmits a remote frame, and receives a data frame.
66     *               : When channel 1 receives a remote frame, it automatically
67     *               : transmits a data frame.
68     *-----------------------------------------------------------------------------
69     * Argument      : void
70     *-----------------------------------------------------------------------------
71     * Return Value  : void
72     *-----------------------------------------------------------------------------
73     * Note          :
74     *""FUNC COMMENT END""*************************************************************/
75    void main(void)
76    {
77        /* ==== Initializing port ==== */
78        io_init_port_ch1();
79        io_init_port_ch0();
80
81        /* ==== Initializing CAN module ==== */
82        io_init_can_ch1();
83        io_init_can_ch0();
```

3.4 Sample Program Listing "main.c" (3/3)

```
84
85          /* ==== CAN remote frame transmission ==== */
86          io_remote_send_ch0();
87
88          /* ==== CAN remote frame reception ==== */
89          io_remote_receive_ch1();
90
91        /* ==== CAN data frame reception ==== */
92          io_data_receive_ch0();
93
94          while(1){
95              /* loop */
96          }
97      }
98
99      /* End of File */
```

3.5 Sample Program Listing "can0.c" (1/6)

```
1     /***********************************************************************
2     *   DISCLAIMER
3     *
4     *   This software is supplied by Renesas Technology Corp. and is only
5     *   intended for use with Renesas products. No other uses are authorized.
6     *
7     *   This software is owned by Renesas Technology Corp. and is protected under
8     *   all applicable laws, including copyright laws.
9     *
10    *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11    *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12    *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13    *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14    *   DISCLAIMED.
15    *
16    *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17    *   TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18    *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19    *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20    *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21    *
22    *   Renesas reserves the right, without notice, to make changes to this
23    *   software and to discontinue the availability of this software.
24    *   By using this software, you agree to the additional terms and
25    *   conditions found by accessing the following link:
26    *   http://www.renesas.com/disclaimer
27    ************************************************************************
28    *   Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.
29    *""FILE COMMENT""*********** Technical reference data *********************
30    *   System Name : SH7264 Sample Program
31    *   File Name   : can0.c
32    *   Abstract    : CAN Module Application (Remote Frame Transmit)
33    *   Version     : 1.00.00
34    *   Device      : SH7262/SH7264
35    *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36    *               : C/C++ compiler package for the SuperH RISC engine family
37    *               :                             (Ver.9.03 Release00).
38    *   OS          : None
39    *   H/W Platform: M3A-HS64G50 (CPU board) + M3A-HS64G02 (IO board)
40    *   Description :
41    ************************************************************************
42    *   History     : Nov.20,2009 ver.1.00.00
43    *""FILE COMMENT END""*****************************************************/
44    #include "iodefine.h"       /* SH7264 iodefine  */
45
```

3.6 Sample Program Listing "can0.c" (2/6)

```
46    /* ---- prototype declaration ---- */
47    void io_init_port_ch0(void);
48    void io_init_can_ch0(void);
49    void io_remote_send_ch0(void);
50    void io_data_receive_ch0(void);
51
52    /* ---- symbol definition ---- */
53    #define CAN_GSR3 0x0008
54    #define CAN_IRR0 0x0001
55    #define CAN_MB0  0x0001
56    #define CAN_MB1  0x0002
57    #define CAN_MB01 0x00000002
58
59    /* ---- RAM allocation variable declaration ---- */
60    static unsigned char   nIDE = 0;           /* ide */
61    static unsigned char   nRTR = 0;           /* rtr */
62    static unsigned char   nDLC = 0;           /* dlc */
63    static unsigned int    nSID = 0;           /* sid */
64    static unsigned int    nEID = 0;           /* eid */
65    unsigned char   gRcv_data[8];       /* data of message */
66
67    /*""FUNC COMMENT""*****************************************************************
68     * ID            :
69     * Outline       : PORT setting
70     *-------------------------------------------------------------------------------
71     * Include       : "iodefine.h"
72     *-------------------------------------------------------------------------------
73     * Declaration   : void io_init_port_ch0(void);
74     *-------------------------------------------------------------------------------
75     * Description   : Set pin functions (CRx0 input, CTx0 output).
76     *-------------------------------------------------------------------------------
77     * Argument      : void
78     *-------------------------------------------------------------------------------
79     * Return Value  : void
80     *-------------------------------------------------------------------------------
81     * Note          :
82     *""FUNC COMMENT END""*************************************************************/
83    void io_init_port_ch0(void)
84    {
85        /* ==== Setting of PORT ==== */
86        PORT.PJCR0.BIT.PJ0MD = 0x1;     /* Set CTx0 */
87        PORT.PJCR0.BIT.PJ1MD = 0x1;     /* Set CRx0 */
88    }
89
```

3.7 Sample Program Listing "can0.c" (3/6)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : RCAN setting
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void io_init_can_ch0(void);
 *-----------------------------------------------------------------------------
 * Description   : Configure the Controller Area Network channel 0.
 *               : Transfer rate is set as 1 Mbps.
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""*************************************************************/
void io_init_can_ch0(void)
{
    int i, j;

  /* ==== Setting of power down mode(RCAN) ==== */
  CPG.STBCR5.BIT.MSTP53 = 0;                /* Module Standby Clear (RCAN0)*/

  /* ==== Initializing CAN module ==== */
    RCAN0.MCR.WORD |= 0x0001;           /* CAN Interface reset mode */
    while((RCAN0.IRR.WORD & CAN_IRR0) != CAN_IRR0){
        /* Reset state waiting */
    }
    /* ==== IRR = 1, GSR = 1 (Auto SET) ==== */

    /* ---- Clear IRR0 ---- */
    RCAN0.IRR.WORD = 0x0001;

    /* ---- RCAN mode selection(MCR15) ---- */
    RCAN0.MCR.WORD |= 0x8000;           /* RCAN is not same as HCAN2 */

    /* ---- Disable all can interrupt ---- */
    RCAN0.IMR.WORD  = 0xffff;

    /* ----All mailbox init ---- */
    for(i = 0; i < 32; i++){
        RCAN0.MB[i].CONTROL0.LONG = 0x00000000;
        RCAN0.MB[i].LAFM.LONG = 0x00000000;
        for(j = 0; j < 8; j++){
            RCAN0.MB[i].MSG_DATA[j] =  0x00;
        }
    }

```

3.8 Sample Program Listing "can0.c" (4/6)

```
139       /* ---- Config mailbox0 as reception slot ---- */
140       RCAN0.MB[0].CONTROL1.WORD = 0x0200;       /* CAN receive data and remote frame */
141       RCAN0.MB[0].CONTROL0.LONG = 0x00000000;   /* Initialize the Message Control Field */
142       RCAN0.MB[0].LAFM.LONG = 0x00000000;
143       for(i = 0; i < 8; i++){                 /* data clear */
144           RCAN0.MB[0].MSG_DATA[i] =  0x00;
145       }
146       /* ---- Config mailbox1 as transmission slot ---- */
147       RCAN0.MB[1].CONTROL1.WORD = 0x0002;       /* CAN send data or remote frame, dlc=2 */
148       RCAN0.MB[1].CONTROL0.LONG = 0x00000000;   /* standard data frame, id=0x000 */
149       RCAN0.MB[1].LAFM.LONG = 0x00000000;
150       for(i = 0; i < 8; i++){                 /* data clear */
151           RCAN0.MB[1].MSG_DATA[i] =  0x00;
152       }
153
154       /* ---- Config baudrate ---- */
155       RCAN0.BCR1.WORD = 0x9600;      /* tsg1=9(10bit),tsg2=6(7bit),sjw=0(1bit),bsp=0 */
156       RCAN0.BCR0.WORD = 0x0000;      /*   1 Mbps */
157   //  RCAN0.BCR0.WORD = 0x0001;      /* 500 Kbps */
158   //  RCAN0.BCR0.WORD = 0x0003;      /* 250 Kbps */
159   //  RCAN0.BCR0.WORD = 0x0007;      /* 125 Kbps */
160
161       /* ---- Clear interrupt flags ---- */
162       RCAN0.IRR.WORD = 0xffff;
163
164       /* ---- Clear reset and halt ---- */
165       RCAN0.MCR.WORD &= 0xf8fc;      /* MCR0, MCR1 clear */
166       while( (RCAN0.GSR.WORD & CAN_GSR3) != 0x0000 ){
167           /* Reset state is end */
168       }
169   }
170
```

3.9 Sample Program Listing "can0.c" (5/6)

```
171   /*""FUNC COMMENT""*****************************************************************
172    * ID            :
173    * Outline       : Remote frame transmit
174    *-------------------------------------------------------------------------------
175    * Include       : "iodefine.h"
176    *-------------------------------------------------------------------------------
177    * Declaration   : void io_remote_send_ch0(void);
178    *-------------------------------------------------------------------------------
179    * Description   : Transmit a remote frame.
180    *-------------------------------------------------------------------------------
181    * Argument      : void
182    *-------------------------------------------------------------------------------
183    * Return Value  : void
184    *-------------------------------------------------------------------------------
185    * Note          :
186    *""FUNC COMMENT END""*************************************************************/
187   void io_remote_send_ch0(void)
188   {
189       /* ---- Transmission waiting ---- */
190       while((RCAN0.TXPR0.LONG & CAN_MB01) == CAN_MB01){
191       }
192
193       /* ---- Transmission data set ---- */
194       RCAN0.MB[1].CONTROL1.WORD = 0x0002;     /* CAN send data or remote frame, dlc=2 */
195       RCAN0.MB[1].CONTROL0.LONG = 0x40000000; /* standard remote frame, id=0x000 */
196
197       /* ---- Tramsmit the data ---- */
198       RCAN0.TXPR0.LONG = CAN_MB01;
199
200       /* ---- Transmission completion waiting ---- */
201       while((RCAN0.TXACK0.WORD & CAN_MB1) != CAN_MB1){
202       }
203
204       /* ---- Transmission completion flag clear ---- */
205       RCAN0.TXACK0.WORD = CAN_MB1;
206   }
207
```

3.10 Sample Program Listing "can0.c" (6/6)

```
/*""FUNC COMMENT""*****************************************************************
 * ID            :
 * Outline       : Data frame receive
 *-----------------------------------------------------------------------------
 * Include       : "iodefine.h"
 *-----------------------------------------------------------------------------
 * Declaration   : void io_data_receive_ch0(void);
 *-----------------------------------------------------------------------------
 * Description   : Receive a data frame.
 *-----------------------------------------------------------------------------
 * Argument      : void
 *-----------------------------------------------------------------------------
 * Return Value  : void
 *-----------------------------------------------------------------------------
 * Note          :
 *""FUNC COMMENT END""*************************************************************/
void io_data_receive_ch0(void)
{
    int i;

    /* ---- Reception completion waiting ---- */
    while((RCAN0.RXPR0.WORD & CAN_MB0) != CAN_MB0){
    }

    /* ---- Receive data storage ---- */
    nIDE = RCAN0.MB[0].CONTROL0.BIT.IDE;
    nRTR = RCAN0.MB[0].CONTROL0.BIT.RTR;
    nDLC = RCAN0.MB[0].CONTROL1.BIT.DLC;
    nSID = RCAN0.MB[0].CONTROL0.BIT.STDID;
    nEID = RCAN0.MB[0].CONTROL0.BIT.EXTID;
    if(nDLC > 8){
        nDLC = 8;
    }
    for(i = 0; i < nDLC; i++){
        gRcv_data[i] = RCAN0.MB[0].MSG_DATA[i];
    }

    /* ---- Reception completion flag clear ---- */
    RCAN0.RXPR0.WORD = CAN_MB0;
}
/* End of File */
```

4. References

- Software Manual
  SH-2A/SH2A-FPU Software Manual Rev. 3.00
  The latest version of the software manual can be downloaded from the Renesas website.

- Hardware Manual
  SH7262 Group, SH7264 Group Hardware Manual Rev. 2.00
  The latest version of the hardware manual can be downloaded from the Renesas website.

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jan. 08, 2010	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.