SH7216 Group

Controller Area Network, Configuration to Receive Data Frames

R01AN0944EJ0102
Rev.1.02
Jan. 19, 2012

Summary

This application note describes the configuration example of the SH7216 microcomputers (MCUs) to receive data frames using the controller area network (RCAN-ET).

Target Device

SH7216 MCU

Contents

1. Introduction 2
2. Applications 3
3. Sample Program Listing 10
4. References 16

1. Introduction

1.1 Specifications

- Transmission speed: 1 Mbps
- Receive mailbox: Mailbox 0
- Receives the 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	SH7216
Operating Frequencies	Internal clock: 200 MHz Bus clock: 50 MHz Peripheral clock: 50 MHz
Integrated Development Environment	Renesas Electronics Corporation High-performance Embedded Workshop Ver.4.05.01
C compiler	Renesas Electronics SuperH RISC engine Family C/C++ Compiler Package Ver.9.03 Release 00
Compiler options	-cpu=sh2afpu -fpu=single -include="$(WORKSPDIR)\inc" -object="$(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:

- SH7216 Group Controller Area Network, Configuration to Transmit Data Frames
- SH7216 Group Controller Area Network, Configuration to Transmit Remote Frames
- SH7216 Group Controller Area Network, Configuration to Receive Remote Frames

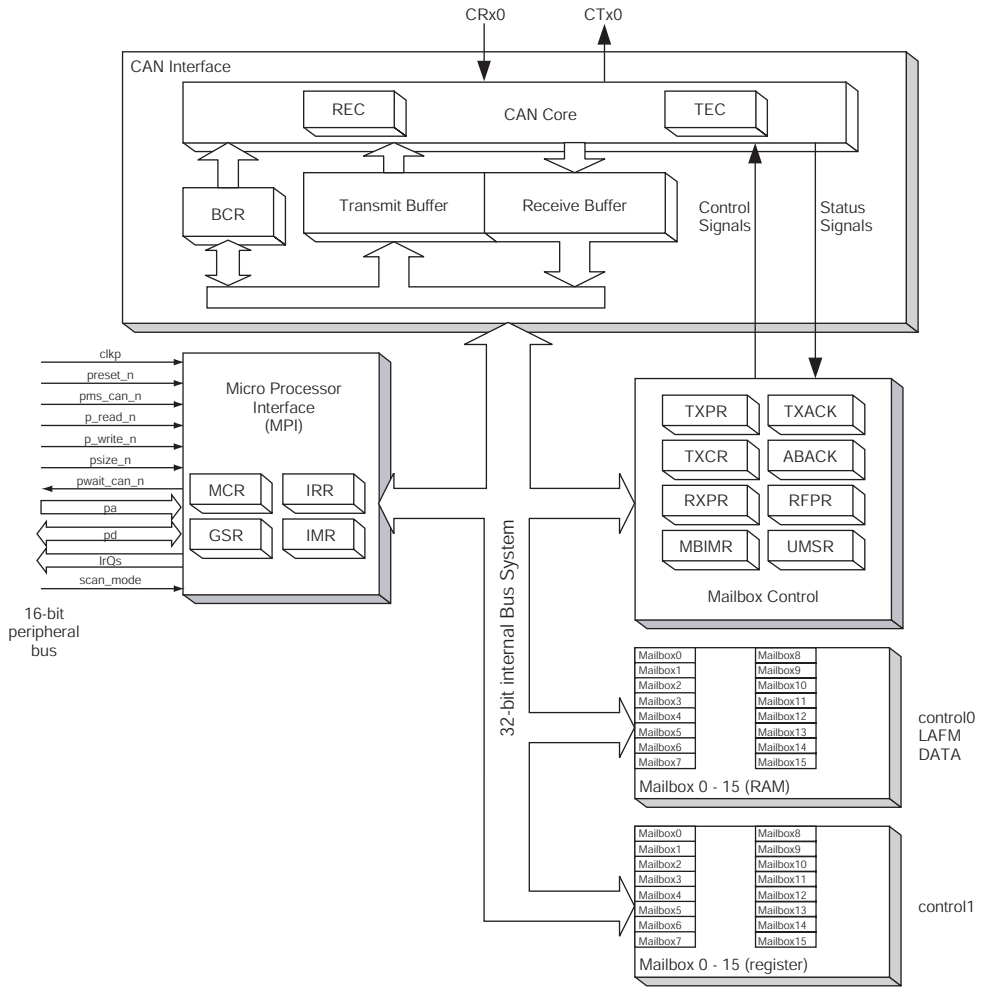
2. Applications

This application note uses the CAN module to receive a standard data frame with identifier 0.

2.1 CAN Overview

The SH7216 includes a CAN module which is compliant with the CAN protocol, version 2.0B active, and ISO 11898.

The CAN module has 15 programmable mailboxes for transmission/reception, one mailbox for reception, and one 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 SH7216 Group Hardware Manual.



Note: RCAN-ET core is designed based on the 32-bit bus system, however, 16-bit bus interface applies to CPU on the RCAN-ET (including the MPI to CPU). Therefore, execute the longword (32-bit) access using two neighboring words (16-bit). The longword access explained in the Hardware Manual means to access by such two neighboring words.

Figure 1 CAN Block Diagram

2.2 Configuration Procedure

This section describes how to configure the SH7216 MCU to receive data 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 sets two mailboxes in SH7216 - one transmit mailbox and one receive mailbox. **Figure 2** and **Figure 3** show the flow charts for configuring the CAN module. For details on register settings, refer to the SH7216 Group Hardware Manual.

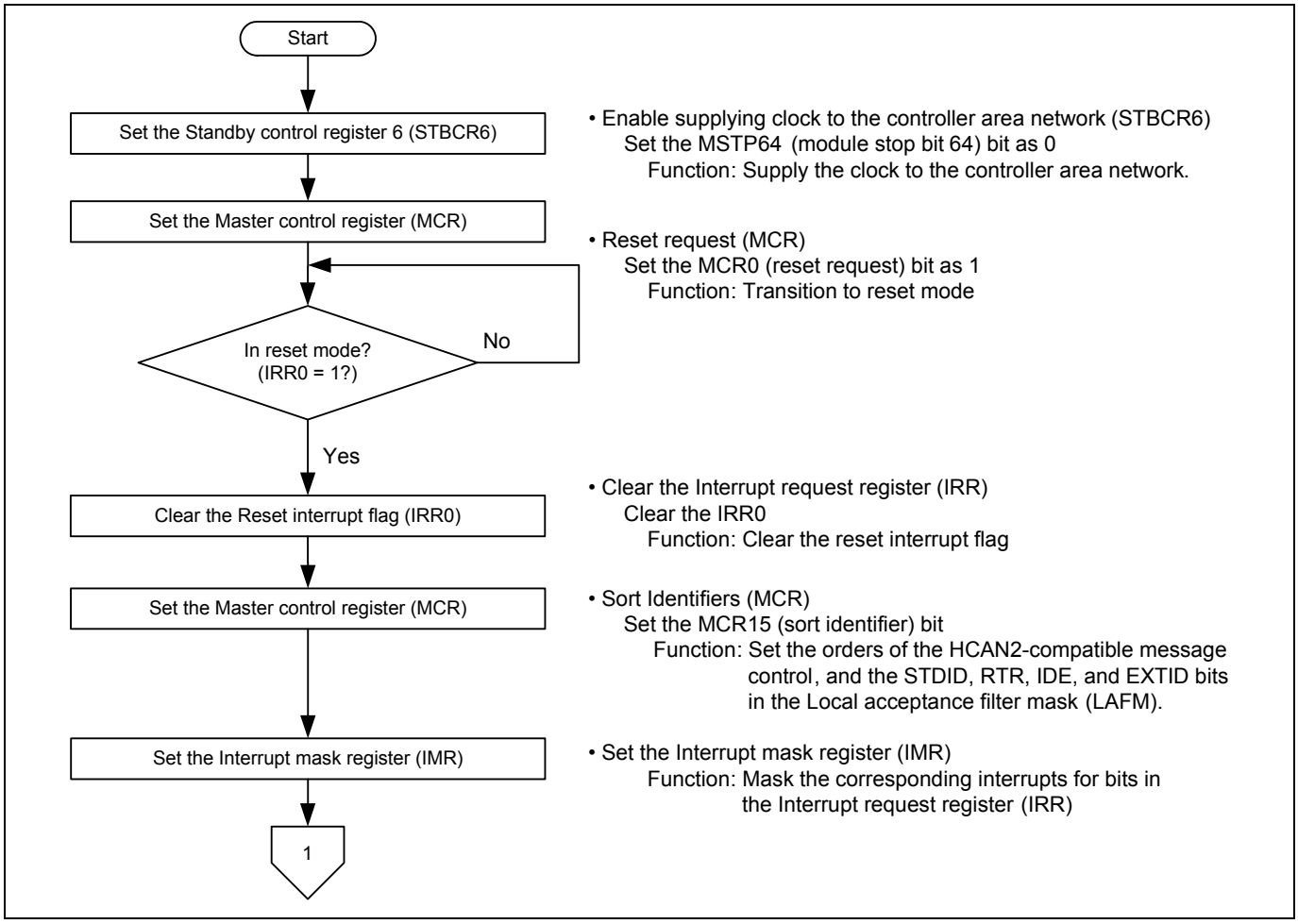
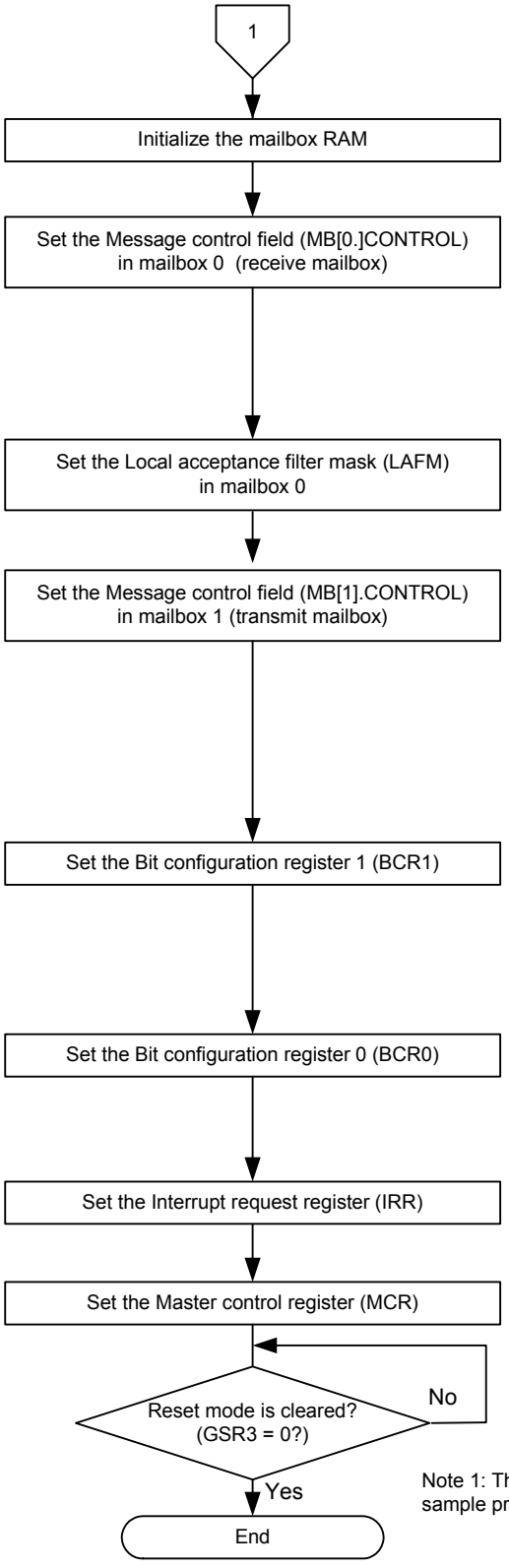


Figure 2 Flow Chart for Configuring the CAN Module (1/2)



• Clear the RAM area in the mailbox

• Set the Message control register (MB[0].CONTROL)
Set "B'010" in the MBC[2:0] bits (Mailbox configuration)
Function: Receiving a remote frame or a data frame is enabled
Set "0" in the IDE bit (Identifier extension)
Function: Specify the standard format
Set "0" in the RTR bit (Remote transmission request)
Function: Specify a data frame
Set "0" in the STDID[10:0] bits (Standard Identifier)
Function: Specify 0 as the ID
• Set the Local acceptance filter mask (LAFM)
Set H'0000 0000 in the LAFM
Function: Mask is not specified

• Set the Message control register (MB[1].CONTROL) (1)
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 DLC bit (Data length code)
Function: Specify the data length as 2 bytes
Specify "0" in IDE bit (Identifier extension)
Function: Specify the standard format
Specify "0" in 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 Bit configuration register 1 (BCR1)
Specify "B'1111" (16 time quanta) in TSG1 [3:0] bits (Time segment 1)
Specify "B'111" (8 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
Specify 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 x peripheral bus clock
Transfer speed is 1 Mbps by TSG1, TSG2, and baud rate prescaler.
(Refer to 2.3 Bit configuration and transfer 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

Note 1: This operation is not required in receiving data, however, this sample program assumes both transmission and reception.

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 = Tq, PRSEG = 8 Tq, PHSEG1 = 8 Tq, and PHSEG2 = 8 Tq.

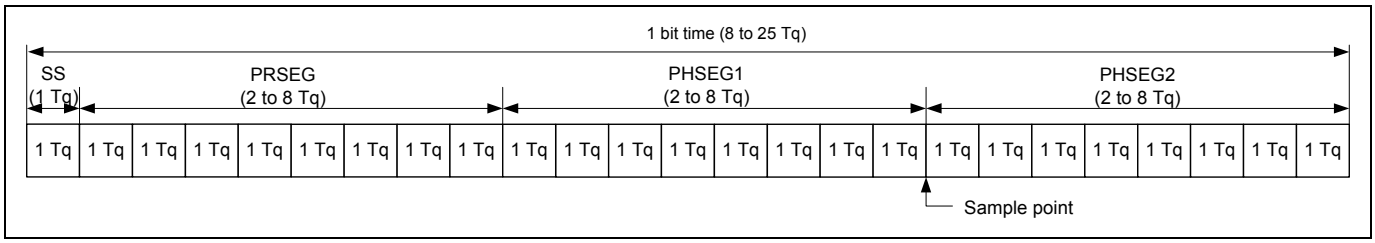


Figure 4 Bit Configuration

The CAN module sets the number of Tqs of PRSEG + PHSEG1 to bits TSG1 [3:0] in the BCR1 register, 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 register.

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 SH7216 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:

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

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

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.

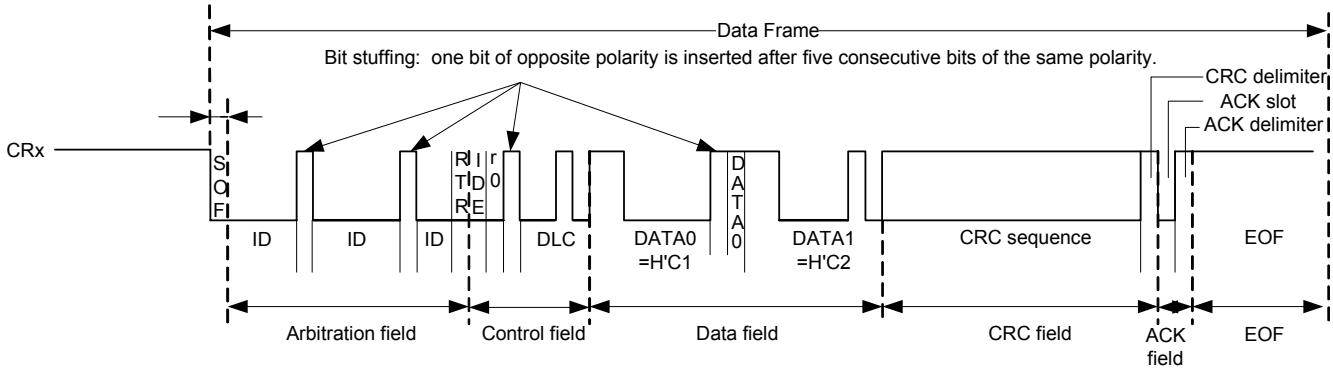
8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta
TSEG2 ≥ 2

As this sample program specifies the peripheral bus clock as 50 MHz, BRP = 0, TSEG 1 = 15, and TSEG2 = 7, the transmission speed is calculated as follows:

$$\text{Transmission speed} = \frac{50\mathrm{M}}{(2\times(0+1)\times(15+1)+(7+1)+1)} = 1\mathrm{M}....1\ \mathrm{Mbps}$$

2.4 Sample Program Operation

This sample program receives a standard data frame in mailbox 0 at 1 Mbps. **Figure 5** shows the reception waveform.



Legend:

SOF (Start of frame): Indicates the beginning of a data frame or remote frame
ID (Identifier): Indicates the data priority
RTR (Remote transmission request): Indicates differences between a data frame and a 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.)
DATA: This is the data to transfer. Number of bytes DATA specified by the DLC is transferred.
CRC sequence: This is the value for frame check.
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 a frame

Figure 5 CAN Reception 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.

Table 1 CAN Settings

Register Name	Address	Setting	Description
Standby control register (STBCR6)	H'FFFE 041C	H'8F	MSTP64 = "0": CAN is operating
Master control register (MCR)	H'FFFF D000	H'0001	MCR0 = "1": Reset mode transition request
		H'8001	MCR15 = "1": The order of the RCAN-ET message and of the HCAN2 message are different
		H'8000	MCR0 = "0": Reset mode is cleared
Interrupt mask register (IMR)	H'FFFF D00A	H'FFFF	All interrupts in the CAN are disabled
Bit configuration register 1 (BCR1)	H'FFFF D004	H'F700	TSEG1 [3:0] = "B'1111": PRSEG + PHSEG1 = 16 Tq TSEG2 [2:0] = "B'111": PHSEG2 = 8 Tq SJW="0": 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 x Pφ
Message control field in mailbox 0 (MB[0].CONTROL1H)	H'FFFF D110	H'0200	MBC [2:0] = "B'010": Receiving the data frame or remote frame is enabled
Message control field in mailbox 1 (MB[1].CONTROL1H)	H'FFFF D130	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
Mailbox 1 control field in mailbox 0 (MB[1].CONTROL0H)	H'FFFF D120	H'0000 0000	IDE = "0": Standard format RTR = "0": Data frame STDID [10:0] = "0": Standard identifier is 0
Local acceptance filter mask in mailbox 0 (MB[0].LAFMH)	H'FFFF D104	H'0000 0000	Clear: Mask is not specified
Message data field in mailbox 1 (MB[1].MSG_DATA_0)	H'FFFF D128	H'0000	Clear the data field (Clear the RAM area)
Data frame receive pending register (RxPR0)	H'FFFF D042	H'0001	RXPR [31:0] = H'0001: Clear the receive pending flag

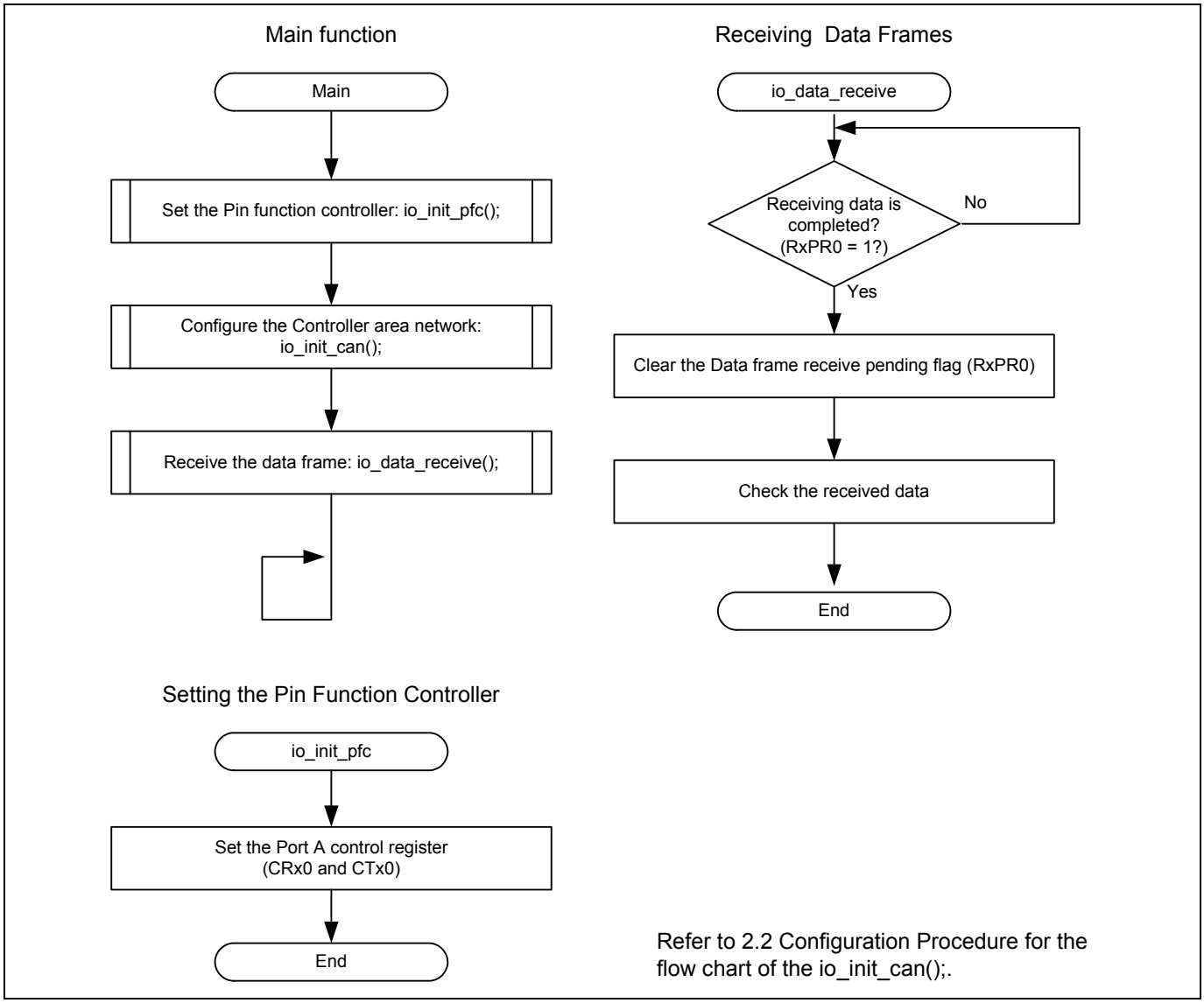


Figure 6 Sample Program Flow Chart

3. Sample Program Listing

3.1 Sample Program Listing "main.c" (1/6)

```
/*****************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
*****************************************************************************
*   Copyright (C) 2009(2010) Renesas Electronics Corporation.
*   All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7216 Sample Program
*   File Name   : main.c
*   Abstract    : CAN Module Application (Data Frame Receive).
*   Version     : 1.01.00
*   Device      : SH7216
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.03 Release00).
*   OS          : None
*   H/W Platform: R0K572167 (CPU board)
*   Description :
*****************************************************************************
*   History     : Jun.29,2009 Ver.1.00.00
*               : Jun.21,2010 Ver.1.01.00  Changing the corporate name and
*               :                           the copyright format
*               :                          - Dividing FRQCR setting into
*               :                             subroutine "io_set_cpg_frqcr"
*               :                             which is allocated to on-chip RAM
*               :                          - Adding processing of section copy
*               :                             function "_seccpy"
```

3.2 Sample Program Listing "main.c" (2/6)

```
51    *""FILE COMMENT END""**********************************************************/
52    #include "iodefine.h"
53
54    /* ---- prototype declaration ---- */
55    void main(void);
56    void io_init_pfc(void);
57    void io_init_can(void);
58    void io_data_receive(void);
59
60    /* ---- symbol definition ---- */
61    #define CAN_GSR3 0x0008
62    #define CAN_IRR0 0x0001
63    #define CAN_MB0  0x0001
64
65    /* ---- RAM allocation variable declaration ---- */
66    unsigned char   nIDE = 0;           /* ide */
67    unsigned char   nRTR = 0;           /* rtr */
68    unsigned char   nDLC = 0;           /* dlc */
69    unsigned int    nSID = 0;           /* sid */
70    unsigned int    nEID = 0;           /* eid */
71    unsigned char   gRcv_data[8];       /* data of message */
72
73    /*""FUNC COMMENT""***************************************************************
74     * ID            :
75     * Outline       : Sample program main
76     *-------------------------------------------------------------------------------
77     * Include       : "iodefine.h"
78     *-------------------------------------------------------------------------------
79     * Declaration   : void main(void);
80     *-------------------------------------------------------------------------------
81     * Description   : Set the PFC and configure the RCAN to receive the data frame.
82     *-------------------------------------------------------------------------------
83     * Argument      : void
84     *-------------------------------------------------------------------------------
85     * Return Value  : void
86     *-------------------------------------------------------------------------------
87     * Note          :
88     *""FUNC COMMENT END""***********************************************************/
89    void main(void)
90    {
91        /* ==== Setting of PFC ==== */
92        io_init_pfc();
93
94        /* ==== Initializing CAN module ==== */
95        io_init_can();
96
97        /* ==== Receives a CAN data frame ==== */
98        io_data_receive();
99
100       while(1){
101           /* loop */
102       }
103   }
```

3.3 Sample Program Listing "main.c" (3/6)

```
104
105  /*""FUNC COMMENT""*****************************************************************
106   * ID            :
107   * Outline       : PFC setting
108   *-------------------------------------------------------------------------------
109   * Include       : "iodefine.h"
110   *-------------------------------------------------------------------------------
111   * Declaration   : void io_init_pfc(void);
112   *-------------------------------------------------------------------------------
113   * Description   : Set pin functions.
114   *               : CRx0 input, CTx0 output
115   *-------------------------------------------------------------------------------
116   * Argument      : void
117   *-------------------------------------------------------------------------------
118   * Return Value  : void
119   *-------------------------------------------------------------------------------
120   * Note          :
121   *""FUNC COMMENT END""*************************************************************/
122  void io_init_pfc(void)
123  {
124      /* ==== Setting of PFC ==== */
125      /* ---- Port A control register L2 ---- */
126      PFC.PACRL1.BIT.PA0MD = 0x5;     /* Set CRx0 */
127      PFC.PACRL1.BIT.PA1MD = 0x5;     /* Set CTx0 */
128  }
129
130  /*""FUNC COMMENT""*****************************************************************
131   * ID            :
132   * Outline       : RCAN setting
133   *-------------------------------------------------------------------------------
134   * Include       : "iodefine.h"
135   *-------------------------------------------------------------------------------
136   * Declaration   : void io_init_can(void);
137   *-------------------------------------------------------------------------------
138   * Description   : Configure the Controller Area Network (RCAN).
139   *               : Transfer rate is at 1 Mbps.
140   *-------------------------------------------------------------------------------
141   * Argument      : void
142   *-------------------------------------------------------------------------------
143   * Return Value  : void
144   *-------------------------------------------------------------------------------
145   * Note          :
146   *""FUNC COMMENT END""*************************************************************/
147  void io_init_can(void)
148  {
149      int i,j;
150
151      /* ==== Setting of power down mode(RCAN) ==== */
152      STB.CR6.BYTE = 0x8f;                /* Module Standby Clear */
153                                          /* RCAN */
```

3.4 Sample Program Listing "main.c" (4/6)

```
154       /* ==== Initializing CAN module ==== */
155       RCANET.MCR.WORD |= 0x0001;          /* CAN Interface reset mode */
156       while((RCANET.IRR.WORD & CAN_IRR0) != CAN_IRR0){
157           /* Reset state waiting */
158       }
159       /* ==== IRR = 1, GSR = 1 (Auto SET) ==== */
160
161       /* ---- Clear IRR0 ---- */
162       RCANET.IRR.WORD = 0x0001;
163
164       /* ---- RCAN mode selection(MCR15) ---- */
165       RCANET.MCR.WORD |= 0x8000;          /* RCAN-ET is not same as HCAN2 */
166
167       /* ---- Disable all can interrupt ---- */
168       RCANET.IMR.WORD  = 0xffff;
169
170       /* ----All mailbox init ---- */
171       for(i = 0; i < 16; i++){
172           RCANET.MB[i].CTRL0.LONG = 0x00000000;
173           RCANET.MB[i].LAFM.LONG = 0x00000000;
174           for(j = 0; j < 8; j++){
175               RCANET.MB[i].MSG_DATA[j] =  0x00;
176           }
177       }
178
179       /* ---- Config mailbox0 as reception slot ---- */
180       RCANET.MB[0].CTRL1.WORD = 0x0200;       /* Receiving data frame or remote frame */
181                                               /* is enabled */
182       RCANET.MB[0].CTRL0.LONG = 0x00000000;   /* Initialize the Message Control Field */
183       RCANET.MB[0].LAFM.LONG = 0x00000000;
184       for(i = 0; i < 8; i++){                 /* Data clear */
185           RCANET.MB[0].MSG_DATA[i] =  0x00;
186       }
187       /* ---- Config mailbox1 as transmission slot ---- */
188       RCANET.MB[1].CTRL1.WORD = 0x0002;       /* Transmitting data frame or remote frame */
189                                               /* with dlc=2 is enabled */
190       RCANET.MB[1].CTRL0.LONG = 0x00000000;   /* Standard data frame, id=0x000 */
191       RCANET.MB[1].LAFM.LONG = 0x00000000;
192       for(i = 0; i < 8; i++){                 /* Data clear */
193           RCANET.MB[1].MSG_DATA[i] =  0x00;
194       }
195
```

3.5 Sample Program Listing "main.c" (5/6)

```
196         /* ---- Config baud rate ---- */
197         RCANET.BCR1.WORD = 0xf700;      /* tsg1=15(16-bit),tsg2=7(8-bit),sjw=0(1-bit),bsp=0 */
198         RCANET.BCR0.WORD = 0x0000;      /*   1 Mbps */
199     //  RCANET.BCR0.WORD = 0x0001;      /* 500 Kbps */
200     //  RCANET.BCR0.WORD = 0x0003;      /* 250 Kbps */
201     //  RCANET.BCR0.WORD = 0x0007;      /* 125 Kbps */
202
203         /* ---- Clear interrupt flags ---- */
204         RCANET.IRR.WORD = 0xffff;
205
206         /* ---- Clear reset and halt ---- */
207         RCANET.MCR.WORD &= 0xf8fc;      /* MCR0,MCR1 clear */
208         while( (RCANET.GSR.WORD & CAN_GSR3) != 0x0000 ){
209             /* Reset state is end */
210         }
211     }
212
```

3.6 Sample Program Listing "main.c" (6/6)

```
213  /*""FUNC COMMENT""*****************************************************************
214   * ID            :
215   * Outline       : Data frame receive
216   *-------------------------------------------------------------------------------
217   * Include       : "iodefine.h"
218   *-------------------------------------------------------------------------------
219   * Declaration   : void io_data_receive(void);
220   *-------------------------------------------------------------------------------
221   * Description   : After completing to receive the data frame, this function
222   *               : stores the mailbox 0 received data in the gRcv_data[i].
223   *-------------------------------------------------------------------------------
224   * Argument      : void
225   *-------------------------------------------------------------------------------
226   * Return Value  : void
227   *-------------------------------------------------------------------------------
228   * Note          :
229   *""FUNC COMMENT END""*************************************************************/
230  void io_data_receive(void)
231  {
232      int i;
233
234      /* ---- Waits for completing to receive data ---- */
235      while((RCANET.RXPR0.WORD & CAN_MB0) != CAN_MB0){
236      }
237
238      /* ---- Stores the receive data ---- */
239      nIDE = RCANET.MB[0].CTRL0.BIT.IDE;
240      nRTR = RCANET.MB[0].CTRL0.BIT.RTR;
241      nDLC = RCANET.MB[0].CTRL1.BIT.DLC;
242      nSID = RCANET.MB[0].CTRL0.BIT.STDID;
243      nEID = RCANET.MB[0].CTRL0.BIT.EXDID;
244      if(nDLC > 8){
245          nDLC = 8;
246      }
247      for(i = 0; i < nDLC; i++){
248          gRcv_data[i] = RCANET.MB[0].MSG_DATA[i];
249      }
250
251      /* ---- Reception pending flag clear ---- */
252      RCANET.RXPR0.WORD = CAN_MB0;
253  }
254
255  /* 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
  SH7216 Group Hardware Manual Rev. 3.00
  The latest version of the hardware manual can be downloaded from the Renesas website.

Website and Support

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

Inquiries
http://www.renesas.com/contact

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Jul 24, 2009	—	First edition issued
1.01	Jul.24, 2010	—	Changing the corporate name and the copyright format
1.02	Jan.16, 2012	—	Changing the R-number and the copyright format.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1